Wireless Networks
https://doi.org/10.1007/s11276-019-02000-y

OKRA: optimal task and resource allocation for energy minimization in mobile edge computing systems

Weiwei Fang[1] · Shuai Ding[1] · Yangyang Li[2] · Wenchen Zhou[1] · Naixue Xiong[3]

Abstract
To cope with the computational and energy constraints of mobile devices, Mobile Edge Computing (MEC) has recently emerged as a new paradigm that provides IT and cloud-computing services at mobile network edge in close proximity to mobile devices. This paper investigates the energy consumption problem for mobile devices in a multi-user MEC system with different types of computation tasks, random task arrivals, and unpredictable channel conditions. By jointly considering computation task scheduling, CPU frequency scaling, transmit power allocation and subcarrier bandwidth assignment, we formulate it as a stochastic optimization problem aiming at minimizing the power consumption of mobile devices and to maintain the long-term stability of task queues. By leveraging the Lyapunov optimization technique, we propose an online control algorithm (OKRA) to solve the formulation. We prove that this algorithm is able to provide deterministic worst-case latency guarantee for latency-sensitive computation tasks, and balance a desirable tradeoff between power consumption and system stability by appropriately tuning the control parameter. Extensive simulations are carried out to verify the theoretical analysis, and illustrate the impacts of critical parameters to algorithm performance.

Keywords Mobile edge computing · Energy minimization · Resource and task allocation · Lyapunov optimization · Queue stability

1 Introduction

Mobile devices, such as wearables, tablets, and smartphones, have penetrated into our daily life as the most important tools for communication, information and entertainment. With the support of embedded sensors and cameras, new mobile applications with advanced features, e.g., online gaming, augmented reality, and object recognition, are becoming prevalent and attracting significant attention [5]. Such mobile applications usually demand intensive computation as well as tight latency [4, 32]. However, mobile devices normally have limited battery energy and constrained computation resources to support sophisticated applications. The resource scarceness thus poses an intractable challenge for the development of mobile platforms and the improvement of mobile service qualities [20].

Computation offloading is envisioned as a promising solution to cope with the above challenge, by migrating offloadable computation tasks from mobile devices to more powerful servers via wireless access [14]. One possible

✉ Weiwei Fang
wwfang@bjtu.edu.cn

Shuai Ding
15231213@bjtu.edu.cn

Yangyang Li
liyangyang@cetc.com.cn

Wenchen Zhou
16120465@bjtu.edu.cn

Naixue Xiong
naixuexiong@gmail.com

[1] School of Computer and Information Technology, Beijing Jiaotong University, Beijing, China

[2] National Engineering Laboratory for Public Safety Risk Perception and Control, China Academy of Electronics and Information Technology, Beijing, China

[3] Department of Mathematics and Computer Science, Northeastern State University, Tahlequah, USA

Published online: 11 May 2019

solution is to offload mobile tasks to remote resource-rich clouds [25], such as Amazon EC2 and Microsoft Azure [17, 23, 24]. However, the long propagation distance from mobile users to the remote cloud would bring about unpredictably high latency for mobile applications [22, 33]. This is especially problematic for a number of emerging applications that are sensitive to processing latency [26]. To address this limitation, cloudlet based computation offloading has been proposed as an alternative approach to powerful remote clouds [34]. A cloudlet is a resource-rich server or a cluster of servers that can be accessed by nearby mobile devices through one-hop Wi-Fi access. Because of the physical proximity, computation offloading to the cloudlet bypasses the uncontrollable high latency for data exchange through the wide area network. However, current Wi-Fi networks suffer from limited coverage, thus can not provide pervasive services to mobile users. Moreover, computation resources of the cloudlet may not adequate to meet QoS requirements of a large number of users. Very recently, mobile researchers propose a new paradigm called Mobile Edge Computing (MEC) [31], which aims to provide information processing and cloud computing services at the edge of mobile network, so as to offer a computation offloading environment characterized by close proximity, low latency, and high rate service. This paradigm enables telecom operators to deploy resource-rich cloud computing infrastructures of their own at the edge of cellular networks, allowing mobile users and applications to access pervasive and agile computation services when and where is needed [26, 43].

Although MEC technology promises attractive benefits, designing an energy-efficient MEC offloading policy for mobile devices still faces a number of challenging issues. Firstly, not all mobile application tasks are suitable for being offloaded to the MEC server. Some tasks have to be unconditionally executed on the mobile device, either because these tasks must access local components (e.g., sensor-relevant) or because these tasks might bring privacy problems when executed remotely (e.g., photo-relevant) [15, 40]. When a MEC offloading policy chooses local CPU resources, unoffloadable tasks would interfere with those offloadable ones since the CPU makes an attempt to process these two kinds of tasks simultaneously [15]. Such a problem is to be addressed in the design of dynamic resource allocation mechanisms. Secondly, to offload computation tasks for remote execution, a mobile device needs to deliver task inputs through the base station it is associated with to the MEC server. Generally, a base station provides communication services for a number of mobile users concurrently, and different users may have distinct demands on task offloading [42]. As a result, the MEC offloading policy has to resolve the potential contention on shared communication resources among mobile users. Thirdly, in real offloading scenarios, task arrivals and channel conditions are not static, but temporally dynamic [35]. The MEC offloading policy should be able to exploit such variations of system dynamics by jointly controlling CPU clock frequency and wireless transmit power to maximize energy efficiency of mobile devices under given constraints on processing latency [9, 36, 39]. As one extreme, if the task arrival rate is low and the communication condition is bad, then the offloading policy could prioritize local CPU resources for energy saving. As the other extreme, if the task arrival rate is high and the communication condition is good, then the offloading policy may prefer to offload computation to the MEC server by transmitting most offloadable tasks through the base station. Fourthly, computation tasks' arrival information is often unpredictable and even bursty. Meanwhile, the wireless channel state is also unknown and unpredictable to mobile devices. Thus, conventional deterministic control approaches, e.g., [39, 42], are not applicable to the problems stated above.

To tackle the aforementioned challenges, in this paper, we jointly consider multi-dimensional tasks and resource allocation, network stability and latency guarantee to formulate the power minimization problem for mobile devices in MEC systems. The problem is formulated as a stochastic optimization problem. It aims to minimize the power consumption of mobile devices subject to constraints on queue stability, task scheduling, CPU scaling, power allocation, subcarrier assignment, and worst-case latency. By leveraging the Lyapunov optimization theory [29, 30], we propose a new algorithm, referred to as OKRA (Optimal tasK and Resource Allocation), to solve the formulation. By exploiting the special structure of the subproblems in the OKRA design, we have designed simple yet optimal approaches to make online control decisions including computation task scheduling, CPU frequency scaling, transmit power allocation and subcarrier bandwidth assignment, all of which have closed-form solutions and don't require any iteration operations or optimization tools. Under the framework of Lyapunov optimization, OKRA can provide explicit performance bounds and approach the optimum with tunable deviation, without requiring pre-knowledge of system dynamics (e.g., task arrivals and channel conditions). In particular, OKRA is capable to ensure persistent service with bounded latency guarantee for latency-sensitive computation tasks. Rigorous mathematical analysis and extensive empirical evaluation are conducted to validate the effectiveness of OKRA in terms of power optimality, system stability and latency guarantee.

The rest of this paper is organized as follows. The review of related work is presented in Sect. 2. In Sect. 3, we describe system models. In Sect. 4, we formulate the

optimization problem by Lyapunov optimization, and propose our OKRA algorithm. The performance of OKRA is analyzed in Sect. 5. Section 6 illustrates simulation results and analysis. Finally, we conclude this paper in Sect. 7.

2 Related work

In this section, we briefly review existing work on resource management techniques for Mobile Edge Computing, and Lyapunov optimization theory for stochastic systems.

2.1 MEC resource management techniques

Resource management is very important in realizing low-latency and energy-efficient MEC systems, which is facilitated by the network architecture where the base station and the MEC server are co-located. We review existing studies on this issue in two categories, i.e., studies for a single-user MEC system and for a multi-user MEC system [41].

On one hand, some earlier studies aim at resource management for single-user MEC systems with only one dedicated MEC server. In [39], the computational speed, transmit power and offloading ratio are jointly optimized to achieve two different design objectives, i.e., minimizing energy consumption and minimizing execution latency, of the mobile device. The proposed problems are modeled in a deterministic form, and solved based on given system conditions that are known in advance. The authors of [21] use Markov Decision Process to solve the energy-constrained latency minimization problem in MEC, where computation tasks are scheduled based on the task queue size, local execution state and channel side information. Reference [27] investigates a green MEC system with energy harvesting mobile devices, and adopt the execution cost, which considers both execution latency and task failure, as the performance metric. A low-complexity algorithm is designed to control and optimize energy harvesting and computation offloading, so as to minimize time-average execution cost and avoid battery energy outage. On the other hand, a number of studies consider the multi-user MEC system comprising more than one mobile device that may share a MEC server or server-cluster. Reference [42] proposes scheduling algorithms of both radio and computational resources for minimizing energy consumption of mobile devices in an MEC system based on orthogonal frequency-division multiple access(OFDMA). However, the full offloading strategy adopted in [42] has been revealed by [39] as inefficient and costly. Reference [41] designs centralized resource management schemes to achieve the mimimal weighted sum energy consumption in the multi-user MEC system based on TDMA and OFDMA. The authors assume that the MEC server has perfect prior knowledge of local computation energy consumption, multi-user channel gains and input data sizes of all users, which is costly or impractical in most cases. A moving MEC system based on unmanned aerial vehicle (UAV) is considered in [13]. The bit allocation for uplink/downlink communication and for server computation, as well as the UAV's path planning, are jointly optimized to minimize the mobile energy consumption under the constraints on the UAV's mobility limitation and battery capacity. With the expectation of small cell base station being densely deployed in future cellular networks, a mobile device will have more communication and computation resources from engaging the help of multiple MEC servers [6], while MEC servers can further assist each other on caching and processing capabilities to satisfy mobile users' customized task requests [37].

Due to space limitations, interested readers are suggested to refer to [1, 26] for complete reviewing of relevant studies on MEC research.

2.2 Lyapunov optimization theory

Lyapunov optimization [29, 30] has received growing interest in solving problems of joint performance optimization and system stability in communication networks and stochastic systems. In order to optimize a certain time-average objective, the Lyapunov optimization algorithm is developed to make control operations that greedily minimize a bound on the so-called drift-plus-penalty expression over fixed-length time periods. Lyapunov optimization does not rely on statistics or prediction on stochastic models, but instead the information of queue backlogs, for making optimal control decisions in an online manner. This makes it different from the traditional techniques such as Markov Decision Process and Dynamic Programming, which inevitably incur the "curse of dimensionality" problem where the computation complexity for obtaining the optimal result rises with the system size [29, 38].

