Distributed Threshold-based Offloading for Heterogeneous Mobile Edge Computing

Xudong Qin* Qiaomin Xie† Bin Li*
*School of EECS, Pennsylvania State University, State College, Pennsylvania, USA
†Department of ISyE, University of Wisconsin-Madison, Madison, Wisconsin, USA

***Abstract*—In this paper, we consider a large-scale heterogeneous mobile edge computing system, where each device's mean computing task arrival rate, mean service rate, mean energy consumption, and mean offloading latency are drawn from different bounded continuous probability distributions to reflect the diverse compute-intensive applications, mobile devices with different computing capabilities and battery efficiencies, and different types of wireless access networks (e.g., 4G/5G cellular networks, WiFi). We consider a class of distributed threshold-based randomized offloading policies and develop a threshold update algorithm based on its computational load, average offloading latency, average energy consumption, and edge server processing time, depending on the server utilization. We show that there always exists a unique Mean-Field Nash Equilibrium (MFNE) in the large-system limit when the task processing times of mobile devices follow an exponential distribution. This is achieved by carefully partitioning the space of mean arrival rates to account for the discrete structure of each device's optimal threshold. Moreover, we show that our proposed threshold update algorithm converges to the MFNE. Finally, we perform simulations to corroborate our theoretical results and demonstrate that our proposed algorithm still performs well in more general setups based on the collected real-world data and outperforms the well-known probabilistic offloading policy.**

I. INTRODUCTION

With the trend of pushing artificial intelligence to mobile devices with constrained CPU/GPU capabilities, many applications leverage mobile edge computing schemes to enable real-time compute-intensive machine learning tasks such as Internet of Things (IoT) health monitoring systems (e.g., [1], [2]) and animals monitoring and tracking on farms with IoT devices and edge computing systems (e.g., [3]). This is achieved by offloading compute-intensive tasks to powerful edge servers to reduce the task processing time and energy consumption of mobile devices. However, users experience offloading latency and processing latency at edge servers as well as offloading energy consumption if they offload their computing tasks to the edge servers. The processing delay at edge servers depends on the edge server utilization. The larger the server utilization, the larger the processing delay at edge servers. As such, when more users offload their computing tasks to the edge, they experience large processing delays at edge servers. Therefore, a central question in mobile edge computing systems is how each device offloads its computing tasks to the edge to optimize the task processing delay and energy consumption.

While edge computing has received significant research interest in recent years (see [4], [5] for a comprehensive survey), much of the prior work on mobile edge computing systems (e.g., [6], [7], [8], [9], [10]) focused on the static model, where the profiles of all computing tasks (such as the number of tasks, each task's processing time) are available before the algorithm operation. For example, [6] and [7] developed offloading algorithms that minimize the average energy consumption in mobile devices. [8] proposed offloading strategies that minimize the average task processing latency. [9] and [10] jointly optimized energy efficiency and task processing latency in mobile devices. However, this line of work fails to capture the dynamics of computing tasks, which is ubiquitous in practical systems.

There have been some works on edge computing systems (e.g., [11], [12], [13], [14]) considering the dynamic model, where computing tasks dynamically arrive at the IoT devices and are processed either by local devices or edge/cloud servers. However, they focused on centralized solutions based on a stochastic network optimization framework (see [15] for an overview) and thus did not apply to large-scale edge computing systems. Another line of research work (e.g., [16], [17]) considered a distributed probabilistic offloading design for the dynamic model, where each mobile device determines its offloading probability that its computing tasks are uploaded to edge servers to minimize the average cost. For example, the authors in [17] formulated a game theory model to determine the offloading probability.

On the other hand, the cost optimization for the dynamic model can be formulated as a Markov decision process (MDP) problem whose optimal solutions typically have threshold-based structure (e.g., [18], [19]): an incoming task is processed locally if the number of tasks in the local device is less than some threshold, and offloaded to edge servers otherwise. Indeed, the threshold-based policy typically outperforms the well-studied probabilistic offloading policy (see Section IV-C). Moreover, threshold-based policies have a distributed nature and are easy to be deployed in large-scale mobile edge computing systems.

As such, we are interested in the class of distributed threshold-based offloading policies, where each user makes its offloading decision based on its own threshold. In a recent work [20], the authors considered a distributed threshold-based algorithm design for large-scale homogeneous mobile edge computing, where all mobile devices have the same task arrival

This work has been supported in part by NSF under the grants CNS-2152657, CNS-2152658 and CNS-1955997.

rate and service rate. However, mobile edge computing systems are *heterogeneous*, consisting of diverse IoT devices with different CPU/GPU capabilities and battery efficiencies, different types of wireless access networks (e.g., 4G/5G cellular networks, WiFi), and diverse compute-intensive applications. The algorithm and analysis developed in [20] do not apply to such a heterogeneous edge computing system.

In this paper, we consider a *heterogeneous* mobile edge computing system, where each mobile device's mean computing task arrival rate, mean service rate, the average energy consumption of processing and offloading a task, and mean offloading latency are drawn from different bounded continuous probability distributions to model the system heterogeneity such as diverse compute-intensive applications, mobile devices with different computing capabilities and energy consumption, and various wireless access networks. We focus on the class of distributed threshold-based algorithms. Given the distributed nature of the threshold-based offloading policy, we are interested in investigating whether there exists a unique Mean-Field Nash Equilibrium (MFNE) under which each device has no incentive to deviate from its optimal threshold. If such a unique MFNE exists, can we design a distributed threshold update algorithm under which the system converges to equilibrium? The main challenges to answering these questions are the following: (i) In contrast to the distributed probabilistic offloading algorithms that optimize the offloading probability, the optimal thresholds for the distributed threshold-based offloading policies exhibit a discrete nature, which, together with the heterogeneity of the system, makes it difficult to characterize the MFNE; (ii) It is challenging to develop a distributed threshold update algorithm that converges to the MFNE, since each device only has its local task processing information, energy consumption, edge server utilization, and offloading latency, without the knowledge of any other devices' information.

The main results and contributions of this paper are summarized as follows:

- We propose a Distributed Threshold Update (DTU) Algorithm that iteratively updates each device's threshold based on its average queue length, task offloading latency, energy consumption, and edge server utilization (see Algorithm 1 in Section III-A).
- We show that there always exists a unique Mean-Field Nash Equilibrium (MFNE) in the large-system limit when the task processing time in local devices follows an exponential distribution (see Theorem 1 in Section III-B). The proof is quite involved since the optimal threshold exhibits a discrete nature for each individual device. It is non-trivial to establish the continuity of the best response function with respect to server utilization. We tackle this challenge by partitioning the space of mean arrival rates in a novel way.
- We further show that our proposed DTU Algorithm converges to the unique MFNE. This is achieved by exploring the bisection property of the "estimated" server utilization, i.e., it always increases or decreases towards the MFNE (see Theorem 2 in Section III-B).
- In Section IV, we first perform simulations to validate our theoretical findings. We then demonstrate that our proposed DTU Algorithm still performs well in practical setups, including real-world data for local processing time, offloading latency, and asynchronous threshold updates. Finally, we demonstrate the superior performance of our proposed algorithm over the well-studied distributed probabilistic offloading policy.

II. SYSTEM MODEL

We consider a mobile edge computing system with N IoT devices (referred to as *users*), where users offload computing tasks to edge servers via wireless networks, as shown in Fig. 1. Tasks arrive at each user n $(n = 1, 2, \cdots, N)$ according to the Poisson process with the rate a_n. Each user n decides whether the newly arriving task is offloaded to the edge or processed in its local device. We assume that each user n is able to process a newly arriving task in the local device with mean service time $1/s_n$, and each user n maintains a queue in its local device that holds incoming tasks and processes tasks in the First-Come-First-Serve (FCFS) manner. We let $q_n(t)$ be the queue length of user n at time t, denoting the number of awaiting tasks. If a task is offloaded to edge servers, it can be processed with the total service rate of Nc, where c is the service capacity such that all the computing tasks of each user can be processed at edge servers.

To model the heterogeneity of computing applications and mobile devices, we assume that both mean arrival rate a_n and mean service rate s_n are sampled from probability distributions of independent bounded non-negative continuous random variables A and S, respectively. In particular, we assume that $0 < A \leq A_{\max}$ and $S_{\min} \leq S \leq S_{\max}$ for some positive constants $A_{\max}, S_{\min}$, and $S_{\max}$. As such, users have different mean task arrival rates and service rates, capturing the fact that they might use different compute-intensive applications and their devices have different processing capabilities. Here, we assume that $A_{\max} < c$ (and hence $a_n < c, \forall n$) to ensure that all incoming tasks can be processed by edge servers.

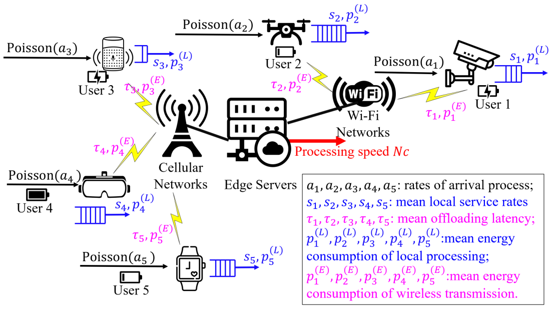

Poisson(a_2)
$s_2, p_2^{(L)}$
Poisson(a_3)
$s_3, p_3^{(L)}$
User 2
Poisson(a_1)
$s_1, p_1^{(L)}$
User 3
$\tau_3, p_3^{(E)}$
$\tau_2, p_2^{(E)}$
Wi-Fi
User 1
$\tau_1, p_1^{(E)}$
Wi-Fi Networks
Processing speed Nc
$\tau_4, p_4^{(E)}$
Cellular Networks
Edge Servers
Poisson(a_4)
User 4
$s_4, p_4^{(L)}$
$\tau_5, p_5^{(E)}$
Poisson(a_5)
User 5
$s_5, p_5^{(L)}$
a_1, a_2, a_3, a_4, a_5: rates of arrival process;
s_1, s_2, s_3, s_4, s_5: mean local service rates
$\tau_1, \tau_2, \tau_3, \tau_4, \tau_5$: mean offloading latency;
$p_1^{(L)}, p_2^{(L)}, p_3^{(L)}, p_4^{(L)}, p_5^{(L)}$:mean energy consumption of local processing;
$p_1^{(E)}, p_2^{(E)}, p_3^{(E)}, p_4^{(E)}, p_5^{(E)}$:mean energy consumption of wireless transmission.

Fig. 1: A system with $N = 5$ users

On the one hand, computing tasks that are processed locally will suffer from both queueing delays and large processing delays in local devices due to constrained local processing capabilities. On the other hand, offloading computing tasks to high-performance edge servers can be processed much faster while experiencing task offloading latency (including both communication delay and processing time in edge servers).

In order to model the heterogeneity of each user's network conditions, we assume that offloading latency of each computing task of user n follows a probability distribution with mean τ_n, where τ_n is sampled from a probability distribution of a bounded non-negative continuous random variable T, and $0 < T \leq T_{\max}$ for some positive constant $T_{\max}$. We let $\gamma \in [0,1]$ denote the edge *server utilization* and let function $g(\gamma)$ denote the delay experienced at edge servers when the current server utilization is γ, where $g : [0,1] \mapsto [0, G_{\max}]$ is an increasing and continuous function for some positive constant $G_{\max}$. This is motivated by the fact that a larger server utility typically results in a larger delay.

For each incoming task, each user n processes tasks locally and offloads tasks to edge servers with the average energy consumption of $p_{n,L}$ and $p_{n,E}$ per task[1], respectively. To capture the heterogeneity of users' energy consumption, we further assume that both $p_{n,L}$ and $p_{n,E}$ are sampled from the probability distribution of two different bounded non-negative continuous random variables P_L and P_E, respectively, where $0 < P_L \leq P_{L,\max}$ and $0 < P_E \leq P_{E,\max}$ for some positive constants $P_{L,\max}$ and $P_{E,\max}$, respectively.

To minimize both the computing delay of tasks and the energy consumption of mobile devices, each user needs to carefully decide whether a newly incoming computing task will be processed locally or offloaded to edge servers. Note that the offloading problem of each user shares a similar structure with the optimal admission control of a single queue whose solution has a threshold-based structure (see [21]). In addition, threshold-based policies are easy to implement in a distributed manner, especially when there are a lot of IoT devices. Moreover, we demonstrate via simulations (see Section IV-C) that the threshold-based policy outperforms widely-studied probabilistic offloading policy (e.g., [22], [23], [24] and [25]) under which each user offloads incoming computing tasks with a certain probability.

To that end, we focus on the following threshold-based offloading policy for each user. Let $\lfloor y \rfloor$ denote the largest integer that is not greater than y, and we recall that $q_n(t)$ is the number of computing tasks in the user's local device n at time t. Then, we consider the following Threshold-based Randomized Offloading (TRO) policy:

Threshold-based Randomized Offloading (TRO) Policy: Each user n with a real-value threshold $x_n \geq 0$ makes the following offloading decision when a new computing task arrives:

(i) If $q_n(t) < \lfloor x_n \rfloor$, then the new task joins the local device;

(ii) If $q_n(t) = \lfloor x_n \rfloor$, then the new task joins the local device with probability $x_n - \lfloor x_n \rfloor$ and is uploaded to edge servers with probability $1 - (x_n - \lfloor x_n \rfloor)$;

(iii) If $q_n(t) \geq \lfloor x_n \rfloor + 1$, then the new task will be uploaded to edge servers.

Under the TRO policy with a threshold x_n, when $x_n = 0$, user n will upload all its incoming tasks to edge servers; when $x_n = 2.5$, user n will admit an incoming task locally if its queue-length is less than 2, and upload the incoming task with probability 0.5 if its queue-length is equal to 2, and upload the incoming task if its queue-length is greater than or equal to 3. Note that our TRO policy is a generalization of the threshold-based offloading policy studied in [20], where the threshold parameters are integers. We let $Q_n(x_n)$ denote the *average queue-length* of user n, and let $\alpha_n(x_n)$ be the average task *offloading probability* of user n (i.e., the fraction of time offloading tasks). While both the average queue length and the average task offloading probability depend on the arrival rate, service time distribution, and threshold decision, we explicitly use $Q_n(x_n)$ and $\alpha_n(x_n)$ to emphasize the dependence on the threshold x_n, which will be optimized to minimize the average computing delay and average energy consumption.

For each incoming task at user n with threshold parameter x_n, if it is processed locally, then it experiences the average delay $\frac{Q_n(x_n)}{a_n(1-\alpha_n(x_n))}$ by Little's Law, and its average energy consumption is $p_{n,L}$, where $a_n(1-\alpha_n(x_n))$ denotes the average arrival rate of computing tasks processed in the local device. If it is offloaded to the edge servers with the utilization γ, then it incurs the offloading latency of mean τ_n and the processing delay $g(\gamma)$ at the edge servers as well as the average energy consumption $p_{n,E}$ for wireless transmissions. Noting that the task is offloaded to the edge servers with probability $\alpha_n(x_n)$, the average cost (including both average computing delay and energy consumption) of user n with threshold x_n is defined as follows:

$$w_n(1-\alpha_n(x_n))p_{n,L} + \frac{Q_n(x_n)}{a_n} + (w_n p_{n,E} + g(\gamma) + \tau_n)\alpha_n(x_n), \quad (1)$$

where $0 < w_n \leq w_{\max}, \forall n$ are system weight parameters that characterize the trade-off between task processing latency cost and energy consumption, and $w_{\max}$ is some positive constant. The larger the parameter w_n, the more emphasis on the energy consumption in the overall cost of user n (cf. (1)).

In this paper, we are interested in the large-scale mobile edge computing system (i.e., N is sufficiently large). We aim to develop a *distributed offloading algorithm* under which each user updates its own threshold to minimize its cost function without knowing all other users' thresholds. This raises two fundamental questions: 1) does such an algorithm converge? 2) If so, what does it converge to? We address these questions from a mean field game perspective. In particular, we assume our considered system operates in a Quasi-Stationary manner as the number of users $N \to \infty$, i.e., each user optimizes their cost (1) in a slower time scale while the server utilization is updated in a faster time scale. Therefore, the server utilization is a constant from the users' point of view whenever users update their thresholds (see [26] and [27] for more detailed explanations about the two different time scales).

Here, we consider two mappings that characterize server utilization and users' thresholds updating, respectively. We first define $J_1 : (x_n)_{n=1}^N \to \gamma$, i.e., given all users' thresholds $(x_n)_{n=1}^N$, we have a server utilization $\gamma \in [0,1]$, which

[1] Our model can be easily adapted to the case with the average energy consumption per time unit.

is updated in a faster time scale. Then, for the fixed edge server utilization $\gamma \in [0,1]$, each user n minimizes its own cost function (1) and obtains a new threshold x_n based on server utilization γ. We define this process as mapping $J_2 : \gamma \to (x_n)_{n=1}^N$, which occurs in a slower time scale.

Having characterized the two different mappings, we define γ^* to be the *Mean Field Nash Equilibrium (MFNE)* of the system if and only if

$$\gamma^* = J_1(J_2(\gamma^*)). \tag{2}$$

In the mapping J_2, each user plays their *best response*, minimizing the cost function (1), given the current server utilization γ. The resulting average edge server utilization is $\sum_{n=1}^N a_n \alpha_n(x_n^*(\gamma; a_n, \theta_n, \tau_n, p_{n,L}, p_{n,E}))/(Nc)$, which converges to $\mathbb{E}_{A,\Theta,T,P_L,P_E}\left[A\alpha(x^*(\gamma; A, \Theta, T, P_L, P_E))/c\right]$ almost surely as $N \to \infty$ according to the Strong Law of Large Numbers, where $\Theta \triangleq A/S$. Therefore, the MFNE γ^* (cf. Eq. (2)) can be rewritten as

$$\gamma^* = \mathbb{E}_{A,\Theta,T,P_L,P_E}\left[\frac{A\alpha(x^*(\gamma^*; A, \Theta, T, P_L, P_E))}{c}\right]. \tag{3}$$

Here, we study the problem in a large-system limit (i.e., $N \to \infty$) where each user's decision on the threshold has a minimal impact on the server utilization γ and thus, each user treats the server utilization as a fixed constant when optimizing its own cost function. If the system reaches the MFNE (when it exists), each user adopts the optimal threshold; thus, no user has the incentive to change the current threshold unilaterally, and the server utilization will remain the same.

We remark that our game formulation in the large-system regime corresponds to the so-called mean field game (MFG) [28], [29]). However, to the best of our knowledge, our problem is not a special case of any existing work, and their analysis of MFNE is not applicable to our setting. In particular, the existing literature on MFG (e.g., [30], [31], [32], [33]) primarily focused on either finite-time horizon or infinite horizon with discounted cost. In contrast, our problem involves infinite-horizon average cost—including average queue length, average offloading cost, and average energy consumption. Some recent work on MFG with infinite-horizon average cost focuses on settings with homogeneous players/users (e.g., [34]), while we consider heterogeneous users (i.e., each user has its own arrival rate, service rate, average offloading latency, and average energy consumption). Furthermore, some work (e.g., [32]) assumed the cost function to be continuously differentiable, while the cost function in our problem is not differentiable everywhere (i.e., it is not differentiable at all integer points, as shown in Fig. 8 in Appendix A).

Next, we develop a distributed threshold update algorithm and show that it converges to the unique MFNE, assuming that the user's local processing time follows an exponential distribution.

III. Algorithm Design and Main Results

In this section, we present a distributed threshold update algorithm under which each user iteratively updates its threshold based on server utilization information of the edge servers without knowing any other users' threshold information. Then, under the assumption of exponential processing time in the local devices, we show that a unique MFNE always exists, and our proposed algorithm converges to this MFNE.

A. Algorithm Description

In this subsection, we present a Distributed Threshold Update (DTU) Algorithm under which each user iteratively updates its threshold based on edge server utilization. Let $\epsilon \in (0,1)$ be a given parameter that controls the convergence accuracy of the DTU Algorithm. We use $\gamma_t \in (0,1)$ and $\widehat{\gamma}_t \in [0,1]$ to denote the true server utilization and the "estimated" server utilization in the t^{th} iteration, respectively. We use η_t to denote the non-increasing step size in the t^{th} iteration. We introduce a counter L to control the step size η_t. Let $\widehat{x}_n^{(t)}$ be the optimal threshold of user n at the t^{th} iteration given the estimated server utilization $\widehat{\gamma}_t$.

Algorithm 1 Distributed Threshold Update (DTU) Algorithm

```
Given any $0 < \eta_0 \le 1, \widehat{\gamma}_0 = 0, \widehat{\gamma}_{-1} = 1, 0 < \epsilon < 1$, and
  $L = 1, t = 1$, performs the following:
while $|\widehat{\gamma}_{t-1} - \widehat{\gamma}_{t-2}| > \epsilon$, the edge servers do
    $\widehat{\gamma}_t \leftarrow \min\left\{1, \widehat{\gamma}_{t-1} + \eta_{t-1} \cdot \frac{\gamma_t - \widehat{\gamma}_{t-1}}{|\gamma_t - \widehat{\gamma}_{t-1}|}\right\}$,   (4)
    and broadcasts $\widehat{\gamma}_t$ to all users.
    for each user $n = 1, 2, \cdots, N$ do
        $\widehat{x}_n^{(t+1)} \in \operatorname*{argmin}_{x_n \ge 0} \Big\{ w_n p_{n,L}(1 - \alpha_n(x_n)) + \frac{Q_n(x_n)}{a_n}$
            $+ (w_n p_{n,E} + g(\widehat{\gamma}_t) + \tau_n)\, \alpha_n(x_n) \Big\}$.   (5)
    end for
    The edge servers perform:
    if $t \ge 2$ and $\widehat{\gamma}_t = \widehat{\gamma}_{t-2}$ then
        $L \leftarrow L + 1$,
        $\eta_t \leftarrow \frac{\eta_0}{L}$,
    else
        $\eta_t \leftarrow \eta_{t-1}$.
    end if
    $\gamma_{t+1} \leftarrow \frac{1}{N} \sum_{n=1}^{N} \frac{a_n \alpha_n\left(\widehat{x}_n^{(t+1)}\right)}{c}$.   (6)
    $t \leftarrow t + 1$.
end while
```

While the server utilization information γ_t is available at the beginning of each iteration t, we introduce the "estimated" server utilization $\widehat{\gamma}_t$ to facilitate each user to control its own threshold decision. The motivation comes from the observation that the server utilization γ_{t-1} in the previous iteration for the threshold decision-making in (5) in the t^{th} iteration is out of the system's control since γ_{t-1} depends on all users' thresholds, and users do not share their threshold updates information with edge servers. Moreover, directly using actual server utilization in (5) does not have a theoretical guarantee that the algorithm will converge. Therefore, we choose to use "estimated" server utilization $\widehat{\gamma}_t$ rather than the actual server utilization γ_t.