Heretofore, Lyapunov optimization has been extensively adopted for solving the task and resource management problems in several areas, e.g., cloud computing [11, 18], mobile system [10, 11, 28], wireless communication [19, 30], and smart grid [38]. Among them, our work is mainly motivated by recent studies [28] and [19]. Reference [28] investigates the power-latency tradeoff in a MEC system, and proposes an algorithm to determine the optimal policy on local execution and remote offloading for power consumption minimization. This work assumes that all kinds of mobile computation tasks are offloadable, and they can tolerant unbounded and even excessive execution

latency. However, these assumptions are usually unreasonable in practice [26, 34]. Reference [19] addresses a latency-aware and energy-aware transmission problem in heterogeneous wireless networks by jointly optimizing subcarrier assignment, power allocation, and time fraction determination. However, only the power consumption on wireless transmission is taken into consideration in [19].

3 System models

In this paper, we consider a multi-user MEC system in which a number of $\mathcal{N} = \{1, 2, ..., N\}$ mobile devices (MDs) are served by one MEC server. This MEC server could be a small-scale data center collocated with a cellular base station deployed by some telecom operator [4]. Thus, it can be accessed by all associated MDs through wireless channels, and would execute offloaded computation tasks from these MDs. We assume that time is divided into slots $t = \{0, 1, 2, ...\}$ with equal length, and the slot length is denoted as τ.

3.1 Task arrival model

Generally, we classify mobile computation tasks into two types, i.e., unoffloadable and offloadable [40]. Unoffloadable tasks are executed using local CPU resources of the MD, while offloadable tasks could be either executed locally by a MD's CPU or offloaded to the MEC server. In every time slot t, computation tasks arrive at each MD. We denote the number of unoffloadable and offloadable tasks arriving at MD n by $W_{u,n}(t)$ and $W_{o,n}(t)$, respectively. We assume that the task arrival process is independent and identically distributed (i.i.d) in each time slot [11]. It is also independent of the current number of unprocessed tasks in the system. Besides, we assume that there exists some $W_{u,n}^{max}$ and $W_{o,n}^{max}$ such that $W_{u,n}(t) \in [0, W_{u,n}^{max}]$ and $W_{o,n}(t) \in [0, W_{o,n}^{max}]$ for all n and t.

3.2 Local processing model

Typically, modern CPUs have the Dynamic Voltage and Frequency Scaling (DVFS) capability [39]. Thus, MD n is able to adjust its CPU clock speed $f_n(t)$ (in cycles/s) in each time slot t, where $f_n(t) \in [0, f_n^{max}]$. The computation task (in bits) requires a certain number of CPU processing resources per bit [21, 28], which is denoted by γ_n (in cycles/bit). Then, the total number of locally executed tasks at MD n can be expressed as

$$C_n(t) = \tau f_n(t) \gamma_n^{-1} \tag{1}$$

According to existing studies, the power consumption model for CPU of MD n [16] is known as follows:

$$P_{c,n}(t) = \alpha_1 f_n(t)^x + \alpha_2 \tag{2}$$

where α_1 and α_2 are the empirical coefficients of power consumption, while x ranges from 2 to 3 [15].

3.3 Computation offloading model

Compared with local processing, computation offloading saves computation and energy resources for MDs, but will incur additional time and energy in network communication. For simplicity, it is assumed that the MEC server has unconstrained computation resources, and the processing latency at the MEC server is negligible [28].

We consider a mobile system based on the OFDMA technique, e.g., the 3GPP LTE. Here, we use the same channel model as the one in [19]. There are in total S subcarriers, each of which may be shared by multiple devices in a time-division manner. The total bandwidth is denoted by B, so each subcarrier has a portion of bandwidth $B_s = B/S$. Let $\boldsymbol{P}(t) \triangleq (P_{s,n}(t))$, where $P_{s,n}(t) \in [0, P_{s,n}^{max}]$ denote the wireless transmit power of MD n on subcarrier s, and $\boldsymbol{\chi}(t) \triangleq \chi_{s,n}(t) \geq 0$ denote the time-sharing factor of MD n on subcarrier s. The transmit rate (in bits/s) of MD n on subcarrier s in time slot t is formulated as

$$r_{s,n}(t) = \begin{cases} \chi_{s,n}(t) B_s \log_2 \left(1 + \dfrac{P_{s,n}(t) g_{s,n}(t)}{\chi_{s,n}(t)}\right), & \chi_{s,n}(t) > 0 \\ 0, & \chi_{s,n}(t) = 0 \end{cases} \tag{3}$$

where $g_{s,n}(t) = \frac{|h_{s,n}(t)|^2}{N_0 B_s}$ is the channel gain, $h_{s,n}(t)$ is the frequency response of user n on subcarrier s, and N_0 is the single-sided spectral density of the additive white Gaussian noise, respectively [19]. Besides, for all s and t, we have $\sum_{n=1}^{N} \chi_{s,n}(t) \leq 1$. We assume that there exists certain finite constants $g_{s,n}^{max}$ and $r_{s,n}^{max}$, such that $g_{s,n}(t) \in [0, g_{s,n}^{max}]$ and $r_{s,n}(t) \in [0, r_{s,n}^{max}]$ for all subcarrier s, MD n, and time slot t [19].

Accordingly, the transmit rate from MD n to the MEC server over all subcarriers is calculated as

$$R_n(t) = \sum_{s=1}^{S} r_{s,n}(t) \tag{4}$$

3.4 Task queue model

Based on the task type (i.e., unoffloadable/offloadable), task requests arriving in MD n but not yet processed are distinguished and queued separately in two different buffer queues, $Q_{u,n}$ and $Q_{o,n}$. The buffer queues are maintained in memory of MD [11, 19, 21], and the queued tasks are processed as per First Come First Served principle [29]. To characterize the queue dynamics, we define $\boldsymbol{Q}(t) \triangleq ((Q_{u,1}(t), Q_{o,1}(t)), (Q_{u,2}(t), Q_{o,2}(t)), ..., (Q_{u,N}(t), Q_{o,N}(t)))$ as queue backlogs at the start of the time slot t, and $\boldsymbol{Q}(0) = \boldsymbol{0}$. Then, the queuing dynamics [29] can be characterized by

$$Q_{u,n}(t+1) = \max\{Q_{u,n}(t) - C_{u,n}(t), 0\} + W_{u,n}(t) \tag{5}$$

$$Q_{o,n}(t+1) = \max\{Q_{o,n}(t) - C_{o,n}(t) - M_{o,n}(t), 0\} + W_{o,n}(t) \tag{6}$$

where $C_{u,n}(t) \in [0, C_n(t)]$ is unoffloadable tasks processed locally at MD n in time slot t, $C_{o,n}(t) = C_n(t) - C_{u,n}(t)$ is the portion of offloadable tasks processed locally at MD n in t, and $M_{o,n}(t) = \tau R_n(t)$ is the other portion of offloadable tasks processed remotely at the MEC server in t. Accordingly, the queue stability constraint, which guarantees that the queue length is finite, can be formally defined. Throughout this paper, the queue stability is defined as follows:

$$\overline{Q} = \lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} \sum_{n=1}^{N} \mathbb{E}\{Q_{u,n}(t) + Q_{o,n}(t)\} < \infty \tag{7}$$

In addition, a worst-case latency constraint is taken into account for offloadable computation tasks. Worst-case latency, denoted as D_n^{max} in this paper, is the maximal time that an offloadable task experiences in the queue before it is scheduled [30]. This constraint is used to provide good experience for mobile users with MDs running computation intensive tasks [2, 42]. As other work using Lyapunov optimization [10, 11, 15, 28], in this paper we don't take into account the communication latency due to the complexity it would bring to our problem [7, 8].

4 Optimal task and resource allocation algorithm

In this section, a stochastic optimization programming is firstly formulated to investigate the power minimization problem in the given multi-user MEC system, subject to resource/task allocation constraints and a bounded latency requirement. Then, an online control algorithm framework, referred to as OKRA, is developed to solve this problem by adopting the Lyapunov optimization theory.

4.1 Problem formulation

In time slot t, the total power consumption of N MDs, including the computing power consumed by MDs' CPUs and the transmit power for task offloading, is given by the following:

$$P(t) = \sum_{n=1}^{N} \left[P_{c,n}(t) + \sum_{s=1}^{S} P_{s,n}(t) \right] \tag{8}$$

In this paper, we are interested in making control decisions on computation task scheduling, CPU frequency scaling, transmit power allocation, and subcarrier bandwidth assignment, so as to minimize the long-term time-average expected power consumption while serving all task arrivals within the capacity region. Based on the models above, our problem could be formulated as a stochastic program ***P1***:

$$\boldsymbol{P1:} \quad \min \quad \overline{P} \triangleq \lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T} \mathbb{E}\{P(t)\} \tag{9}$$

$$\text{s.t.} \quad \overline{Q} < \infty \tag{10}$$

$$C_{o,n}(t) + C_{u,n}(t) = C_n(t), \quad C_{o,n}(t), C_{u,n}(t) \geq 0 \tag{11}$$

$$0 \leq f_n(t) \leq f_n^{max} \tag{12}$$

$$0 \leq P_{s,n}(t) \leq P_{s,n}^{max} \tag{13}$$

$$0 \leq \chi_{s,n}(t) \leq 1 \tag{14}$$

$$\sum_{n=1}^{N} \chi_{s,n}(t) \leq 1 \tag{15}$$

$$\begin{aligned} & D_n^{max} < \infty \\ \text{var.} \quad & C_{o,n}(t), f_n(t), P_{s,n}(t), \chi_{s,n}(t) \end{aligned} \tag{16}$$

In problem ***P1***, constraint (10) guarantees that all arrived tasks could leave the queue within a finite time, so as to maintain system queue stability. Constraints (11)–(15) describe the feasible region for each decision variable. Constraint (16) ensures that the worst-case latency for offloadable tasks is finite.

The challenge here is how to link a deterministic latency requirement to the control decision variables. To handle this issue, we apply ϵ-persistent service queue technique [30] to establish the relationship between the queue occupancy and the worst-case latency. Let Z_n denote a latency-aware virtual queue associated with $Q_{o,n}(t)$, with $Z_n(0) = 0$. The virtual queue $Z_n(t)$ is updated according to

$$Z_n(t+1) = \max\{Z_n(t) - C_{o,n}(t) - M_{o,n}(t) + \epsilon_n 1_{\{Q_{o,n}(t) > 0\}}, 0\} \tag{17}$$

where $1_{\{Q_{o,n}(t) > 0\}}$ is an indicator function which takes the value 1 if $Q_{o,n}(t) > 0$, and 0 otherwise. The intuition is that $Z_n(t)$ has the same service process as $Q_{o,n}(t)$, but has a

different growing process that adds ϵ_n whenever queue $Q_{o,n}$ is not empty. The parameter ϵ_n controls the growing rate of Z_n, which has an impact on the waiting time of queued tasks. This guarantees that $Z_n(t)$ grows if there are tasks remaining in $Q_{o,n}(t)$ that have not been processed for a long time. In this manner, the size of Z_n can provide a bound on the latency of tasks in the queue $Q_{o,n}$. If the proposed algorithm could make control decisions to guarantee that $Q_{o,n}(t)$ and $Z_n(t)$ have finite upper bounds, then we could provide persistent service to the queued offloadable tasks with bounded worst-case latency, as illustrated in Lemma 1.

Lemma 1 *For any given time slot t, suppose this MEC system is controlled to ensure that* $Q_{o,n}(t) < Q_{o,n}^{max}$ *and* $Z_n(t) < Z_n^{max}$, *for some positive constants* $Q_{o,n}^{max}$ *and* Z_n^{max}. *Then, all tasks queued in* $Q_{o,n}$ *is processed with a maximal latency of* D_n^{max}, *given by*

$$D_n^{max} = \lceil (Q_{o,n}^{max} + Z_n^{max})/\epsilon_n \rceil \tag{18}$$

Proof Fix any time slot $t \geq 0$. We prove this theorem by contradiction. Specifically, we assume that all arrivals $W_{o,n}(t)$ are served at $t+d$, where $d > \lceil (Q_{o,n}^{max} + Z_n^{max})/\epsilon_n \rceil$. Because queueing tasks are served in a First Come First Served manner, arrived tasks during the time slot $[t+1, t+d]$ are not served, and the served tasks are merely the ones arrived before t. Then we have

$$\sum_{\tau=t+1}^{t+d} [C_{o,n}(t) + M_{o,n}(t)] \leq Q_{o,n}(t) - W_{o,n}(t) < Q_{o,n}(t) < Q_{o,n}^{max} \tag{19}$$

According to (17), we have $Z_n(t+1) \geq Z_n(t) - C_{o,n}(t) - M_{o,n}(t) + \epsilon_n$. By summing it over $[t+1, t+d]$, we further have $Z_n(t+d) - Z_n(t) \geq -\sum_{\tau=t+1}^{t+d} [C_{o,n}(t) + M_{o,n}(t)] + d\epsilon_n$. Rearranging and using the fact that $Z_n(t) \geq 0$ and $Z_n(t+d) \leq Z_n^{max}$ yields

$$\sum_{\tau=t+1}^{t+d} [C_{o,n}(t) + M_{o,n}(t)] \geq d\epsilon_n - Z_n^{max} \tag{20}$$

By combining (19) and (20), we get

$$d < (Q_{o,n}^{max} + Z_n^{max})/\epsilon_n$$

which is contradictory with the assumption. This completes the proof. □

According to Lemma 1, the worst-case latency constraint (16) could be equivalently transformed to a constraint on buffer occupancy as

$$Q_{o,n}(t) < \infty, Z_n(t) < \infty, \quad \forall n, t\,. \tag{21}$$

4.2 Lyapunov optimization

Let $\boldsymbol{\Theta}(t) \triangleq [\boldsymbol{Q}(t); \boldsymbol{Z}(t)]$ be a concatenated vector of all real queues $\boldsymbol{Q}(t)$, and virtual queues $\boldsymbol{Z}(t)$. As a scalar measure of the queue backlogs of all mobile devices, a quadratic Lyapunov function [29] is defined as follows:

$$L(\boldsymbol{\Theta}(t)) \triangleq \frac{1}{2} \sum_{n=1}^{N} [Q_{u,n}^2(t) + Q_{o,n}^2(t) + Z_n^2(t)] \tag{22}$$

An intuitive observation is that, a small value of $L(\boldsymbol{\Theta}(t))$ indicates that all the queue sizes are small. Accordingly, the MEC system has strong stability. In order to push this Lyapunov function towards a lower congestion state, we define the conditional one-slot Lyapunov drift [29] as

$$\Delta(\boldsymbol{\Theta}(t)) \triangleq \mathbb{E}[L(\boldsymbol{\Theta}(t+1)) - L(\boldsymbol{\Theta}(t)) | \boldsymbol{\Theta}(t)] \tag{23}$$

Following the drift-plus-penalty framework in [29], our aim is to make control decisions on $f_n(t)$, $P_{s,n}(t)$, $\chi_{s,n}(t)$, $C_{o,n}(t)$ and $C_{u,n}(t)$ to minimize the upper bound of the following drift-plus-penalty expression given $\boldsymbol{\Theta}(t)$:

$$\Delta(\boldsymbol{\Theta}(t)) + V\mathbb{E}\{P(t) | \boldsymbol{\Theta}(t)\} \tag{24}$$

where $V \geq 0$ is a coefficient which is selected to control the tradeoff between power minimization (i.e., ***P1***) and system stability. The critical derivation step is to find an upper bound on the expression in (24). Theorem 1 establishes this bound.

Theorem 1 (Drift-plus-penalty Bound) *For any scheduling algorithm that satisfies constraints* (11)–(15), *all possible values of* $\boldsymbol{\Theta}(t)$, *and all parameters V, the drift-plus-penalty expression in* (24) *is upper bounded by*

$$\begin{aligned}
&\Delta(\boldsymbol{\Theta}(t)) + V\mathbb{E}\{P(t) | \boldsymbol{\Theta}(t)\} \leq Y \\
&+ \sum_{n=1}^{N} \mathbb{E}\{Q_{u,n}(t) W_{u,n}(t) + Q_{o,n}(t) W_{o,n}(t) + Z_n(t)\epsilon_n | \boldsymbol{\Theta}(t)\} \\
&+ \sum_{n=1}^{N} \mathbb{E}\{[Q_{u,n}(t) - Q_{o,n}(t) - Z_n(t)] C_{o,n}(t) | \boldsymbol{\Theta}(t)\} \\
&+ \sum_{n=1}^{N} \mathbb{E}\{V P_{c,n}(t) - Q_{u,n}(t) C_n(t) | \boldsymbol{\Theta}(t)\} \\
&+ \sum_{n=1}^{N} \mathbb{E}\left\{ V \sum_{s=1}^{S} P_{s,n}(t) - (Q_{o,n}(t) + Z_n(t)) M_{o,n}(t) | \boldsymbol{\Theta}(t) \right\}
\end{aligned} \tag{25}$$

where

$$Y = \frac{\sum_{n=1}^{N}\{[W_{u,n}^{max}]^2 + [W_{o,n}^{max}]^2 + [(\tau f_n^{max}\gamma_n^{-1})^2 + (\tau f_n^{max}\gamma_n^{-1} + \tau\sum_{s=1}^{S} r_{s,n}^{max})^2] + \max[\epsilon_n^2, (\tau f_n^{max}\gamma_n^{-1} + \tau\sum_{s=1}^{S} r_{s,n}^{max})^2]\}}{2}$$

Proof Squaring both sides of the equation in (5) and (6), and because $(\max[Q-b,0]+a)^2 \le Q^2 + a^2 + b^2 + 2Q(a-b)$ for any $Q \ge 0$, $b \ge 0$, $a \ge 0$, we have

$$\begin{aligned}
&Q_{u,n}^2(t+1) - Q_{u,n}^2(t) \\
&\le [W_{u,n}^{max}]^2 + (\tau f_n^{max}\gamma_n^{-1})^2 + 2Q_{u,n}(t)[W_{u,n}(t) - C_{u,n}(t)] \\
&= [W_{u,n}^{max}]^2 + (\tau f_n^{max}\gamma_n^{-1})^2 + 2Q_{u,n}(t)[W_{u,n}(t) - C_n(t) + C_{o,n}(t)] \\
&Q_{o,n}^2(t+1) - Q_{o,n}^2(t) \\
&\le [W_{o,n}^{max}]^2 + \left(\tau f_n^{max}\gamma_n^{-1} + \tau\sum_{s=1}^{S} r_{s,n}^{max}\right)^2 \\
&\quad + 2Q_{o,n}(t)[W_{o,n}(t) - C_{o,n}(t) - M_{o,n}(t)]
\end{aligned}$$

Squaring the equation for updating $Z_n(t)$ in (17), and using the fact that $(\max[Q-b+a,0])^2 \le Q^2 + max(a^2,b^2) + 2Q(a-b)$ for any $Q \ge 0, b \ge 0, a \ge 0$, we have

$$\begin{aligned}
Z_n^2(t+1) - Z_n^2(t) \le &\max\left[\epsilon_n^2, \left(\tau f_n^{max}\gamma_n^{-1} + \tau\sum_{s=1}^{S} r_{s,n}^{max}\right)^2\right] \\
&+ 2Z_n(t)[\epsilon_n - C_{o,n}(t) - M_{o,n}(t)]
\end{aligned}$$

Summing the three bounds above over $n = 1, ..., N$, combining them all, and taking expectations with respect to $\boldsymbol{\Theta}(t)$, we get the one-slot conditional Lyapunov drift:

$$\begin{aligned}
\Delta(\boldsymbol{\Theta}(t)) \le Y &+ \sum_{n=1}^{N}\mathbb{E}\{Q_{u,n}(t)W_{u,n}(t) + Q_{o,n}(t)W_{o,n}(t) \\
&+ Z_n(t)\epsilon_n|\boldsymbol{\Theta}(t)\} \\
&+ \sum_{n=1}^{N}\mathbb{E}\{[Q_{u,n}(t) - Q_{o,n}(t) - Z_n(t)]C_{o,n}(t)|\boldsymbol{\Theta}(t)\} \\
&- \sum_{n=1}^{N}\mathbb{E}\{Q_{u,n}(t)C_n(t)|\boldsymbol{\Theta}(t)\} \\
&- \sum_{n=1}^{N}\mathbb{E}\{(Q_{o,n}(t) + Z_n(t))M_{o,n}(t)|\boldsymbol{\Theta}(t)\}
\end{aligned}$$

Now adding the penalty expression $+V\mathbb{E}\{P(t)|\boldsymbol{\Theta}(t)\}$ in (24) to both sides, we can see that this theorem holds. □

4.3 Algorithm design

Following the design principle of Lyapunov optimization theory, we introduce our Optimal tasK and Resource Allocation (OKRA) algorithm in this section. The intuition is to approximately minimize the upper bound obtained in the right-hand-side of (25), subject to constraints (11)–(15). We can decouple this problem into a series of independent subproblems, as there are no coupling constraints or objective functions between them. Then, these subproblems can be solved simultaneously and independently in a decentralized manner. Specifically, in each time slot t, based on the online observations on $\boldsymbol{\Theta}(t)$, the OKRA algorithm executes four phases of control operations, including computation task scheduling, CPU frequency scaling, power allocation and subcarrier assignment, and queue update.