According to (4), if the server utilization is underestimated (i.e., $\widehat{\gamma}_{t-1} < \gamma_t$), then the "estimated" server utilization will increase (i.e., $\widehat{\gamma}_t > \widehat{\gamma}_{t-1}$). Otherwise, it will decrease. In addition, if the "estimated" server utilization oscillates (i.e., $\widehat{\gamma}_t = \widehat{\gamma}_{t-2}$), then it implies that the convergence point is between $\widehat{\gamma}_t$ and $\widehat{\gamma}_{t-2}$ and thus we need to reduce the step size to make sure that the "estimated" server utilization is closer to the convergence point. As such, the "estimated" server utilization gets closer and closer to the desired value and eventually converges.

Next, we are interested in understanding whether our proposed DTU Algorithm can converge, and if it does, will it converge to the MFNE of the system? To answer these questions, we assume that the processing time of each task for each user follows an exponential distribution. Under this assumption, the number of tasks of each user forms a Continuous-Time Markov Chain (CTMC) under the TRO policy with threshold x, and thus we can explicitly calculate the average queue-length $Q(x)$ and offloading probability $\alpha(x)$ given its arrival rate a and service rate s, i.e.,

$$Q(x) = \begin{cases} \pi_0\left(\frac{\theta(1-\theta^{\lfloor x\rfloor})}{(1-\theta)^2} + (\lfloor x\rfloor + 1)(x - \lfloor x\rfloor)\theta^{\lfloor x\rfloor+1} \right. \\ \qquad\qquad\qquad \left. -\frac{\lfloor x\rfloor\theta^{\lfloor x\rfloor+1}}{1-\theta}\right), \text{ if } \theta \neq 1, \\ \frac{(\lfloor x\rfloor+1)(2x-\lfloor x\rfloor)}{2(x+1)}, \text{ if } \theta = 1, \end{cases} \tag{7}$$

$$\alpha(x) = \begin{cases} \frac{(1-\theta)\theta^{\lfloor x\rfloor}(1-(1-\theta)(x-\lfloor x\rfloor))}{1-\theta^{\lfloor x\rfloor+1}+(x-\lfloor x\rfloor)(1-\theta)\theta^{\lfloor x\rfloor+1}}, \text{ if } \theta \neq 1, \\ \frac{1}{x+1}, \text{ if } \theta = 1, \end{cases} \tag{8}$$

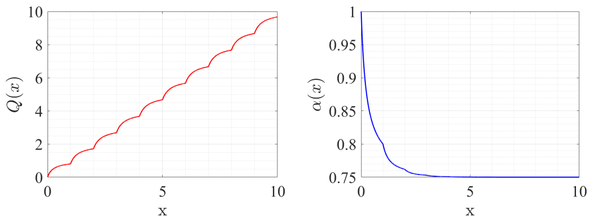
where we recall that $\theta = a/s$ denotes the arrival intensity, and

$$\pi_0 = \frac{1-\theta}{1-\theta^{\lfloor x\rfloor+1} + (x - \lfloor x\rfloor)(1-\theta)\theta^{\lfloor x\rfloor+1}}$$

represents the probability of no tasks in the local device.

It is easy to verify via basic calculus that for any fixed arrival intensity θ, both $Q(x)$ and $\alpha(x)$ are continuous with respect to the threshold x. Fig. 2a and Fig. 2b show $Q(x)$ and $\alpha(x)$ with respect to x with arrival intensity $\theta = 4$, illustrating that both of them are indeed continuous with respect to x.

We remark that our distributed threshold update algorithm is different from the algorithm developed in [20] for a *homogeneous* mobile cloud computing, where all mobile devices have the same arrival rate and service rate. In particular, we introduce the "estimated" server utilization to facilitate each user to control its own threshold decision. Moreover, under our proposed threshold updating algorithm, the best response function with respect to (w.r.t.) the server utilization first maps interval $[0, 1]$ to a non-negative integer space and then maps it back to the interval $[0, 1]$. Such a mapping introduces a significant challenge in proving the continuity of the best response function w.r.t. the server utilization, which is the key to establishing the existence and uniqueness of the MFNE.



(a) $Q(x)$ with respect to x (b) $\alpha(x)$ with respective x

Fig. 2: $\alpha(x)$ and $Q(x)$ when $\theta = 4$.

B. Main Results

In this subsection, we study the convergence property of our proposed DTU Algorithm in the large-system limit (i.e., $N \to \infty$) under the assumption that the processing time of computing tasks in local devices follows an exponential distribution. In particular, we first prove that the considered large-scale heterogeneous mobile edge computing system always has a unique MFNE and then show that our proposed DTU Algorithm converges to this unique MFNE.

Theorem 1: There always exists a unique MFNE $\gamma^* \in (0, 1)$ under the assumption that the local processing time of each task for each user follows an exponential distribution.

Proof: Here, we provide a proof sketch. We first characterize the optimal solution $x^*(\gamma; a, \theta, \tau, p_L, p_E)$ that minimizes the individual user's average cost, given her arrival rate a, arrival intensity θ (i.e., the service rate is equal to $a\theta$), mean offloading latency τ, average local processing energy consumption p_L, and offloading energy consumption p_E as well as the edge server utilization γ. Based on the structure of the optimal solution, we are able to capture the server utilization after users' threshold decision update given the current server utilization γ (called the *best response*), i.e.,

$$V(\gamma) \triangleq \mathbb{E}_{A,\Theta,T,P_L,P_E}\left[\frac{A\alpha(x^*(\gamma; A, \Theta, T, P_L, P_E))}{c}\right]. \tag{9}$$

Then, we show that $V(\gamma)$ is continuous and non-increasing. Finally, noting that $V(0) < 1$ under the assumption that $A_{\max} < c$ (guaranteeing all tasks can be processed by the edge) and the fact that the offloading probability $\alpha(\cdot)$ is always not greater than 1, we conclude that there always exists the unique solution γ^* to the equation $V(\gamma) = \gamma$ and $\gamma^* \in (0, 1)$. The detailed proof is available in Appendix A. ■

Remarks 1: The proof is highly non-trivial since the optimal threshold $x^*(\gamma; a, \theta, \tau, p_L, p_E)$ is a non-negative integer, thus the best response function $V(\cdot)$ (cf. Eq. (9)) is a composition of the mapping from $\gamma \in [0, 1]$ to non-negative integers $x^*(\gamma; a, \theta, \tau, p_L, p_E)$ and the mapping from non-negative integers to a multi-dimensional integral. As such, it is not obvious that $V(\gamma)$ is continuous with respect to γ, which is crucial to establish the uniqueness and existence of MFNE. As illustrated in Fig. 3, the offloading probability of each user is discontinuous with respect to the server utilization γ. Nevertheless, we can establish the continuity of $V(\gamma)$ by carefully partitioning the space of mean arrival rates with the following two nice properties: (i) the total number of partitions is almost continuous with respect to the server utilization γ; (ii) all users have the same optimal threshold within each partition and hence $V(\gamma)$ is continuous within each partition.

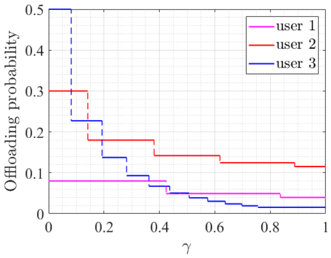
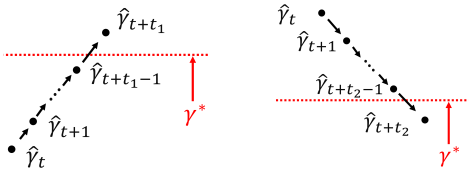


Fig. 3: User's offloading prob. w.r.t. server utilization γ.

Theorem 1 indicates that there always exists a unique MFNE such that all users in the system have no incentive to deviate from their current optimal thresholds. The next theorem shows that our proposed DTU Algorithm converges to the unique NE $\gamma^* \in (0,1)$.

Theorem 2: The proposed DTU Algorithm eventually converges to the unique MFNE γ^*.

Proof: The proof is based on the bisection property of "estimated" server utilization $\widehat{\gamma}_t$ under the DTU Algorithm, i.e., $\widehat{\gamma}_t$ always increases or decreases towards the MFNE γ^*. In particular, we show that there exist the following two cases: (i) If $\widehat{\gamma}_t < \gamma^*$, then $\widehat{\gamma}_t$ will increase until $\widehat{\gamma}_{t+t_1} > \gamma^*$ for some $t_1 > 0$, as demonstrated in Fig. 4a; (ii) If $\widehat{\gamma}_t > \gamma^*$, then $\widehat{\gamma}_t$ will decrease until $\widehat{\gamma}_{t+t_2} < \gamma^*$ for some $t_2 > 0$, as shown in Fig. 4b. In both cases, $\widehat{\gamma}_t$ gets closer and closer to γ^* under our step size update rule. The detailed proof is provided in Appendix D. ■



(a) Case $\widehat{\gamma}_t < \gamma^*$

(b) Case $\widehat{\gamma}_t > \gamma^*$

Fig. 4: Dynamics of $\widehat{\gamma}_t$

While the existence and uniqueness of the MFNE and the convergence of our proposed DTU Algorithm are established under the assumption that the task processing time of each user follows an exponential distribution. Our simulations (cf. Section IV-B) demonstrate that the results still hold under general scenarios, such as real-world tasks' local processing time and offloading latency distributions and asynchronous threshold updates.

IV. Simulations

In this section, we first perform simulations to validate our theoretical results (cf. Theorem 1 and 2) and then demonstrate that our proposed algorithm works well in practical setups, including real-world local processing time and offloading latency distributions, and asynchronous threshold updates. Finally, we demonstrate the superior performance of the proposed DTU Algorithm over the well-studied probabilistic offloading algorithm counterpart.

A. Validation of Theoretical Results

In this subsection, we first perform simulations to validate the existence and uniqueness of the MFNE (cf. Theorem 1), where the processing times of computing tasks in local devices follow an exponential distribution. Then, we run simulations to validate the convergence of the DTU Algorithm (cf. Theorem 2). We consider $N = 10^4$ users and the cost of using edge servers given by $g(\gamma) = 1/(1.1 - \gamma)$. Mean arrival rate A, mean service rate S, and offloading latency T follow different uniform distributions. In particular, we consider $S \sim U(1,5)$, $T \sim U(0,1)$, $P_L \sim U(0,3)$, $P_E \sim U(0,1)$ and $w_n = 1, \forall n$ for all simulation setups, while we consider three different uniform distributions for A: (i) $A \sim U(0,4)$ under which $\mathbb{E}[A] < \mathbb{E}[S]$; (ii) $A \sim U(0,6)$ under which $\mathbb{E}[A] = \mathbb{E}[S]$; (iii) $A \sim U(0,8)$ under which $\mathbb{E}[A] > \mathbb{E}[S]$.