4.3.1 Computation task scheduling

For each MD $n \in \mathcal{N}$ served by the MEC server, the scheduling of computation tasks in time slot t is determined with the following optimization problem ***P2***:

$$\begin{aligned}
\textbf{\textit{P2}}: \quad \min_{C_{o,n}(t), C_{u,n}(t)} \quad & [Q_{u,n}(t) - Q_{o,n}(t) - Z_n(t)]C_{o,n}(t) \\
\text{s.t.} \quad & C_{o,n}(t) + C_{u,n}(t) = C_n(t), \\
& C_{o,n}, C_{u,n} \ge 0 \quad \forall n, t
\end{aligned} \tag{26}$$

The problem ***P2*** is a linear programming that is easy to solve. We can obtain its optimal solution as

$$C_{o,n}(t) = \begin{cases} C_n(t), & Q_{u,n}(t) < Q_{o,n}(t) + Z_n(t) \\ 0, & otherwise \end{cases} \tag{27}$$

For each MD n, it can calculate $C_{o,n}(t)$ as well as $C_{u,n}(t)$ independently based on its local information on $Q_{u,n}(t)$, $Q_{o,n}(t)$ and $Z_n(t)$, according to (27).

4.3.2 CPU frequency scaling

For each MD $n \in \mathcal{N}$ served by the MEC server, the CPU-cycle frequency in time slot t could be optimized via solving the following problem ***P3***:

$$\boldsymbol{P3:}\quad \min_{f_n(t)} \quad VP_{c,n}(t) - Q_{u,n}(t)C_n(t) \qquad \text{s.t.} \quad 0 \le f_n(t) \le f_n^{max} \quad \forall n,t \tag{28}$$

By substituting $C_n(t)$ and $P_{c,n}(t)$ in (1) and (2) and then differentiating the objective function with respect to $f_n(t)$, we obtain the optimal solution to ***P3*** as follows:

$$f_n(t) = \min\left\{ \sqrt[x-1]{\frac{Q_{u,n}(t)\tau}{xV\alpha_1\gamma_n}}, f_n^{max} \right\} \tag{29}$$

For each MD n, it can calculate $f_n(t)$ independently based on its local information on $Q_{u,n}(t)$ and other known parameters, according to (29).

4.3.3 Power allocation and subcarrier assignment

According to (25) and the definition of $M_{o,n}(t)$ and $R_{o,n}(t)$, the power allocation and subcarrier assignment problem can be determined by

$$\begin{aligned} \boldsymbol{P4:}\quad \min_{\boldsymbol{P}(t),\boldsymbol{\chi}(t)} \quad & \sum_{n=1}^{N}\sum_{s=1}^{S}[VP_{s,n}(t) - (Q_{o,n}(t) + Z_n(t))\tau r_{s,n}(t)] \\ \text{s.t.} \quad & 0 \le P_{s,n}(t) \le P_{s,n}^{max} \quad \forall s,n,t \\ & 0 \le \chi_{s,n}(t) \le 1 \quad \forall s,n,t \\ & \sum_{n=1}^{N} \chi_{s,n}(t) \le 1 \quad \forall s,t \end{aligned} \tag{30}$$

Theorem 2 *Problem **P4** is jointly convex in $\boldsymbol{P}(t)$ and $\boldsymbol{\chi}(t)$.*

Proof Denote $f(P_{s,n}(t)) = \log_2(1 + P_{s,n}(t)g_{s,n}(t))$, and we know that $f(P_{s,n}(t))$ is concave in $P_{s,n}(t)$. Then, its perspective function $g(P_{s,n}(t), \chi_{s,n}(t)) = \chi_{s,n}(t) f(\frac{P_{s,n}(t)}{\chi_{s,n}(t)})$ is also concave in $(P_{s,n}(t), \chi_{s,n}(t))$ [3]. Note that $r_{s,n}(t) = B_s g(P_{s,n}(t), \chi_{s,n}(t))$, we know that the objective function in ***P4*** is jointly convex in $\boldsymbol{P}(t)$ and $\boldsymbol{\chi}(t)$ since it is the sum of convex functions. Meanwhile, all constraints in ***P4*** are linear, and thus they will construct a convex set for $\boldsymbol{P}(t)$ and $\boldsymbol{\chi}(t)$. In all, $\boldsymbol{P}(t)$ minimizes a convex function over a convex set, so it is a convex optimization problem. □

Accordingly, we can use standard convex optimization techniques and tools to solve problem ***P4*** [3]. However, the generic convex algorithms would bring about relatively high computation complexity, since these algorithms are designed for general purposes [19]. Fortunately, the special structure of problem ***P4*** can be exploited to devise a low-complexity and closed-form solution. Before we elaborate solution details, we first introduce Lemma 2 in convex optimization.

Lemma 2 *According to* [3]*, we always have*

$$\inf_{x,y} f(x,y) = \inf_{x} \tilde{f}(x) \tag{31}$$

where $\tilde{f}(x) = \inf_y f(x,y)$.

Proof See [3]. □

This is a simple and general principle that can be used to minimize a function by firstly minimizing over a certain one of the variables and then minimizing over the rest ones. According to Lemma 2, the problem ***P4*** is solved by firstly optimizing $\boldsymbol{P}(t)$ and then $\boldsymbol{\chi}(t)$.

The function $\tilde{f}$ of $\boldsymbol{\chi}(t)$ is defined as

$$\begin{aligned} \min_{\boldsymbol{P}(t)} \quad & \tilde{f}(\boldsymbol{\chi}(t)) = \sum_{n=1}^{N}\sum_{s=1}^{S}[VP_{s,n}(t) - (Q_{o,n}(t) + Z_n(t))\tau r_{s,n}(t)] \\ \text{s.t.} \quad & 0 \le P_{s,n}(t) \le P_{s,n}^{max} \quad \forall s,n,t \end{aligned} \tag{32}$$

Then the problem ***P4*** can be transformed to

$$\begin{aligned} \min_{\boldsymbol{\chi}(t)} \quad & \tilde{f}(\boldsymbol{\chi}(t)) \\ \text{s.t.} \quad & 0 \le \chi_{s,n}(t) \le 1 \quad \forall s,n,t \\ & \sum_{n=1}^{N} \chi_{s,n}(t) \le 1 \quad \forall s,t \end{aligned} \tag{33}$$

We solve (33) by differentiating $\tilde{f}(\boldsymbol{\chi}(t))$ with respect to $\boldsymbol{P}(t)$ and setting the derivatives equal to zero. Then the optimal power allocation $\boldsymbol{P}(t)$ can be derived as:

$$P_{s,n}(t) = \min\left\{ \left[\frac{(Q_{o,n}(t) + Z_n(t))\tau B_s}{V \ln 2} - \frac{1}{g_{s,n}(t)} \right]^+ \chi_{s,n}(t), P_{s,n}^{max} \right\} \tag{34}$$

where $[x]^+ \triangleq \max[0,x]$. For each MD n, it can calculate $P_{s,n}(t)$ for each $s \in S$ independently based on its local information on $Q_{o,n}(t)$, $Z_n(t)$, $g_{s,n}(t)$, $\chi_{s,n}(t)$, and other known parameters, according to (34). The time complexity for n is $\mathcal{O}(S)$.

By substituting (34) into $\tilde{f}(\boldsymbol{\chi}(t))$, we can recast (33) to

$$\begin{aligned} \min_{\boldsymbol{\chi}(t)} \quad & \tilde{f}(\boldsymbol{\chi}(t)) = \sum_{n=1}^{N}\sum_{s=1}^{S} \psi_{s,n}(t)\chi_{s,n}(t) \\ \text{s.t.} \quad & 0 \le \chi_{s,n}(t) \le 1 \quad \forall s,n,t \\ & \sum_{n=1}^{N} \chi_{s,n}(t) \le 1 \quad \forall s,t \end{aligned} \tag{35}$$

where

$$\psi_{s,n}(t)=\left[\frac{(Q_{o,n}(t)+Z_n(t))\tau B_s}{\ln 2}-\frac{V}{g_{s,n}(t)}\right]^+ -(Q_{o,n}(t)+Z_n(t))\tau B_s\left[\log_2\left(\frac{(Q_{o,n}(t)+Z_n(t))\tau B_s g_{s,n}(t)}{V\ln 2}\right)\right]^+ \tag{36}$$

Theorem 3 provides the solution to optimal subcarrier assignment $\boldsymbol{\chi}(t)$ from (35).