We first run numerical simulations using the Monte Carlo method to obtain the unique MFNE under our theoretical settings with different distributions for the mean arrival rate. Table I summarizes the unique MFNE under three different setups in our numerical simulation. From Table I, we can see that the unique MFNE is $0.13, 0.21$ and 0.28 when $\mathbb{E}[A] < \mathbb{E}[S], \mathbb{E}[A] = \mathbb{E}[S]$ and $\mathbb{E}[A] > \mathbb{E}[S]$, respectively.

System Setup	NE
$\mathbb{E}[A] < \mathbb{E}[S]$	$\gamma^* = 0.13$
$\mathbb{E}[A] = \mathbb{E}[S]$	$\gamma^* = 0.21$
$\mathbb{E}[A] > \mathbb{E}[S]$	$\gamma^* = 0.28$

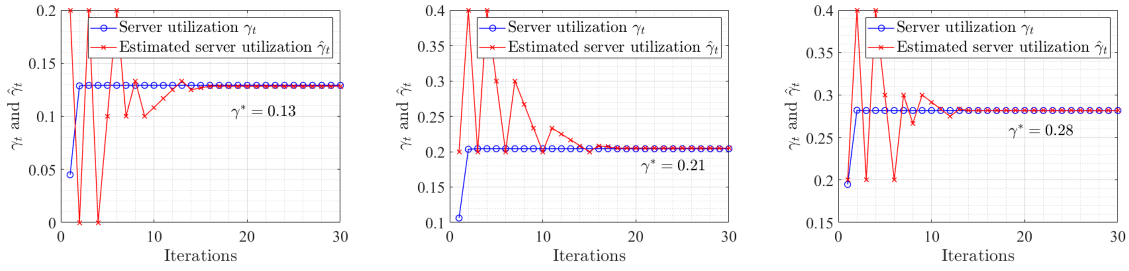
TABLE I: MFNE under theoretical settings.

Fig. 5 demonstrates the convergence of the DTU Algorithm. We can see from Fig. 5a that both server utilization γ_t and "estimated" server utilization $\widehat{\gamma}_t$ converge to the unique MFNE $\gamma^* = 0.13$ (in the case with $\mathbb{E}[A] < \mathbb{E}[S]$) within 20 iterations. Moreover, "estimated" server utilization also exhibits a bisection property, i.e., always increasing or decreasing towards the MFNE. Similarly, we can observe from Fig. 5b and Fig. 5c that our proposed algorithm converges to the corresponding MFNE $\gamma^* = 0.21$ (in the case with $\mathbb{E}[A] = \mathbb{E}[S]$) and $\gamma^* = 0.28$ (in the case with $\mathbb{E}[A] > \mathbb{E}[S]$), respectively, around 20 iterations.

B. Convergence under Practical Scenarios

In this subsection, we consider a variety of practical simulation setups: each user's local processing time is measured from image recognition applications; offloading latency of each user is measured using a mobile device in a wireless network environment; each user updates its threshold asynchronously. The probability distributions for the mean arrival rate A, the average energy consumption of the local device, and the average offloading energy consumption remain the same as in Section IV-A.

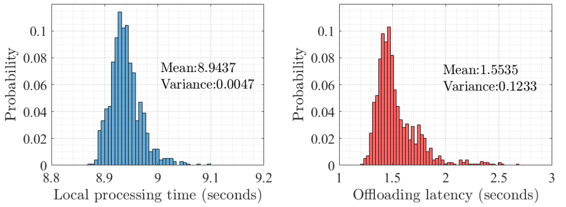
In particular, we first implement YOLOv3 (see [35] and [36] for more details about the YOLOv3 framework), which



(a) $\mathbb{E}[A] < \mathbb{E}[S]$
(b) $\mathbb{E}[A] = \mathbb{E}[S]$
(c) $\mathbb{E}[A] > \mathbb{E}[S]$

Fig. 5: Convergence of DTU Algorithm under theoretical settings.

is a real-time object detection framework on a Raspberry Pi 4 microcontroller board to emulate task processing process on mobile devices. We then perform object detection tasks using 1000 different images (see VOC2012 [37] for the details of the image dataset) on the Raspberry Pi 4 microcontroller board with the YOLOv3 framework implemented, and we measure the object detection time for each image, respectively. Furthermore, we use the Raspberry Pi 4 to upload the same 1000 images to Google Drive via WiFi network and collect the offloading latency for each image, respectively. Fig. 6 shows the normalized histogram of the real-world data we collected.



(a) Local processing time
(b) Offloading latency

Fig. 6: Statistics of the data we have collected.

Moreover, in each iteration of the DTU Algorithm, each user updates its threshold with probability 0.8 to model asynchronous updates. Recall from Section IV-A, we set the cost of using edge servers $g(\gamma) = 1/(1.1 - \gamma)$ and let $P_L \sim U(0,3)$, $P_E \sim U(0,1)$ and $w_n = 1, \forall n$. We consider $N = 10^3$ users in different simulation setups. We also consider three different uniform distributions for the arrival rate A: (i) $\mathbb{E}[A] = 8 < \mathbb{E}[S] = 8.9437$ for $A \sim U(4,12)$; (ii) $\mathbb{E}[A] = \mathbb{E}[S] = 8.9437$ for $A \sim U(7.3474, 10.54)$; (iii) $\mathbb{E}[A] = 10 > \mathbb{E}[S] = 8.9437$ for $A \sim U(8,12)$.

System Setup	NE
$\mathbb{E}[A] < \mathbb{E}[S]$	$\gamma^* = 0.43$
$\mathbb{E}[A] = \mathbb{E}[S]$	$\gamma^* = 0.44$
$\mathbb{E}[A] > \mathbb{E}[S]$	$\gamma^* = 0.46$

TABLE II: MFNE under practical settings.

Similar to Section IV-A, we summarize the unique NE under three different setups in Table II. We can observe from Table II that there exists the unique MFNE $\gamma^* = 0.43$ (in the case with $\mathbb{E}[A] < \mathbb{E}[S]$), $\gamma^* = 0.44$ (in the case with $\mathbb{E}[A] = \mathbb{E}[S]$) and $\gamma^* = 0.46$ (in the case with $\mathbb{E}[A] > \mathbb{E}[S]$), respectively. Moreover, Fig.7 demonstrates that our proposed DTU Algorithm converges to the corresponding unique NE within 20 iterations.

C. *Comparison with Probabilistic Counterpart*

In this subsection, we demonstrate the superior performance of the DTU Algorithm compared to the well-studied Distributed Probabilistic Offloading (DPO) policy (e.g., [22], [23] and [25]) under which each user selects the offloading probability to minimize its own cost. We consider $N = 10^3$ users, $T \sim U(0,5)$, $P_L \sim U(0,3)$ and $P_E \sim U(0,1)$ for theoretical settings. As for practical settings, we use the same settings as in Section IV-B and the real-world data we have collected. For the DPO policy, we perform repeated simulations under the same setting 5×10^3 times and calculate 98% confidence interval for the mean cost.

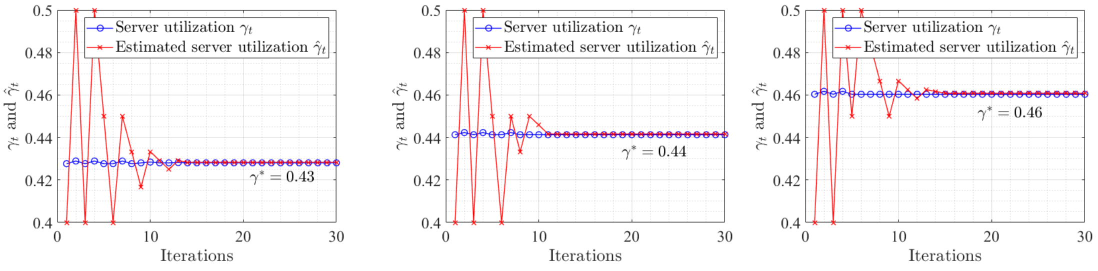
In particular, we perform simulations in two different scenarios: (i) theoretical settings: $S \sim U(1,5)$ and $A \sim U(0, A_{\max})$ for $A_{\max} = 4, 6$ and 8, respectively; (ii) Practical settings: we let $\mathbb{E}[A] = 8, 8.9437$ and 10, respectively. Both mean local processing time S and offloading latency T are sampled from the real-world data we have collected in Section IV-B. We can see from Table III that the average cost under our proposed DTU algorithm for both theoretical and practical settings reduces cost up to 30.76% and 20.07% compared to the DPO policy, respectively.

Policies and Costs		Cost under DTU algorithm	Mean cost under DPO Policy (98% Confidence Interval)	Cost reduction (%)
Theoretical settings	$\mathbb{E}[A] < \mathbb{E}[S]$	2.33	3.04 ± 0.0018	30.76
	$\mathbb{E}[A] = \mathbb{E}[S]$	2.58	3.18 ± 0.0015	23.26
	$\mathbb{E}[A] > \mathbb{E}[S]$	2.84	3.27± 0.0014	15.14
Practical settings	$\mathbb{E}[A] < \mathbb{E}[S]$	11.56	13.88 ± 0.0004	20.07
	$\mathbb{E}[A] = \mathbb{E}[S]$	11.46	13.59 ± 0.0005	18.50
	$\mathbb{E}[A] > \mathbb{E}[S]$	11.42	13.42 ± 0.0005	17.51

TABLE III: DTU Algorithm vs. DPO Policy.

V. Conclusion

In this paper, we considered large-scale heterogeneous mobile edge computing systems for IoT applications, where each user's mean arrival rate, mean service rate, mean offloading latency, and mean energy consumption are drawn from different bounded continuous probability distributions. We focused on a class of distributed threshold-based randomized offloading policies and developed a distributed threshold update algorithm under which each user updates its threshold to minimize its average cost, consisting of local processing delay, offloading latency, edge server processing delay, and average energy consumption, without any information from other users thresholds. We showed that there always exists a unique Mean-Field Nash Equilibrium (MFNE) in the large-system limit



(a) $\mathbb{E}[A] < \mathbb{E}[S]$ (b) $\mathbb{E}[A] = \mathbb{E}[S]$ (c) $\mathbb{E}[A] > \mathbb{E}[S]$

Fig. 7: Convergence of DTU Algorithm under practical settings

such that all users' thresholds and server utilization remain unchanged if the system reaches NE under the assumption that users' task processing time in local devices follows an exponential distribution. We further showed that our proposed algorithm converges to the unique MFNE. Finally, we performed simulations to corroborate our theoretical findings and demonstrated that our proposed algorithm can still work well in practical setups and significantly outperforms the well-known distributed probabilistic offloading algorithm.

Appendix A
Proof of Theorem 1

Our proof starts with characterizing the optimal solution $x^*(\gamma; a, \theta, \tau, p_L, p_E)$ that minimizes the user's cost function (cf. (1)) given his/her arrival rate a, arrival intensity θ, and mean offloading latency τ as well as the edge server utilization γ. To that end, we consider the following function:

$$f(m|\theta) \triangleq \begin{cases} 0, & \text{if } m = 0, \\ \sum_{i=1}^{m}(m-i+1)\theta^i, & \forall m \in \mathbb{N}^+, \end{cases} \tag{10}$$

where $\mathbb{N}^+$ denotes the set of natural numbers, i.e., $\mathbb{N}^+ \triangleq \{1, 2, 3, \cdots\}$. It is easy to see that $f(m|\theta)$ is strictly increasing with respect to m given any $\theta > 0$.

Next, we have the following lemma that characterizes the solution to the minimization of the user's cost function.

Lemma 1: Given the user's arrival rate $a > 0$, arrival intensity $\theta > 0$, server utilization $\gamma \in [0, 1]$, mean offloading latency $\tau > 0$, wireless transmission energy consumption p_E and local processing energy consumption p_L, if $-w_{\max}A_{\max}P_{L,\max} < a(g(\gamma) + \tau + w(p_E - p_L)) < f(1|\theta)$, then $x^*(\gamma; a, \theta, \tau, p_L, p_E) = 0$; if $f(m|\theta) \leq a(g(\gamma) + \tau + w(p_E - p_L)) < f(m+1|\theta)$ for some positive integer m, then $x^*(\gamma; a, \theta, \tau, p_L, p_E) = m$.

The proof of Lemma 1 is available in Appendix B. Based on Lemma 1, we are ready to investigate the properties of $V(\gamma) \triangleq \mathbb{E}_{A,\Theta,T,P_L,P_E}\left[A\alpha(x^*(\gamma; A, \Theta, T, P_L, P_E))/c\right]$.

Lemma 2: $V(\gamma)$ is continuous and non-increasing with respect to $\gamma \in [0, 1]$.

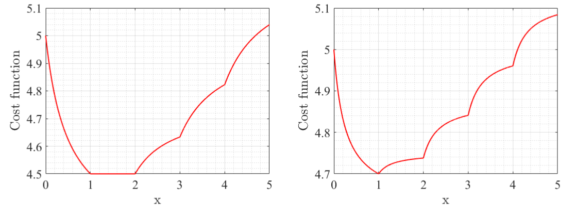
The proof of Lemma 2 is available in Appendix C. Lemma 2, together with $V(0) < 1$ under the assumption that $A_{\max} < c$ (ensuring that all computing tasks can be processed by edge servers, which is typically true in practice.) and the fact that the offloading probability $\alpha(\cdot)$ is always not greater than 1, implies that there always exists the unique solution γ^* to the equation $V(\gamma) = \gamma$ and $\gamma^* \in (0, 1)$. ■

Appendix B
Proof of Lemma 1

To facilitate our proof, we use $T(x|\gamma)$ to denote the individual user's cost function when the threshold is $x \geq 0$ given his/her arrival rate $a > 0$, offloading latency $\tau > 0$, arrival intensity $\theta > 0$, and the edge server utilization $\gamma \in [0, 1]$, i.e.,

$$T(x|\gamma) \triangleq wp_L(1 - \alpha(x)) + \frac{Q(x)}{a} + (g(\gamma) + \tau + wp_E)\,\alpha(x),$$

where we recall that $Q(x)$ and $\alpha(x)$ are the average queue length and the offloading probability, respectively, when the threshold is x, and their expressions are given in (7) and (8), respectively. We can easily verify via basic calculus that given any system parameters a, θ, τ, p_L, p_E and γ, $T(x|\gamma)$ is continuous with respect to $x \geq 0$, and is differentiable at any non-integer value x. Fig. 8a and Fig. 8b show the function $T(x|\gamma = \sqrt{3/10})$ when the arrival intensity $\theta = 2$ and $\theta = 4$, respectively, where the function is continuous with respect to x and differentiable at non-integer values in both cases.



(a) Arrival intensity $\theta = 2$ (b) Arrival intensity $\theta = 4$

Fig. 8: Cost function $T(x|\gamma = \sqrt{3/10})$ when $\tau = 1, p_L = 3$, $p_E = 1$ and $w = 1$.

Next, we would like to study the monotonicity of the cost function $T(x|\gamma)$ with respect to x. We first take the derivative of $T(x|\gamma)$ with respect to x when $l - 1 < x < l$ for some $l = 1, 2, \cdots$, i.e.,

$$T'(x|\gamma) = \frac{\theta^{l-1}\left(f(l|\theta) - a(g(\gamma) + \tau + w(p_E - p_L))\right)}{a\left(\sum_{j=0}^{l-1}\theta^j + (x - l + 1)\theta^l\right)^2}.$$

Noting that $\frac{\theta^{l-1}}{a\left(\sum_{j=0}^{l-1}\theta^j + (x-l+1)\theta^l\right)^2} > 0, \forall l = 1, 2, 3, \cdots$, we only need to consider the sign of $f(l|\theta) - a(g(\gamma) + \tau + w(p_E - p_L))$ in $T'(x|\gamma)$. Then, we consider the following two different cases:

(i) If $-w_{\max}A_{\max}P_{L,\max} < a(g(\gamma)+\tau+w(p_E-p_L)) < f(1|\theta)$, then we have $f(l|\theta)-a(g(\gamma)+\tau+w(p_E-p_L)) > 0, \forall l \in \mathbb{N}^+$ (recall that $\mathbb{N}^+$ denotes the set of natural numbers), since $f(l|\theta)$ is non-decreasing with respect to l. Therefore, we have $T'(x|\gamma) > 0, \forall x \in (l-1,l), \forall l \in \mathbb{N}^+$, which implies that $T(x|\gamma)$ is increasing when $x > 0$. Since $T(x|\gamma)$ is increasing and continuous, we have $x^*(\gamma; a, \theta, \tau, p_L, p_E) = 0$.

(ii) If $f(m|\theta) \le a(g(\gamma)+\tau+w(p_E-p_L)) < f(m+1|\theta)$ for some positive integer m, then by the monotone increasing property of function $f(l|\theta)$, we have $f(l|\theta) - a(g(\gamma)+\tau+w(p_E-p_L)) \le 0, \forall 0 \le l \le m$ and $f(l|\theta)-a(g(\gamma)+\tau+w(p_E-p_L)) > 0, \forall l > m$. Therefore, we have $T'(x|\gamma) \le 0, \forall x \le m$ and $T'(x|\gamma) > 0, \forall x > m$. Hence $T(x|\gamma)$ is non-increasing and non-decreasing when $x \le m$ and $x > m$, respectively. Thus, we have $x^*(\gamma; a, \theta, \tau, p_L, p_E) = m$ is the optimal threshold. ■

Note that $x^*(\gamma; a, \theta, \tau, p_L, p_E)$ is not necessarily unique. For example, given $a, \theta, k, \gamma, \tau, p_L, p_E$ and w, if $f(m|\theta) = a(g(\gamma)+\tau+w(p_E-p_L))$ for some positive integer m, then the optimal threshold could be any value between m and $m+1$, i.e., $x^*(\gamma; a, \theta, \tau, p_L, p_E) \in [m, m+1)$. As shown in Fig. 8a, the optimal threshold can be any value between 1 and 2.

APPENDIX C
PROOF OF LEMMA 2

In this section, we will show that $V(\gamma) = \mathbb{E}_{A,\Theta,T,P_L,P_E}\left[A\alpha(x^*(\gamma; A, \Theta, T, P_L, P_E))/c\right]$ is non-increasing and continuous with respect to $\gamma \in [0,1]$.

The non-increasing property of $V(\gamma)$ is proved by showing that for each individual sample path, $a\alpha(x^*(\gamma; a, \theta, \tau, p_L, p_E))/c$ is non-increasing with respect to γ given any arrival rate a, arrival intensity θ, offloading latency τ, local processing energy consumption p_L and offloading energy consumption p_E. Indeed, by Lemma 1, we have $x^*(\gamma; a, \theta, \tau, p_L, p_E) = 0$ if $-w_{\max}A_{\max}P_{L,\max} < a(g(\gamma) + \tau + w(p_E-p_L)) < f(0|\theta)$, and $x^*(\gamma; a, \theta, \tau, p_L, p_E) = m$ if $f(m|\theta) \le a(g(\gamma)+\tau+w(p_E-p_L)) < f(m+1|\theta)$. Hence, $x^*(\gamma; a, \theta, \tau, p_L, p_E)$ is non-decreasing with respect to γ. This together with the fact that $\alpha(x)$ is non-increasing with respect to x, implies that $a\alpha(x^*(\gamma; a, \theta, \tau, p_L, p_E))/c$ is non-increasing with respect to γ and hence $V(\gamma)$ is non-increasing.

Next, we show that $V(\gamma)$ is continuous with respect to γ. We will expand $V(\gamma)$ by leveraging Lemma 1. To facilitate our proof, we let $\mathcal{A} \triangleq (0, A_{\max}]$, $\mathcal{I} \triangleq (0, A_{\max}/S_{\min}]$, $\mathcal{T} \triangleq (0, T_{\max}]$, $\mathcal{P}_L \triangleq (0, P_{L,\max}]$ and $\mathcal{P}_E \triangleq (0, P_{E,\max}]$.

Note that we can rewrite $V(\gamma)$ as follows:

$$V(\gamma) = \mathbb{E}_{\Theta,T,P_L,P_E}\left[\widetilde{V}(\gamma|\theta,\tau,p_L,p_E)\right],$$

where

$$\widetilde{V}(\gamma|\theta,\tau,p_L,p_E) \triangleq \mathbb{E}_A[A\alpha(x^*(\gamma; A, \Theta, T, P_L, P_E))/c \\ |\Theta=\theta, T=\tau, P_L=p_L, P_E=p_E].$$

Note that $\forall \theta \in \mathcal{I}$ and $\forall \tau \in \mathcal{T}$, the function $\widetilde{V}(\gamma|\theta,\tau,p_L,p_E)$ is bounded:

$$\widetilde{V}(\gamma|\theta,\tau,p_L,p_E) \\ \le \mathbb{E}_A\left[A/c|\Theta=\theta, T=\tau, P_L=p_L, P_E=p_E\right] \le A_{\max}/c.$$

By the Bounded Convergence Theorem, it is sufficient to prove the continuity of function $\widetilde{V}(\gamma|\theta,\tau,p_L,p_E)$ with respect to $\gamma \in [0,1]$ for any given $\theta \in \mathcal{I}$, $\tau \in \mathcal{T}$, $p_L \in \mathcal{P}_L$ and $p_E \in \mathcal{P}_E$. In the rest of the proof, we fix $\theta \in \mathcal{I}$, $\tau \in \mathcal{T}$, $p_L \in \mathcal{P}_L$ and $p_E \in \mathcal{P}_E$. Noting that $f(0|\theta) = 0$ and $f(m|\theta)$ is non-decreasing with respect to m, and $f(m|\theta) \ge m\theta$ (from (10)), there exists a non-negative integer $M(\gamma)$ such that one of the following inequalities holds:

$$U(A_{\max}, \gamma, \tau, p_L, p_E) \in (-w_{\max}A_{\max}P_{L,\max}, f(0|\theta)), \\ U(A_{\max}, \gamma, \tau, p_L, p_E) \in [f(M(\gamma)|\theta), f(M(\gamma)+1|\theta)),$$

where $U(y, \gamma, \tau, p_L, p_E) \triangleq y(g(\gamma)+\tau+w(p_E-p_L))$ for some $y > 0$.