Theorem 3 *For any given subcarrier $s \in S$, the optimal assignment solution is given as follows:*

$$\chi_{s,n}(t)=\begin{cases}0, & \text{if} \quad \psi_{s,n}(t)\geq 0\\ 1, & \text{if} \quad \psi_{s,n}(t)<0 \quad \text{and} \quad n=\arg\min_{\omega}\psi_{s,\omega}(t)\\ 0, & \text{if} \quad \psi_{s,n}(t)<0 \quad \text{and} \quad n\neq\arg\min_{\omega}\psi_{s,\omega}(t)\end{cases} \tag{37}$$

Proof As shown in (35), the assignment of S subcarriers for MDs is independent. Thus, we can decompose (35) into S subproblems, each of which is given as follows:

$$\min_{\boldsymbol{\chi}(t)} \quad \sum_{n=1}^{N}\psi_{s,n}(t)\chi_{s,n}(t) \tag{38}$$

$$\text{s.t.} \quad 0\leq\chi_{s,n}(t)\leq 1 \quad \forall n,t \tag{39}$$

$$\sum_{n=1}^{N}\chi_{s,n}(t)\leq 1 \quad \forall t \tag{40}$$

For minimizing the objective function (38), we should set $\chi_{s,n}(t)=0$ for MDs with $\psi_{s,n}(t)\geq 0$. On the other hand, the case when $\psi_{s,n}(t)<0$ can be demonstrated from a simple scenario with two MDs. For any $n_1,n_2\in\mathcal{N}$, we have $\chi_{s,n_1}(t)+\chi_{s,n_2}(t)\leq 1$ from (40). Then, (38) is equal to

$$\begin{aligned}&\psi_{s,n1}(t)\chi_{s,n1}(t)+\psi_{s,n2}(t)\chi_{s,n2}(t)\\ &\quad\geq\psi_{s,n1}(t)(1-\chi_{s,n2}(t))+\psi_{s,n2}(t)\chi_{s,n2}(t)\\ &\quad\geq\psi_{s,n1}(t)+(\psi_{s,n2}(t)-\psi_{s,n1}(t))\chi_{s,n2}(t)\end{aligned} \tag{41}$$

Our aim is to minimize (41) subject to constraints (39) and (40). If $\psi_{s,n1}(t)<\psi_{s,n2}(t)<0$, we have $\chi_{s,n1}(t)=1$ and $\chi_{s,n2}(t)=0$. Otherwise, if $\psi_{s,n2}(t)<\psi_{s,n1}(t)<0$, we have $\chi_{s,n1}(t)=0$ and $\chi_{s,n2}(t)=1$. We can easily use similar approaches to deal with the cases with more than two MDs. □

For each MD n, it can calculate $\psi_{s,n}(t)$ for each $s\in S$ independently based on its local information on $Q_{o,n}(t)$, $Z_n(t)$, $g_{s,n}(t)$, and other known parameters, according to (36). Then the MEC server will gather all $\psi_{s,n}(t)$ for each s from each n to make the optimal assignment decision on $\boldsymbol{\chi}(t)$, according to Theorem (3). The time complexity for the MEC server is $\mathcal{O}(NS)$.

4.3.4 Queue update

Finally, the queues $\boldsymbol{Q}(t)$ and $\boldsymbol{Z}(t)$ will be updated according to (5), (6) and (17), by using the optimal results of $C_{o,n}(t)$, $f_n(t)$, $P_{s,n}(t)$ and $\chi_{s,n}(t)$ determined in the phases above.

4.4 Implementation issues

We note that OKRA makes online decisions based on the knowledge of the queue sizes $\boldsymbol{Q}(t)$ and $\boldsymbol{Z}(t)$. It does not need pre-knowledge on task arrivals and channel conditions. This is very useful in practice since these information are usually difficult to obtain or predict. Besides, the MEC server only needs to gather $\psi_{s,n}(t)$ from each MD $n\in\mathcal{N}$, so as to determine $\chi_{s,n}(t)$ based on (37) for these MDs. All the other control decisions and relevant calculations are performed on each MD in a fully distributed manner. This can significantly lighten the computation load and reduce the implementation complexity, which is often desirable in practice [19, 29]. Furthermore, as will be shown in next section, we can analytically characterize the algorithm performance. It is useful to facilitate the parameter tuning in real-world scenarios.

5 Performance analysis

Theorem 4 *Under the OKRA algorithm, we have the following:*

(a) The lengths of $Q_{o,n}$ and Z_n are upper bounded by constants $Q_{o,n}^{max}$ and Z_n^{max} which are given respectively as follows:

$$Q_{o,n}^{max}\triangleq V\Upsilon+W_{o,n}^{max} \tag{42}$$

$$Z_n^{max}\triangleq V\Upsilon+\epsilon_n \tag{43}$$

where $\Upsilon=\max\left[\frac{(f_n^{max})^{x-1}x\alpha_1\gamma_n}{\tau},\frac{\ln 2(P_{s,n}^{max}+\frac{1}{g_{s,n}^{max}})}{\tau B_s}\right]$.

(b) The maximal (i.e., worst-case) latency for tasks in queue $Q_{o,n}$ is:

$$D_n^{max}\triangleq\lceil(2V\Upsilon+W_{o,n}^{max}+\epsilon_n)/\epsilon_n\rceil \tag{44}$$

(c) For any $V>0$, OKRA is able to stabilize the system, and has a resulted time-average queue backlog and power consumption that satisfy the following bound:

$$\overline{Q}\leq\frac{Y+V\overline{P}^*}{\omega} \tag{45}$$

$$\overline{P}\leq\overline{P}^*+\frac{Y}{V} \tag{46}$$

where $\omega > 0$ *is a constant, and* $\overline{P}^*$ *is the optimal value of* ***P1***.

Proof (a) To prove (42), we need to prove $Q_{o,n}(t) \le V\Upsilon + W_{o,n}^{max}$ for all t. Obviously, it holds for $t = 0$ because $Q_{o,n}(0) = 0$. Assume that the Eq. (42) holds for some t, we induce that it also holds for $t + 1$.

(1) if $Q_{o,n}(t) \le V\Upsilon$, we can obtain the following:

$$\begin{aligned} Q_{o,n}(t+1) &= \max\{Q_{o,n}(t) - C_{o,n}(t) - M_{o,n}(t), 0\} + W_{o,n}(t) \\ &\le Q_{o,n}(t) + W_{o,n}(t) \\ &\le V\Upsilon + W_{o,n}^{max} \end{aligned}$$

(2) In the other case, if $V\Upsilon < Q_{o,n}(t) \le V\Upsilon + W_{o,n}^{max}$. In this case, the following holds:

$$Q_{o,n}(t) + Z_n(t) \ge Q_{o,n}(t) > V\Upsilon$$

Besides, we know that the system will maintain the state that $Q_{u,n}(t) \approx Q_{o,n}(t) + Z_n(t)$ according to (27). Based on these results as well as (29) and (34), we can see that $f_n(t)$, $P_{s,n}(t)$ and $\chi_{s,n}(t)$ would choose their maximal values [18], i.e., f_n^{max}, $P_{s,n}^{max}$ and 1, respectively. Thus, $C_{o,n}(t) + M_{o,n}(t) = C_{o,n}^{max} + M_{o,n}^{max}$. In case (2.1), if $Q_{o,n}(t) - C_{o,n}^{max} - M_{o,n}^{max} \le 0$, refers to (6), we have

$$Q_{o,n}(t+1) = W_{o,n}(t) \le W_{o,n}^{max} \le V\Upsilon + W_{o,n}^{max}$$

In case (2.2), if $Q_{o,n}(t) - C_{o,n}^{max} - M_{o,n}^{max} > 0$, recall that the queue $Q_{o,n}$ is assumed to be finite and thus in the optimization process $C_{o,n}^{max} + M_{o,n}^{max} \ge W_{o,n}^{max} \ge W_{o,n}(t)$ holds, we have

$$\begin{aligned} Q_{o,n}(t+1) &= Q_{o,n}(t) - C_{o,n}^{max} - M_{o,n}^{max} \\ &\quad + W_{o,n}(t) \le Q_{o,n}(t) \le V\Upsilon + W_{o,n}^{max} \end{aligned}$$

Therefore, $Q_{o,n}(t) \le V\Upsilon + W_{o,n}^{max}$ for any t. The proof that $Z_n(t) \le V\Upsilon + \epsilon_n$ for all t can be proved similarly, so we omit it for brevity.

(b) This can be proved immediately from part (a) together with Lemma 1.

(c) According to (25), our algorithm attempts to minimize the right-hand-side of the following expression:

$$\begin{aligned} &\Delta(\boldsymbol{\Theta}(t)) + V\mathbb{E}\{P(t)|\boldsymbol{\Theta}(t)\} \le Y + V\mathbb{E}\{P(t)|\boldsymbol{\Theta}(t)\} \\ &+ \sum_{n=1}^{N} \mathbb{E}\{Q_{u,n}(t)[W_{u,n}(t) - C_{u,n}(t)]|\boldsymbol{\Theta}(t)\} \\ &+ \sum_{n=1}^{N} \mathbb{E}\{Q_{o,n}(t)[W_{o,n}(t) - C_{o,n}(t) - M_{o,n}(t)]|\boldsymbol{\Theta}(t)\} \\ &+ \sum_{n=1}^{N} \mathbb{E}\{Z_n(t)[\epsilon_n - C_{o,n}(t) - M_{o,n}(t)]|\boldsymbol{\Theta}(t)\} \end{aligned} \tag{47}$$

Because we assume that the arrival process is within its capacity region, there exists at least one stationary, randomized control policy which is independent of the current queue backlogs $\boldsymbol{\Theta}(t)$ and can stabilize the queue [19, 29], with

$$\mathbb{E}\{P(t)|\boldsymbol{\Theta}(t)\} = \mathbb{E}\{P(t)\} = \overline{P}^* \tag{48}$$

Since the arrival and service pattern of computation tasks can be controlled, we know that there exists some finite number $\omega > 0$ such that the expectation of the differences between services and arrivals of each queue is larger than ω [29]. Then, we have

$$\mathbb{E}\{C_{u,n}(t) - W_{u,n}(t)|\boldsymbol{\Theta}(t)\} \ge \omega \tag{49}$$

$$\mathbb{E}\{C_{o,n}(t) + M_{o,n}(t) - W_{o,n}(t)|\boldsymbol{\Theta}(t)\} \ge \omega \tag{50}$$

$$\mathbb{E}\{C_{o,n}(t) + M_{o,n}(t) - \epsilon_n|\boldsymbol{\Theta}(t)\} \ge \omega \tag{51}$$

Plugging (48)–(51) into (47) results in:

$$\begin{aligned} &\Delta(\boldsymbol{\Theta}(t)) + V\mathbb{E}\{P(t)|\boldsymbol{\Theta}(t)\} \le Y + V\overline{P}^* - \omega \sum_{n=1}^{N} [Q_{u,n}(t) \\ &\quad + Q_{o,n}(t) + Z_n(t)] \end{aligned}$$

This inequality is in the exact form for application of Lyapunov optimization, as shown in Theorem 4.2 of [29]). Accordingly, we know that all queues are mean rate stable [29]. Then, taking expectations over $\boldsymbol{\Theta}(t)$ on both sides, and using iterative expectation law yields:

$$\begin{aligned} &\mathbb{E}\{L(\Theta(t+1)) - L(\Theta(t))\} + V\mathbb{E}\{P(t)\} \le Y + V\overline{P}^* \\ &\quad - \omega \sum_{n=1}^{N} \mathbb{E}[Q_{u,n}(t) + Q_{o,n}(t) + Z_n(t)] \end{aligned} \tag{52}$$

Using telescoping sums over $t \in \{0, 1, ..., T-1\}$, using the fact that $L(\Theta(t)) \ge 0$ and $L(\Theta(0)) = 0$ for all t, and then dividing both sides by T, we obtain

$$\frac{\omega}{T} \sum_{t=0}^{T-1} \sum_{n=1}^{N} \mathbb{E}[Q_{u,n}(t) + Q_{o,n}(t) + Z_n(t)] \le Y + V\overline{P}^* \tag{53}$$

Using the fact that $\mathbb{E}\{Z_n(t)\} \ge 0$ for all n, and taking limits as $T \to \infty$ result in the queue backlog bound given in (45).

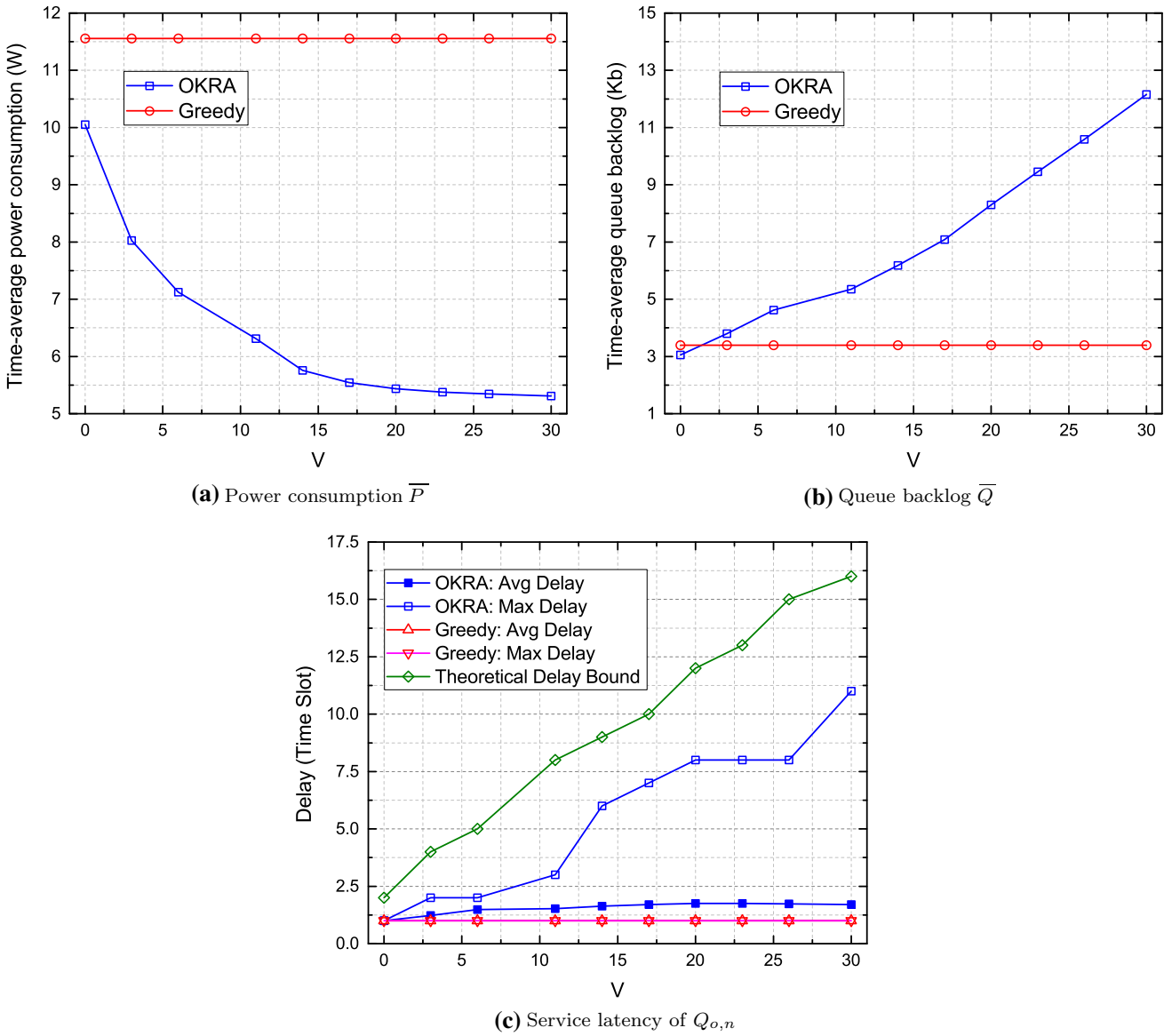
To prove (46), we can use (52) to get

$$V\mathbb{E}\{P(t)\} \le Y + V\overline{P}^* \tag{54}$$

Summing (54) over $t = 0, 1, ..., T-1$, and dividing both sides by TV, we obtain

$$\frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}\{P(t)\} \le \overline{P}^* + \frac{Y}{V}$$

Again, taking limits as $T \to \infty$ results in the power cost bound given in (46). □



(a) Power consumption $\overline{P}$

(b) Queue backlog $\overline{Q}$

(c) Service latency of $Q_{o,n}$

Fig. 1 System performance under different value for the control parameter V

Table 1 Comparison on upper bounds of $Q_{o,1}$ under different V values

V	0	1	4	7	10	13	16	19	22	25	30
TH	2	3	4	6	7	9	10	11	13	14	16
EX	2	2	3	4	4	5	6	6	6	6	7

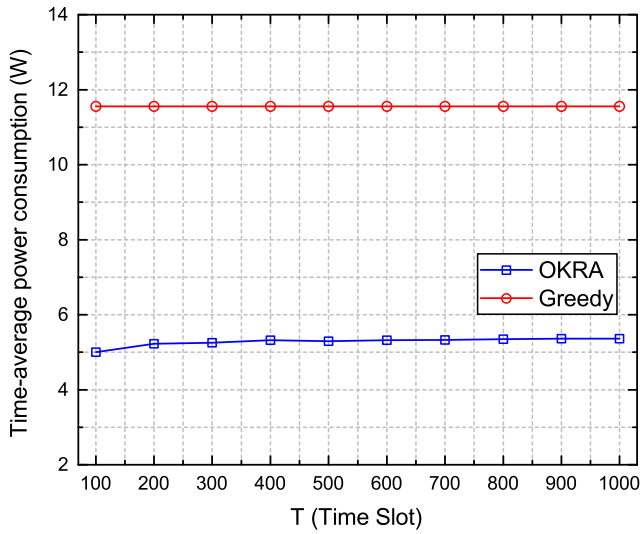
The theorem above shows the $[\mathcal{O}(1/V), \mathcal{O}(V)]$ power-stability tradeoff for the problem ***P1***. It also implies the worst-case latency is upper bounded by $\mathcal{O}(V)$. By tuning the value of V, the near-optimal value of time-average power consumption can be achieved while bringing in larger queue size and higher latency. We verify these with simulation experiments in what follows.

6 Experimental results

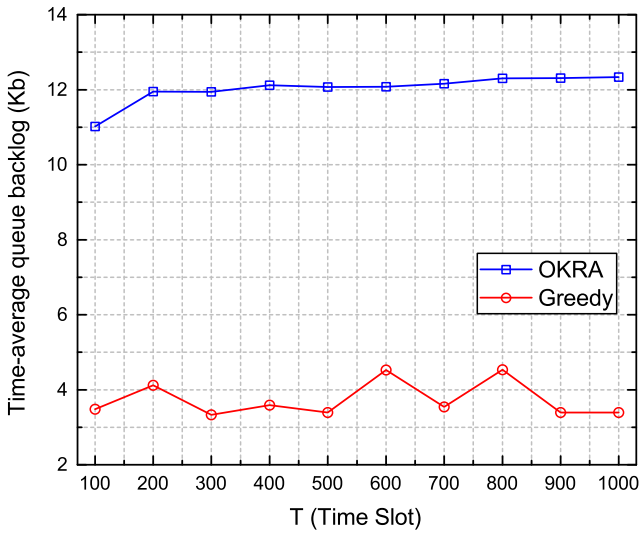
In this section, simulation results are given to demonstrate the performance of our OKRA algorithm.

6.1 Simulation setting

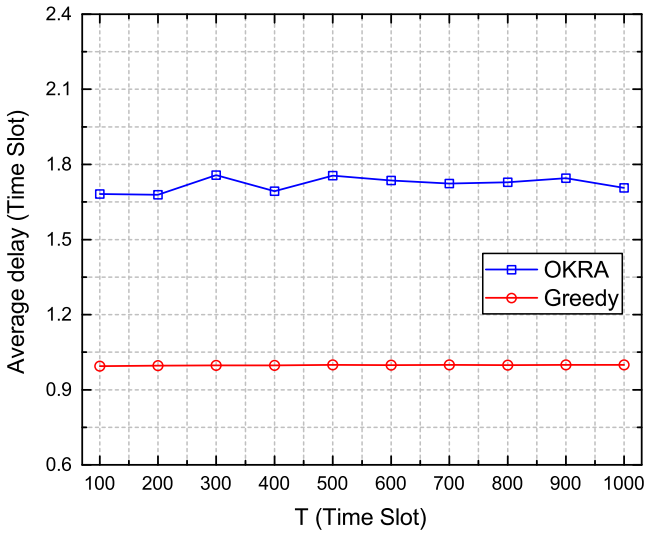
Generally, the parameters in simulation experiments are selected according to some empirical studies in literature. Unless stated otherwise, in all the experiments these parameters are set as follows. We consider a MEC system serving $N = 3$ MDs, each of which has $S = 12$ subcarriers [12]. For each MD $n \in \mathcal{N}$, its CPU computation capacity $f_n^{max} = 1.6$ GHz [16], and $\gamma_n = 737.5$ cycles/bit [28]. We choose $\alpha_1 = 0.33 \times 10^{-18}$, $\alpha_2 = 0.1$ and $x = 3$ [16] for