As such, we can partition the space of arrival rate $\mathcal{A}$ by defining the following events:

$$\mathcal{F}_{-1}(\gamma) \\ \triangleq \{a \in \mathcal{A} : -\omega_{\max}A_{\max}P_{L,\max} < U(a,\gamma,\tau,p_L,p_E) < f(0|\theta)\}, \\ \mathcal{F}_m(\gamma) \\ \triangleq \{a \in \mathcal{A} : f(m|\theta) \le U(a,\gamma,\tau,p_L,p_E) < f(m+1|\theta)\},$$

where $m = 0, 1, \cdots, M(\gamma)$.

If the event $\mathcal{F}_{-1}(\gamma)$ happens, then we have $\widetilde{V}(\gamma|\theta,\tau,p_L,p_E) = \widetilde{V}_{-1}(\gamma|\theta,\tau,p_L,p_E)$ according to Lemma 1, where

$$\widetilde{V}_{-1}(\gamma|\theta,\tau,p_L,p_E) \triangleq \int_0^{A_{\max}} \frac{ah(a|\theta)\alpha(0)}{c} da.$$

Note that $\widetilde{V}_{-1}(\gamma|\theta,\tau,p_L,p_E)$ is continuous since both a and s are sampled from continuous random variables A and S, respectively, and therefore $h(a|\theta)$ is a continuous conditional probability density function for A given $\Theta = \theta$ with $\theta = a/s$, which indicates that the integral is also continuous.

If the event $\cup_{m=0}^{M(\gamma)} \mathcal{F}_m(\gamma)$ happens, then we have

$$\widetilde{V}(\gamma|\theta,\tau,p_L,p_E) = \\ \mathbb{E}_A[\sum_{m=0}^{M(\gamma)} A\alpha(m)/c \cdot 1_{\mathcal{F}_m(\gamma)}|\Theta=\theta, T=\tau, P_L=p_L, P_E=p_E] \\ = \sum_{m=0}^{M(\gamma)} \widetilde{V}_m(\gamma|\theta,\tau,p_L,p_E), \tag{11}$$

where the first step follows from Lemma 1 (i.e., if the event $\mathcal{F}_m(\gamma)$ happens ($m = 0, 1, \cdots, M(\gamma)$), then the event $\mathcal{F}_{-1}(\gamma)$ can not happen, and $x^*(\gamma; a, \theta, \tau, p_L, p_E) = m$ according to Lemma 1.) and the second step is true for

$$\widetilde{V}_m(\gamma|\theta,\tau,p_L,p_E) \triangleq \int_{\frac{f(m|\theta)}{g(\gamma)+\tau+w(p_E-p_L)}}^{\frac{f(m+1|\theta)}{g(\gamma)+\tau+w(p_E-p_L)}} \frac{ah(a|\theta)\alpha(m)}{c} da,$$

when $m = 0, \cdots, M(\gamma)-1$, $h(a|\theta)$ is the conditional probability density function for A given $\Theta = \theta$, and

$$\widetilde{V}_{M(\gamma)}(\gamma|\theta,\tau,p_L,p_E) \triangleq \int_{\frac{f(M(\gamma)|\theta)}{g(\gamma)+\tau+w(p_E-p_L)}}^{A_{\max}} \frac{ah(a|\theta)\alpha(M(\gamma))}{c} da.$$

Note that if $g(\gamma)+\tau+w(p_E-p_L) \leq f(1|\theta)$ ($x^*(\gamma;a,\theta,\tau,p_L,p_E) = 0$ according to Lemma 1), then $\widetilde{V}(\gamma|\theta,\tau,p_L,p_E)$ is continuous, which follows directly from the same argument if event $\mathcal{F}_{-1}(\gamma)$ happens. Therefore, we need to consider the case when $g(\gamma)+\tau+w(p_E-p_L) > f(1|\theta)$. In order to prove the continuity of $\widetilde{V}(\gamma|\theta,\tau,p_L,p_E)$ when $g(\gamma)+\tau+w(p_E-p_L) > f(1|\theta)$, we need to show that $\lim_{\Delta\gamma\to 0}\widetilde{V}(\gamma+\Delta\gamma|\theta,\tau,p_L,p_E) = \widetilde{V}(\gamma|\theta,\tau,p_L,p_E)$. By the definition of $M(\gamma)$, $M(\gamma)$ is a non-negative integer M^* satisfying the following equality $\frac{f(M^*|\theta)}{g(\gamma)+\tau+w(p_E-p_L)} \leq A_{\max} < \frac{f(M^*+1|\theta)}{g(\gamma)+\tau+w(p_E-p_L)}$. We have two cases: (i) $\frac{f(M^*|\theta)}{g(\gamma)+\tau+w(p_E-p_L)} < A_{\max} < \frac{f(M^*+1|\theta)}{g(\gamma)+\tau+w(p_E-p_L)}$; (ii) $\frac{f(M^*|\theta)}{g(\gamma)+\tau+w(p_E-p_L)} = A_{\max}$.

Case (i): $\frac{f(M^*|\theta)}{g(\gamma)+\tau+w(p_E-p_L)} < A_{\max} < \frac{f(M^*+1|\theta)}{g(\gamma)+\tau+w(p_E-p_L)}$. We will first show that there exists a small $\delta > 0$ such that for any $|\Delta\gamma| < \delta$, $M(\gamma+\Delta\gamma) = M(\gamma) = M^*$. Define $\epsilon_1 \triangleq A_{\max} - \frac{f(M^*|\theta)}{g(\gamma)+\tau+w(p_E-p_L)}$ and $\epsilon_2 \triangleq \frac{f(M^*+1|\theta)}{g(\gamma)+\tau+w(p_E-p_L)} - A_{\max}$. Note that both ϵ_1 and ϵ_2 are strictly positive. By the continuity of function $\frac{1}{g(\gamma)+\tau+w(p_E-p_L)}$ with respect to γ, for a given $\epsilon_1 > 0$, there exists $\delta_1 > 0$ such that for any $|\Delta\gamma| < \delta_1$, we have $\left|\frac{f(M^*|\theta)}{g(\gamma+\Delta\gamma)+\tau+w(p_E-p_L)} - \frac{f(M^*|\theta)}{g(\gamma)+\tau+w(p_E-p_L)}\right| < \epsilon_1/2$. Similarly, there exists $\delta_2 > 0$, such that for any $|\Delta\gamma| < \delta_2$, we have $\left|\frac{f(M^*+1|\theta)}{g(\gamma+\Delta\gamma)+\tau+w(p_E-p_L)} - \frac{f(M^*+1|\theta)}{g(\gamma)+\tau+w(p_E-p_L)}\right| < \epsilon_2/2$. Let $\delta = \min\{\delta_1,\delta_2\}$. For each $\Delta\gamma \in [0,\delta)$, by the increasing property of $g(\cdot)$ with respect to γ, we have

$$\begin{aligned}
&\frac{f(M^*|\theta)}{g(\gamma+\Delta\gamma)+\tau+w(p_E-p_L)}\\
\leq&\frac{f(M^*|\theta)}{g(\gamma)+\tau+w(p_E-p_L)} < A_{\max},\\
&\frac{f(M^*+1|\theta)}{g(\gamma+\Delta\gamma)+\tau+w(p_E-p_L)}\\
\geq&\frac{f(M^*+1|\theta)}{g(\gamma)+\tau+w(p_E-p_L)} - \epsilon_2/2 > A_{\max},
\end{aligned}$$

implying $M(\gamma+\Delta\gamma) = M^* = M(\gamma)$. Similarly, for each $\Delta\gamma \in (-\delta, 0]$, we have

$$\begin{aligned}
&\frac{f(M^*|\theta)}{g(\gamma+\Delta\gamma)+\tau+w(p_E-p_L)}\\
\leq&\ \frac{f(M^*|\theta)}{g(\gamma)+\tau+w(p_E-p_L)} + \epsilon_1/2 < A_{\max},\\
&\frac{f(M^*+1|\theta)}{g(\gamma+\Delta\gamma)+\tau+w(p_E-p_L)}\\
\geq&\ \frac{f(M^*+1|\theta)}{g(\gamma)+\tau+w(p_E-p_L)} > A_{\max},
\end{aligned}$$

implying $M(\gamma+\Delta\gamma) = M^* = M(\gamma)$. Therefore, for $|\Delta\gamma| < \delta$, we have $M(\gamma+\Delta\gamma) = M^* = M(\gamma)$. From Eq. (11), we have

$$\begin{aligned}
&\widetilde{V}(\gamma+\Delta\gamma|\theta,\tau,p_L,p_E) - \widetilde{V}(\gamma|\theta,\tau,p_L,p_E)\\
&=\sum_{m=0}^{M(\gamma)}\left[\widetilde{V}_m(\gamma+\Delta\gamma|\theta,\tau,p_L,p_E) - \widetilde{V}_m(\gamma|\theta,\tau,p_L,p_E)\right].
\end{aligned}$$

For each $m \in \{1,\ldots,M(\gamma)-1\}$, we have

$$\begin{aligned}
&\lim_{\Delta\gamma\to 0}\widetilde{V}_m(\gamma+\Delta\gamma|\theta,\tau,p_L,p_E)\\
=&\lim_{\Delta\gamma\to 0}\int_{\frac{f(m|\theta)}{g(\gamma+\Delta\gamma)+\tau+w(p_E-p_L)}}^{\frac{f(m+1|\theta)}{g(\gamma+\Delta\gamma)+\tau+w(p_E-p_L)}}\frac{ah(a|\theta)\alpha(m)}{c}da\\
=&\int_{\frac{f(m|\theta)}{g(\gamma)+\tau+w(p_E-p_L)}}^{\frac{f(m+1|\theta)}{g(\gamma)+\tau+w(p_E-p_L)}}\frac{ah(a|\theta)\alpha(m)}{c}da\\
=&\widetilde{V}_m(\gamma|\theta,\tau,p_L,p_E).
\end{aligned}$$

Additionally, we have

$$\begin{aligned}
&\lim_{\Delta\gamma\to 0}\widetilde{V}_{M(\gamma)}(\gamma+\Delta\gamma|\theta,\tau,p_L,p_E)\\
=&\lim_{\Delta\gamma\to 0}\int_{\frac{f(M(\gamma)|\theta)}{g(\gamma+\Delta\gamma)+\tau+w(p_E-p_L)}}^{A_{\max}}\frac{ah(a|\theta)\alpha(m)}{c}da\\
=&\int_{\frac{f(M(\gamma)|\theta)}{g(\gamma)+\tau+w(p_E-p_L)}}^{A_{\max}}\frac{ah(a|\theta)\alpha(m)}{c}da\\
=&\widetilde{V}_{M(\gamma)}(\gamma|\theta,\tau,p_L,p_E).
\end{aligned}$$

Therefore, we have $\lim_{\Delta\gamma\to 0}\widetilde{V}(\gamma+\Delta\gamma|\theta,\tau,p_L,p_E) = \widetilde{V}(\gamma|\theta,\tau,p_L,p_E)$, which implies the continuity of function function $V(\gamma)$.