(a) Power consumption $\overline{P}$



(b) Queue backlog $\overline{Q}$

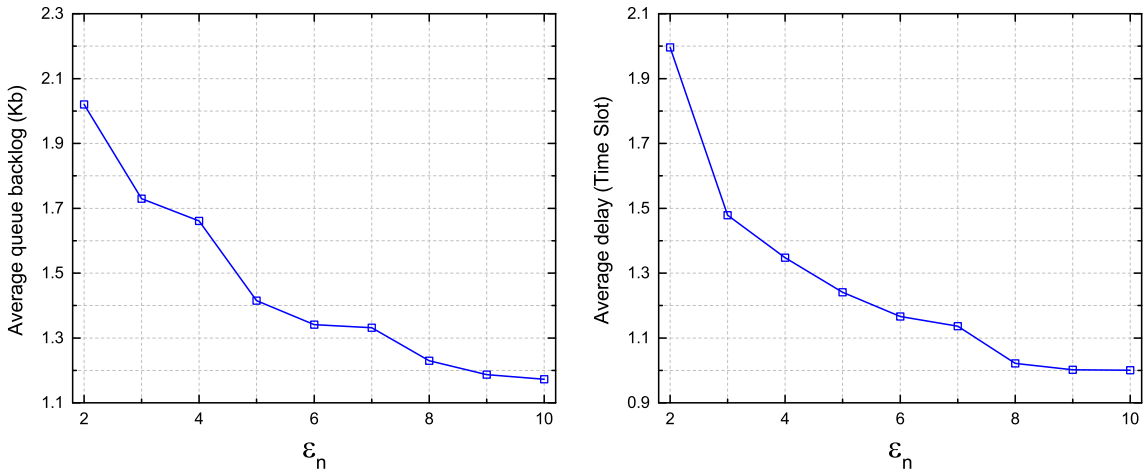


(c) Service latency of $Q_{o,n}$

Fig. 2 System performance under different value for the time scale T

power parameters in (2). The channel power gain $g_{s,n}(t)$ is exponentially distributed with unit mean 1 [28]. Besides, $P_{s,n}^{max} = 0.2$ W [12], and $B_s = 1.8$ MHz [12, 28]. The length of one time slot $\tau = 1$ ms [28]. All task arrivals follows the Poisson Process [11] with $W_{u,n}^{max} = W_{o,n}^{max} = 2$ kB/Slot [28]. Note that our algorithm does not require any special setting for this traffic pattern [11].

To fully study the OKRA performance, we compare it with two heuristic solutions for different verification purposes. The first one ("Greedy") is a baseline online control algorithm, which makes greedy control decisions in each time slot [11, 29]. This greedy algorithm always chooses $f_n(t) = f_n^{max}$ and $P_{s,n}(t) = P_{s,n}^{max}$, and randomly assigns a given subcarrier to any one of the MDs. Based on Theorem 2.4 in [29], a simple control policy is used to enforce that $Q_{o,n}$ will be preferentially served if the average amount of incoming workload is larger than that of outgoing workload in any time slot [11]. It is intuitively that the Greedy algorithm will guarantee good latency performance for offloadable tasks. The second algorithm ("CPU_Only") only uses local CPU resources, and processes the two types of tasks in a round-robin fashion [35]. It is obvious that this algorithm can not provide any guarantees on stability and latency.



(a) Average queue backlog of $Q_{o,1}$ **(b)** Service latency of $Q_{o,1}$

Fig. 3 System performance under different value for the control parameter ϵ_1

6.2 Results and analysis

6.2.1 The impact of *V*

We fix $T = 1000$ and $\epsilon_n = 2$ for all n, and then run simulations with different V values. Figure 1 presents the simulation results with the growth of parameter V. We have three observations about these results. First, we can see that the time-average power consumption of MDs achieved by OKRA falls significantly, and converges quickly to very close to the optimal value as V increases (Fig. 1a). Meanwhile, the time-average queue backlog grows linearly as the value of V increases (Fig. 1b). It clearly shows that the tradeoff between power consumption and system stability can be adjusted by tuning the parameter V, and quantitatively confirms the $[\mathcal{O}(1/V), \mathcal{O}(V)]$ power-stability tradeoff in part (c) of Theorem 4. Second, though Greedy indeed has relatively smaller backlog (Fig. 1b) and lower latency (Fig. 1c) than OKRA, it will bring about much more (i.e., 14.9–117.7%) energy consumption (Fig. 1a) for task processing. That's because Greedy does not take energy saving into consideration in its design. Third, it is obvious that the maximal service latency of offloadable tasks never exceeds the corresponding theoretical bound (Fig. 1c). The results indicate that the increasing rate of the actual maximal latency can be well bounded by a linear function of V, which is consistent with part (b) of Theorem 4.

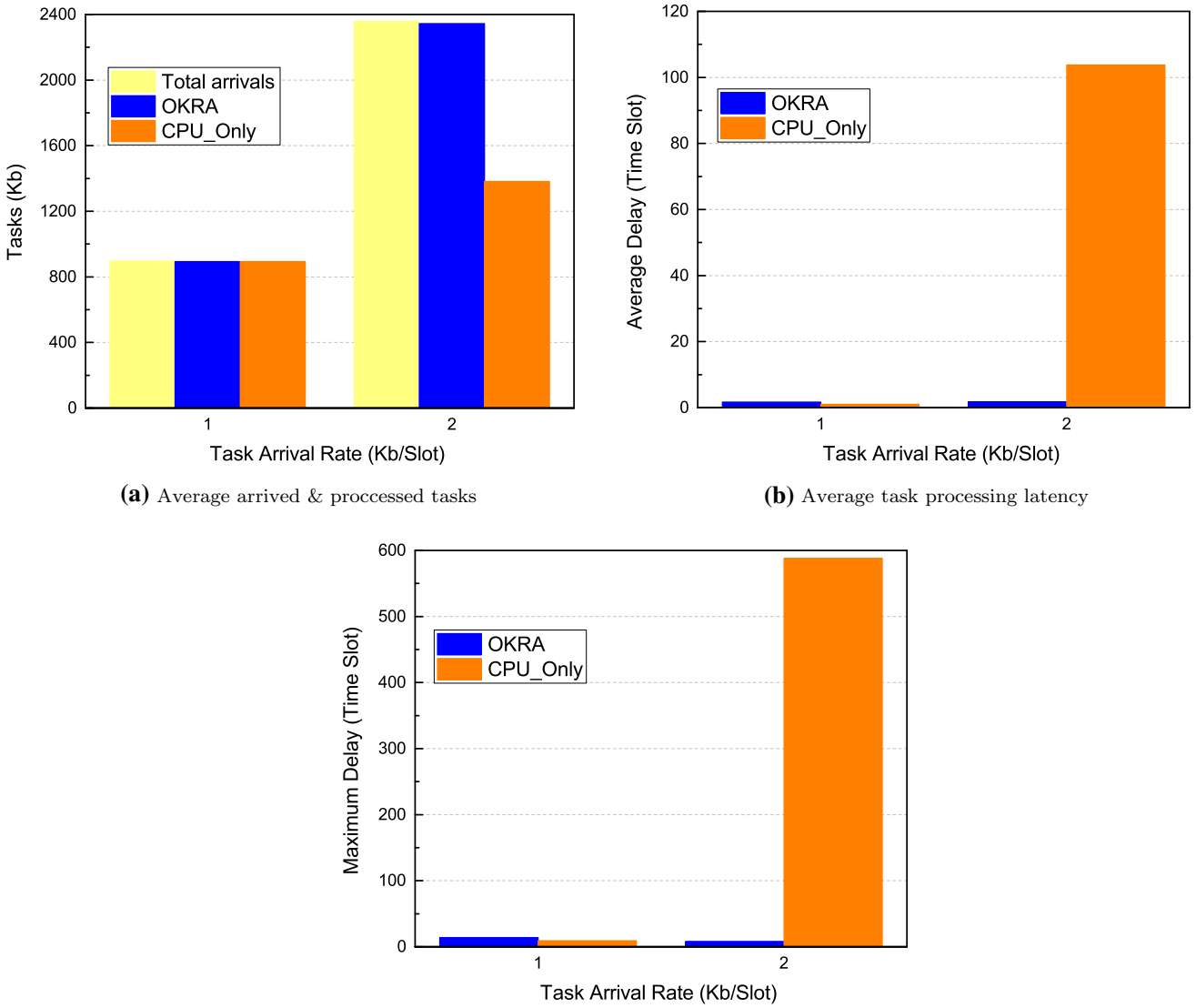
Furthermore, we compare the theoretical (TH) and experimental (EX) upper bounds for queue backlogs of $Q_{o,1}$ in the same scenarios. Table 1 presents the comparison results, from which we can find that the queue backlogs are smaller than the corresponding theoretical bounds, especially when V is relatively large. These observations are consistent with (42) in part (a) of Theorem 4.

6.2.2 The impact of *T*

We fix $V = 30$ and $\epsilon_n = 2$ for all n, and then change T from 100 to 1000 time slots, so as to investigate the characteristics of different time-scales of long-term operation. The simulation results are plotted in Fig. 2. It is clear that changing T has relatively small impacts on system stability and algorithm performance. The fluctuations on power consumption, queue size, and queueing latency of offloadable tasks are $[-5.26\%, 1.52\%]$, $[-8.37\%, 2.55\%]$, and $[-2.60\%, 1.98\%]$, respectively, for OKRA, and are $[-0.0009\%, +0.0009\%]$, $[-10.74\%, 21.52\%]$, and $[-0.31\%, 0.23\%]$, respectively, for Greedy. These results confirm that both OKRA and Greedy can provide stable performance guarantee over time. According to Theorem 2.4 in [29], such a dynamic queueing system will be kept stable if we could make the average arrival workload not greater than the average processed workload in the long run. We ensure this by minimizing the Lyapunov drift (23) in OKRA, and by the simple control policy (described in 6.1) in Greedy.

6.2.3 The impact of ϵ_n

We fix $V = 30$ and $T = 1000$, and then run simulations with different ϵ_n values. Actually, it is not necessary to verify each ϵ_n [18]. Thus, without loss of generality, we take only ϵ_1 into consideration, and study its impact on system performance. Figure 3 presents the results, in which we can find that both the queue size and the service latency of offloadable tasks decrease when the parameter ϵ_1 increases. According to the definition of queue Z_n in (17) and the task scheduling rule in (27), a larger ϵ_n makes the queue $Z_n(t)$ grow faster, so OKRA is more inclined to serve queue $Q_{o,n}$ than queue $Q_{u,n}$. These results are consistent



(a) Average arrived & proccessed tasks

(b) Average task processing latency

(c) Maximal task processing latency

Fig. 4 System performance under different workload intensity ($W_{u,n}^{max}$, $W_{o,n}^{max}$)

with the theoretical bounds on queue size and service delay given in (18) and (44).