Case (ii): $\frac{f(M^*|\theta)}{g(\gamma)+\tau+w(p_E-p_L)} = A_{\max}$. Note that $A_{\max} > 0$, so $M^* > 0$. Let $\epsilon_3 \triangleq \frac{f(M^*+1|\theta)}{g(\gamma)+\tau+w(p_E-p_L)} - \frac{f(M^*|\theta)}{g(\gamma)+\tau+w(p_E-p_L)}$ and $\epsilon_4 \triangleq \frac{f(M^*|\theta)}{g(\gamma)+\tau+w(p_E-p_L)} - \frac{f(M^*-1|\theta)}{g(\gamma)+\tau+w(p_E-p_L)}$. Again, by the continuity of function $\frac{1}{g(\gamma)+\tau+w(p_E-p_L)}$, there exists $\delta_3 > 0$, such that $\left|\frac{f(M^*+1|\theta)}{g(\gamma+\Delta\gamma)+\tau+w(p_E-p_L)} - \frac{f(M^*+1|\theta)}{g(\gamma)+\tau+w(p_E-p_L)}\right| < \epsilon_3/2$ holds for all $|\Delta\gamma| < \delta_3$. Similarly, there exists $\delta_4 > 0$, such that $\left|\frac{f(M^*-1|\theta)}{g(\gamma+\Delta\gamma)+\tau+w(p_E-p_L)} - \frac{f(M^*-1|\theta)}{g(\gamma)+\tau+w(p_E-p_L)}\right| < \epsilon_4/2$ holds for all $|\Delta\gamma| < \delta_4$. Let $\delta = \min\{\delta_3,\delta_4\}$.

For each $\Delta\gamma \in [0,\delta)$, we have

$$\begin{aligned}
&\frac{f(M^*|\theta)}{g(\gamma+\Delta\gamma)+\tau+w(p_E-p_L)}\\
\leq&\frac{f(M^*|\theta)}{g(\gamma)+\tau+w(p_E-p_L)} = A_{\max},\\
&\frac{f(M^*+1|\theta)}{g(\gamma+\Delta\gamma)+\tau+w(p_E-p_L)}\\
\geq&\frac{f(M^*+1|\theta)}{g(\gamma)+\tau+w(p_E-p_L)} - \epsilon_3/2 > A_{\max},
\end{aligned}$$

implying $M(\gamma+\Delta\gamma) = M^* = M(\gamma)$. Following the same line of argument for case (i), we have

$$\lim_{\Delta\gamma\downarrow 0}\widetilde{V}(\gamma+\Delta\gamma|\theta,\tau,p_L,p_E) = \widetilde{V}(\gamma|\theta,\tau,p_L,p_E). \quad (12)$$

For each $\Delta\gamma \in (-\delta, 0)$, we have

$$\begin{aligned}
&\frac{f(M^*|\theta)}{g(\gamma+\Delta\gamma)+\tau+w(p_E-p_L)} \\
>&\frac{f(M^*|\theta)}{g(\gamma)+\tau+w(p_E-p_L)} = A_{\max}, \\
&\frac{f(M^*-1|\theta)}{g(\gamma+\Delta\gamma)+\tau+w(p_E-p_L)} \\
\le&\frac{f(M^*-1|\theta)}{g(\gamma)+\tau+w(p_E-p_L)} + \epsilon_4/2 < A_{\max},
\end{aligned}$$

implying $M(\gamma+\Delta\gamma) = M^*-1 = M(\gamma)-1$. Note that

$$\begin{aligned}
&\widetilde{V}_{M(\gamma)}(\gamma|\theta,\tau,p_L,p_E) \\
=&\int_{\frac{f(M(\gamma)|\theta)}{g(\gamma)+\tau+w(p_E-p_L)}}^{A_{\max}} \frac{ah(a|\theta)\alpha(M(\gamma))}{c} da = 0,
\end{aligned}$$

where we utilize our condition that $\frac{f(M^*|\theta)}{g(\gamma)+\tau+w(p_E-p_L)} = A_{\max}$, and $M^* = M(\gamma)$. From Eq. (11), we then have

$$\begin{aligned}
&\widetilde{V}(\gamma+\Delta\gamma|\theta,\tau,p_L,p_E) - \widetilde{V}(\gamma|\theta,\tau,p_L,p_E) \\
=&\sum_{m=0}^{M(\gamma)-1}\left[\widetilde{V}_m(\gamma+\Delta\gamma|\theta,\tau,p_L,p_E) - \widetilde{V}_m(\gamma|\theta,\tau,p_L,p_E)\right].
\end{aligned}$$

Following the same line of argument for case (i), for each $m \in \{0, 1, \ldots, M(\gamma)-2\}$, we have

$$\lim_{\Delta\gamma\uparrow 0} \widetilde{V}_m(\gamma+\Delta\gamma|\theta,\tau,p_L,p_E) = \widetilde{V}_m(\gamma|\theta,\tau,p_L,p_E).$$

Additionally, we have

$$\begin{aligned}
&\lim_{\Delta\gamma\uparrow 0} \widetilde{V}_{M(\gamma)-1}(\gamma+\Delta\gamma|\theta,\tau,p_L,p_E) \\
=&\lim_{\Delta\gamma\uparrow 0} \int_{\frac{f(M(\gamma)-1|\theta)}{g(\gamma+\Delta\gamma)+\tau+w(p_E-p_L)}}^{A_{\max}} \frac{ah(a|\theta)\alpha(m)}{c} da \\
=&\int_{\frac{f(M(\gamma)-1|\theta)}{g(\gamma)+\tau+w(p_E-p_L)}}^{A_{\max}} \frac{ah(a|\theta)\alpha(m)}{c} da \\
=&\widetilde{V}_{M(\gamma)-1}(\gamma|\theta,\tau,p_L,p_E).
\end{aligned}$$

Together, we have

$$\lim_{\Delta\gamma\uparrow 0} \widetilde{V}(\gamma+\Delta\gamma|\theta,\tau,p_L,p_E) = \widetilde{V}(\gamma|\theta,\tau,p_L,p_E). \quad (13)$$

Equations (12) - (13) imply that $\lim_{\Delta\gamma\to 0} \widetilde{V}(\gamma+\Delta\gamma|\theta,\tau,p_L,p_E) = \widetilde{V}(\gamma|\theta,\tau,p_L,p_E)$.

We complete the proof for the continuity of function $\widetilde{V}(\gamma|\theta,\tau,p_L,p_E)$, which implies the continuity of function function $V(\gamma)$. ■

Appendix D
Proof of Theorem 2

Under our proposed DTU Algorithm, when the "estimated" server utilization $\widehat{\gamma}_t$ reaches the MFNE γ^*, then it will stay at γ^* afterward. As such, we focus on the case when $\widehat{\gamma}_t \neq \gamma^*$ in the rest of the proof. Next, we exhibit the monotone properties of $\widehat{\gamma}_t$ in two different cases, i.e., $\widehat{\gamma}_t < \gamma^*$ and $\widehat{\gamma}_t > \gamma^*$. Then, by comparing $|\widehat{\gamma}_t - \gamma^*|$ with a convergent sequence $\{\eta_t\}$ in our proposed DTU Algorithm, we obtain the desired results.

We recall that $x^*(\gamma^*; a, \theta, \tau, p_L, p_E)$ is the optimal threshold for the system with the arrival rate a, the arrival intensity θ, local processing energy consumption p_L, offloading energy consumption p_E, the mean offloading latency τ, and the server utilization γ^*. We consider two different cases, i.e., (i) $\widehat{\gamma}_t > \gamma^*$; (ii) $\widehat{\gamma}_t < \gamma^*$.

(i) If $\widehat{\gamma}_t > \gamma^*$, then we have $a(g(\widehat{\gamma}_t) + \tau + w(p_E - p_L)) > a(g(\gamma^*) + \tau + w(p_E - p_L))$. By Lemma 1 and our DTU Algorithm (cf. (5)), we have $\widehat{x}^{(t+1)} \ge x^*(\gamma^*; a, \theta, \tau, p_L, p_E)$. Therefore, by the non-increasing property of offloading probability $\alpha(x)$, we have $\alpha(\widehat{x}^{(t+1)}) \le \alpha(x^*(\gamma^*; a, \theta, \tau, p_L, p_E))$, which implies that the next server utilization $\gamma_{t+1} \le \gamma^*$ and hence $\gamma_{t+1} < \widehat{\gamma}_t$. Thus, we have $\widehat{\gamma}_{t+1} = \widehat{\gamma}_t - \eta_t$ (cf. (4)). In this case, $\widehat{\gamma}_t$ will decrease in each iteration until it is less than γ^*. Therefore, there exists a $t_1 > 0$ such that $\widehat{\gamma}_{t+t_1} < \gamma^*$.

(ii) If $\widehat{\gamma}_t < \gamma^*$, then we have $a(g(\widehat{\gamma}_t) + \tau + w(p_E - p_L)) < a(g(\gamma^*) + \tau + w(p_E - p_L))$. Similar to the first case, we have $\widehat{x}^{(t+1)} \le x^*(\gamma^*; a, \theta, \tau, p_L, p_E)$, which implies that $\alpha(\widehat{x}^{(t+1)}) \ge \alpha(x^*(\gamma^*; a, \theta, \tau, p_L, p_E))$. Then, we have $\gamma_{t+1} \ge \gamma^*$ and hence $\gamma_{t+1} > \widehat{\gamma}_t$. Then according to (4), we have $\widehat{\gamma}_{t+1} = \min\{\widehat{\gamma}_t + \eta_t, 1\}$. In this case, $\widehat{\gamma}_t$ will increase after this iteration until it is greater than γ^*. Therefore, there exists a $t_2 > 0$ such that $\gamma_{t+t_2} > \gamma^*$.

Hence, under our proposed DTU Algorithm, we have

$$\widehat{\gamma}_{t+1} = \begin{cases} \widehat{\gamma}_t - \eta_t, & \text{if } \widehat{\gamma}_t > \gamma^*; \\ \min\{\widehat{\gamma}_t + \eta_t, 1\}, & \text{if } \widehat{\gamma}_t < \gamma^*. \end{cases}$$

Moreover, we have $|\widehat{\gamma}_{t+t_i} - \gamma^*| < |\widehat{\gamma}_{t+t_i-1} - \widehat{\gamma}_{t+t_i}| \le \eta_{t+t_i}, \forall i = 1, 2$. Since sequence $\{\eta_t\}$ is non-increasing and bounded, then by Monotone Convergence Theorem, $\{\eta_t\}$ is convergent. Indeed, we will show below that $\eta_t \to 0$ as $t \to \infty$ by contradiction.

Suppose that $\{\eta_t\}$ converges to a constant $d > 0$. According to the update rule of η_t, there exists a constant t_0 such that $\forall t \ge t_0$, $\eta_t = d$. Let us focus on $t \ge t_0$ and consider the following two cases: (i) $\widehat{\gamma}_t > \gamma^*$; (ii) $\widehat{\gamma}_t < \gamma^*$.

(i) If $\widehat{\gamma}_t > \gamma^*$: Note that $\widehat{\gamma}_{t+t_1} < \gamma^*$, so we have $\widehat{\gamma}_{t+t_1+1} = \min\{\widehat{\gamma}_{t+t_1} + d, 1\}$. Suppose that $\widehat{\gamma}_{t+t_1} + d > 1$. Since $\widehat{\gamma}_{t+t_1} = \widehat{\gamma}_{t+t_1-1} - d$, we have $\widehat{\gamma}_{t+t_1-1} = \widehat{\gamma}_{t+t_1} + d > 1$, which contracts with the fact that $\widehat{\gamma}_{t+t_1-1} \le 1$. Thus $\widehat{\gamma}_{t+t_1} + d > 1$ does not hold. Therefore, $\widehat{\gamma}_{t+t_1} + d \le 1$, and thus we have $\widehat{\gamma}_{t+t_1+1} = \widehat{\gamma}_{t+t_1} + d = \widehat{\gamma}_{t+t_1-1} - d + d = \widehat{\gamma}_{t+t_1-1}$. According to the update rule for the counter L (line 5 - 6 in DTU algorithm), L is increased by 1 at $t+t_1+1$. Thus η_{t+t_1+1} is updated with a value smaller than d, which contracts with the assumption that $\eta_t = d$ for all $t \ge t_0$.

(ii) If $\widehat{\gamma}_t < \gamma^*$: Recall that $\widehat{\gamma}_{t+t_2} = \min\{\widehat{\gamma}_{t+t_2-1} + d, 1\} > \gamma^*$, we have $\widehat{\gamma}_{t+t_2+1} = \widehat{\gamma}_{t+t_2} - d$. Suppose that $\widehat{\gamma}_{t+t_2-1} + d < 1$, we have

$$\widehat{\gamma}_{t+t_2+1} = \widehat{\gamma}_{t+t_2} - d = \widehat{\gamma}_{t+t_2-1} + d - d = \widehat{\gamma}_{t+t_2-1}.$$

Following the same line of argument as the case of (i), we can show that η_{t+t_2+1} is updated with a value smaller than d, which again contracts with our assumption of $\eta_t = d$ for all $t \geq t_0$.

Now let us consider the case $\widehat{\gamma}_{t+t_2-1} + d \geq 1$. We then have $\widehat{\gamma}_{t+t_2} = 1 > \gamma^*$. Thus

$$\widehat{\gamma}_{t+t_2+1} = \widehat{\gamma}_{t+t_2} - d = 1 - d < \gamma^*,$$

where the inequality follows from the fact that $\gamma^* > \widehat{\gamma}_{t+t_2-1} \geq 1 - d$. By the update rule, we have

$$\begin{aligned}\widehat{\gamma}_{t+t_2+2} &= \min\{\widehat{\gamma}_{t+t_2+1} + d, 1\} = \min\{1 - d + d, 1\} \\ &= 1 = \widehat{\gamma}_{t+t_2}.\end{aligned}$$

Again we reach the contradiction.

Therefore, the assumption that $\eta_t \to d$ with $d > 0$ does not hold. That is, $\eta_t \to 0$ as $t \to \infty$. Then, for any $\epsilon > 0$ there exists $\widetilde{t} > 0$ such that $\eta_{t+t_i+\widetilde{t}} < \epsilon$, which implies that $|\widehat{\gamma}_{t+t_i+\widetilde{t}} - \gamma^*| < \epsilon$. Therefore, our proposed DTU Algorithm can eventually converge to the MFNE γ^*. ■

References

[1] K. N. Swaroop, K. Chandu, R. Gorrepotu, and S. Deb, "A health monitoring system for vital signs using iot," *Internet of Things*, vol. 5, pp. 116–129, 2019.

[2] P. Verma and S. K. Sood, "Fog assisted-iot enabled patient health monitoring in smart homes," *IEEE Internet of Things Journal*, vol. 5, no. 3, pp. 1789–1796, 2018.

[3] A. Sengupta, S. S. Gill, A. Das, and D. De, "Mobile edge computing based internet of agricultural things: A systematic review and future directions," *Mobile Edge Computing*, pp. 415–441, 2021.

[4] B. Omoniwa, R. Hussain, M. A. Javed, S. H. Bouk, and S. A. Malik, "Fog/edge computing-based iot (feciot): Architecture, applications, and research issues," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4118–4149, 2018.

[5] L. Kong, J. Tan, J. Huang, G. Chen, S. Wang, X. Jin, P. Zeng, M. K. Khan, and S. K. Das, "Edge-computing-driven internet of things: A survey," *ACM Computing Surveys (CSUR)*, 2022.

[6] S. Guo, J. Liu, Y. Yang, B. Xiao, and Z. Li, "Energy-efficient dynamic computation offloading and cooperative task scheduling in mobile cloud computing," *IEEE Transactions on Mobile Computing*, vol. 18, no. 2, pp. 319–333, 2018.

[7] H. Wu, Y. Sun, and K. Wolter, "Energy-efficient decision making for mobile cloud offloading," *IEEE Transactions on Cloud Computing*, vol. 8, no. 2, pp. 570–584, 2018.

[8] X. Sun and N. Ansari, "Latency aware workload offloading in the cloudlet network," *IEEE Communications Letters*, vol. 21, no. 7, pp. 1481–1484, 2017.

[9] X. Hu, L. Wang, K.-K. Wong, M. Tao, Y. Zhang, and Z. Zheng, "Edge and central cloud computing: A perfect pairing for high energy efficiency and low-latency," *IEEE Transactions on Wireless Communications*, vol. 19, no. 2, pp. 1070–1083, 2019.

[10] X. Tao, K. Ota, M. Dong, H. Qi, and K. Li, "Performance guaranteed computation offloading for mobile-edge cloud computing," *IEEE Wireless Communications Letters*, vol. 6, no. 6, pp. 774–777, 2017.

[11] Y. Chen, N. Zhang, Y. Zhang, X. Chen, W. Wu, and X. S. Shen, "Energy efficient dynamic offloading in mobile edge computing for internet of things," *IEEE Transactions on Cloud Computing*, 2019.

[12] Q. Chen, X. Xu, H. Jiang, and X. Liu, "An energy-aware approach for industrial internet of things in 5g pervasive edge computing environment," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 7, pp. 5087–5097, 2020.

[13] H. Wu, X. Lyu, and H. Tian, "Online optimization of wireless powered mobile-edge computing for heterogeneous industrial internet of things," *IEEE Internet of Things Journal*, vol. 6, no. 6, pp. 9880–9892, 2019.

[14] R. Lin, T. Xie, S. Luo, X. Zhang, Y. Xiao, B. Moran, and M. Zukerman, "Energy-efficient computation offloading in collaborative edge computing," *IEEE Internet of Things Journal*, 2022.

[15] M. J. Neely, "Stochastic network optimization with application to communication and queueing systems," *Synthesis Lectures on Communication Networks*, vol. 3, no. 1, pp. 1–211, 2010.

[16] Z. Liao, J. Peng, J. Huang, J. Wang, J. Wang, P. K. Sharma, and U. Ghosh, "Distributed probabilistic offloading in edge computing for 6g-enabled massive internet of things," *IEEE Internet of Things Journal*, vol. 8, no. 7, pp. 5298–5308, 2020.

[17] Y. Wang, P. Lang, D. Tian, J. Zhou, X. Duan, Y. Cao, and D. Zhao, "A game-based computation offloading method in vehicular multiaccess edge computing networks," *IEEE Internet of Things Journal*, vol. 7, no. 6, pp. 4987–4996, 2020.

[18] M. Shifrin, R. Atar, and I. Cidon, "Optimal scheduling in the hybrid-cloud," in *2013 IFIP/IEEE International Symposium on Integrated Network Management (IM 2013)*. IEEE, 2013, pp. 51–59.

[19] W. Lin and P. Kumar, "Optimal control of a queueing system with two heterogeneous servers," *IEEE Transactions on Automatic control*, vol. 29, no. 8, pp. 696–703, 1984.

[20] X. Qin, B. Li, and L. Ying, "Distributed threshold-based offloading for large-scale mobile cloud computing," in *IEEE INFOCOM 2021-IEEE Conference on Computer Communications*. IEEE, 2021, pp. 1–10.

[21] S. Stidham, "Optimal control of admission to a queueing system," *IEEE Transactions on Automatic Control*, vol. 30, no. 8, pp. 705–713, 1985.

[22] Y. Wang, J. Yang, X. Guo, and Z. Qu, "A game-theoretic approach to computation offloading in satellite edge computing," *IEEE Access*, vol. 8, pp. 12 510–12 520, 2019.

[23] R. Dong, C. She, W. Hardjawana, Y. Li, and B. Vucetic, "Deep learning for hybrid 5g services in mobile edge computing systems: Learn from a digital twin," *IEEE Transactions on Wireless Communications*, vol. 18, no. 10, pp. 4692–4707, 2019.

[24] S.-W. Ko, K. Han, and K. Huang, "Wireless networks for mobile edge computing: Spatial modeling and latency analysis," *IEEE Transactions on Wireless Communications*, vol. 17, no. 8, pp. 5225–5240, 2018.

[25] L. Liu, Z. Chang, X. Guo, and T. Ristaniemi, "Multi-objective optimization for computation offloading in mobile-edge computing," in *2017 IEEE Symposium on Computers and Communications (ISCC)*. IEEE, 2017, pp. 832–837.

[26] C. Mouzouni, "On quasi-stationary mean field games models," *Applied Mathematics & Optimization*, vol. 81, no. 3, pp. 655–684, 2020.

[27] D. Narasimha, S. Shakkottai, and L. Ying, "Age-dependent distributed mac for ultra-dense wireless networks," *IEEE/ACM Transactions on Networking*, 2023.

[28] M. Huang, R. P. Malhamé, and P. E. Caines, "Large population stochastic dynamic games: closed-loop mckean-vlasov systems and the nash certainty equivalence principle," *Communications in Information & Systems*, vol. 6, no. 3, pp. 221–252, 2006.

[29] J.-M. Lasry and P.-L. Lions, "Mean field games," *Japanese journal of mathematics*, vol. 2, no. 1, pp. 229–260, 2007.

[30] J. Doncel, N. Gast, and B. Gaujal, "Discrete mean field games: Existence of equilibria and convergence," *arXiv preprint arXiv:1909.01209*, 2019.

[31] A. Cecchin, P. D. Pra, M. Fischer, and G. Pelino, "On the convergence problem in mean field games: a two state model without uniqueness," *SIAM Journal on Control and Optimization*, vol. 57, no. 4, pp. 2443–2466, 2019.

[32] M. Laurière and L. Tangpi, "Convergence of large population games to mean field games with interaction through the controls," *arXiv preprint arXiv:2004.08351*, 2020.

[33] Q. Xie, Z. Yang, Z. Wang, and A. Minca, "Learning while playing in mean-field games: Convergence and optimality," in *International Conference on Machine Learning*. PMLR, 2021, pp. 11 436–11 447.

[34] D. Narasimha, S. Shakkottai, and L. Ying, "A mean field game analysis of distributed mac in ultra-dense multichannel wireless networks," *IEEE/ACM Transactions on Networking*, vol. 28, no. 5, pp. 1939–1952, 2020.

[35] "YOLO: Real-Time Object Detection." [Online]. Available: https://pjreddie.com/darknet/yolo/

[36] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv*, 2018.

[37] "The PASCAL Visual Object Classes Challenge 2012 (VOC2012)." [Online]. Available: http://host.robots.ox.ac.uk/pascal/VOC/voc2012/index.html