6.2.4 The impact of workload intensity

We fix $V = 30$, $T = 1000$ and $\epsilon_n = 2$ for all n, and then run simulations with different arrival rates of computation tasks. The results are depicted in Fig. 4. When the arrival rates are relatively low, i.e., $W_{u,n}^{max} = W_{o,n}^{max} = 1$ kB/Slot, the local mobile CPUs can provide sufficient computation resources for task processing. Therefore, both OKRA and CPU_Only can satisfy task arrivals in the given period of 1000 time slots. Meanwhile, CPU_Only has a better latency performance than OKRA, since the latter trades off processing latency for energy saving. However, when the arrival rates are relatively high, i.e., $W_{u,n}^{max} = W_{o,n}^{max} = 2$ kB/Slot, the performance of CPU_Only is restricted to a large extent because of the constrained computation capacity of mobile CPUs. In the given period, CPU_Only is only able to process 59% of arrived tasks, while OKRA processes nearly all tasks with the aid of MEC server. Furthermore, CPU_Only brings about excessive service latency for computation tasks because of the severe queue congestion. Based on these results, we verify the necessities and benefits of MEC systems for improving user experience in computation services.

7 Conclusion

This paper studies how to minimize power consumption of mobile devices in MEC systems. By leveraging Lyapunov optimization, we designed an online control algorithm called OKRA in response to stochastic task arrivals and time-varying channel conditions. OKRA provides simple and distributed approaches to make optimal decisions on computation task scheduling, CPU frequency scaling, transmit power allocation and subcarrier bandwidth assignment with low complexity. Unlike conventional statistical offline or prediction-based approaches, OKRA does not need to pre-learn any statistical knowledge on system dynamics. The theoretical analysis and simulation results have verified the capability of OKRA in terms of power optimality, queue stability and latency guarantee.

As future work, we are going to extend this work to scenarios with capacity constraints on computation resources for MEC servers [41]. Another direction is to further evaluate the OKRA algorithm in a prototype implementation of MEC system [33], which would be interesting and also very challenging.

Acknowledgements This work was supported by the Fundamental Research Funds for the Central Universities of China under Grants 2019JBM027.

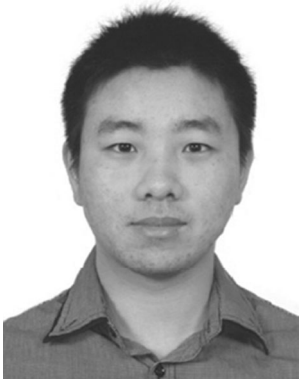
References

1. Ahmed, E., & Rehmani, M. H. (2017). Mobile edge computing: Opportunities, solutions, and challenges. *Future Generation Computer Systems*, *70*, 59–63. https://doi.org/10.1016/j.future.2016.09.015.
2. Barbarossa, S., Sardellitti, S., & Lorenzo, P. D. (2013). Joint allocation of computation and communication resources in multiuser mobile cloud computing. In *IEEE 14th workshop on signal processing advances in wireless communications (SPAWC)* (pp. 26–30). https://doi.org/10.1109/SPAWC.2013.6612005.
3. Boyd, S., & Vandenberghe, L. (2004). *Convex optimization*. New York, NY: Cambridge University Press.
4. Chen, X., Jiao, L., Li, W., & Fu, X. (2016). Efficient multi-user computation offloading for mobile-edge cloud computing. *IEEE/ACM Transactions on Networking*, *24*(5), 2795–2808. https://doi.org/10.1109/TNET.2015.2487344.
5. Cisco. (2017). Cisco Visual Networking Index: Global mobile data traffic forecast update, 2016–2021 Whitepaper.
6. Dinh, T. Q., Tang, J., La, Q. D., & Quek, T. Q. S. (2017). Adaptive computation scaling and task offloading in mobile edge computing. In *IEEE wireless communications and networking conference (WCNC)* (pp. 1–6). https://doi.org/10.1109/WCNC.2017.7925612.
7. Fan, Q., & Ansari, N. (2018). Application aware workload allocation for edge computing-based iot. *IEEE Internet of Things Journal*, *5*(3), 2146–2153. https://doi.org/10.1109/JIOT.2018.2826006.
8. Fan, Q., & Ansari, N. (2018). Towards workload balancing in fog computing empowered IoT. *IEEE Transactions on Network Science and Engineering*,. https://doi.org/10.1109/TNSE.2018.2852762.
9. Fan, Q., Ansari, N., & Sun, X. (2017). Energy driven avatar migration in green cloudlet networks. *IEEE Communications Letters*, *21*(7), 1601–1604. https://doi.org/10.1109/LCOMM.2017.2684812.
10. Fang, W., An, Z., Shu, L., Liu, Q., Xu, Y., & An, Y. (2014). Achieving optimal admission control with dynamic scheduling in energy constrained network systems. *Journal of Network and Computer Applications*, *44*, 152–160. https://doi.org/10.1016/j.jnca.2014.05.009.
11. Fang, W., Li, Y., Zhang, H., Xiong, N., Lai, J., & Vasilakos, A. V. (2014). On the throughput-energy tradeoff for data transmission between cloud and mobile devices. *Information Sciences*, *283*, 79–93. https://doi.org/10.1016/j.ins.2014.06.022. (New trend of computational intelligence in human–robot interaction).
12. Huang, J., Qian, F., Gerber, A., Mao, Z. M., Sen, S., & Spatscheck, O. (2012). A close examination of performance and power characteristics of 4G LTE networks. In *Proceedings of the 10th international conference on mobile systems, applications, and services, MobiSys '12* (pp. 225–238). New York, NY: ACM. https://doi.org/10.1145/2307636.2307658.
13. Jeong, S., Simeone, O., & Kang, J. (2017). Mobile edge computing via a UAV-mounted cloudlet: Optimization of bit allocation and path planning. *IEEE Transactions on Vehicular Technology*,. https://doi.org/10.1109/TVT.2017.2706308.
14. Kumar, K., Liu, J., Lu, Y. H., & Bhargava, B. (2013). A survey of computation offloading for mobile systems. *Mobile Networks and Applications*, *18*(1), 129–140. https://doi.org/10.1007/s11036-012-0368-0.
15. Kwak, J., Choi, O., Chong, S., & Mohapatra, P. (2014). Dynamic speed scaling for energy minimization in delay-tolerant smartphone applications. In *IEEE conference on computer communications, IEEE INFOCOM* (pp. 2292–2300). https://doi.org/10.1109/INFOCOM.2014.6848173.
16. Kwak, J., Kim, Y., Lee, J., & Chong, S. (2015). Dream: Dynamic resource and task allocation for energy minimization in mobile cloud systems. *IEEE Journal on Selected Areas in Communications*, *33*(12), 2510–2523. https://doi.org/10.1109/JSAC.2015.2478718.
17. Li, A., Yang, X., Kandula, S., & Zhang, M. (2010). Cloudcmp: Comparing public cloud providers. In *Proceedings of the 10th ACM SIGCOMM conference on internet measurement, IMC '10* (pp. 1–14). New York, NY: ACM. https://doi.org/10.1145/1879141.1879143.
18. Li, S., Zhou, Y., Jiao, L., Yan, X., Wang, X., & Lyu, M. R. T. (2015). Towards operational cost minimization in hybrid clouds for dynamic resource provisioning with delay-aware optimization. *IEEE Transactions on Services Computing*, *8*(3), 398–409. https://doi.org/10.1109/TSC.2015.2390413.
19. Li, Y., Shi, Y., Sheng, M., Wang, C. X., Li, J., Wang, X., et al. (2016). Energy-efficient transmission in heterogeneous wireless networks: A delay-aware approach. *IEEE Transactions on Vehicular Technology*, *65*(9), 7488–7500. https://doi.org/10.1109/TVT.2015.2472578.
20. Liu, F., Shu, P., Jin, H., Ding, L., Yu, J., Niu, D., et al. (2013). Gearing resource-poor mobile devices with powerful clouds: Architectures, challenges, and applications. *IEEE Wireless Communications*, *20*(3), 14–22. https://doi.org/10.1109/MWC.2013.6549279.
21. Liu, J., Mao, Y., Zhang, J., & Letaief, K. B. (2016). Delay-optimal computation task scheduling for mobile-edge computing systems. In *IEEE international symposium on information theory (ISIT)* (pp. 1451–1455). https://doi.org/10.1109/ISIT.2016.7541539.

22. Liu, L., Chang, Z., & Guo, X. (2018). Socially aware dynamic computation offloading scheme for fog computing system with energy harvesting devices. *IEEE Internet of Things Journal*, *5*(3), 1869–1879. https://doi.org/10.1109/JIOT.2018.2816682.
23. Liu, L., Chang, Z., Guo, X., Mao, S., & Ristaniemi, T. (2018). Multiobjective optimization for computation offloading in fog computing. *IEEE Internet of Things Journal*, *5*(1), 283–294. https://doi.org/10.1109/JIOT.2017.2780236.
24. Liu, L., Guo, X., Chang, Z., & Ristaniemi, T. (2019). Joint optimization of energy and delay for computation offloading in cloudlet-assisted mobile cloud computing. *Wireless Networks*, *25*(4), 2027–2040.
25. Ma, X., Zhao, Y., Zhang, L., Wang, H., & Peng, L. (2013). When mobile terminals meet the cloud: Computation offloading as the bridge. *IEEE Network*, *27*(5), 28–33. https://doi.org/10.1109/MNET.2013.6616112.
26. Mach, P., & Becvar, Z. (2017). Mobile edge computing: A survey on architecture and computation offloading. *IEEE Communications Surveys Tutorials*,. https://doi.org/10.1109/COMST.2017.2682318.
27. Mao, Y., Zhang, J., & Letaief, K. B. (2016). Dynamic computation offloading for mobile-edge computing with energy harvesting devices. *IEEE Journal on Selected Areas in Communications*, *34*(12), 3590–3605. https://doi.org/10.1109/JSAC.2016.2611964.
28. Mao, Y., Zhang, J., Song, S. H., & Letaief, K. B. (2016). Power-delay tradeoff in multi-user mobile-edge computing systems. In *IEEE global communications conference (GLOBECOM)* (pp. 1–6). https://doi.org/10.1109/GLOCOM.2016.7842160.
29. Neely, M. J. (2010). *Stochasitic network optimization with application to communication and queueing systems*. San Rafael: Morgan & Claypool.
30. Neely, M. J. (2011). Opportunistic scheduling with worst case delay guarantees in single and multi-hop networks. In *Proceedings IEEE INFOCOM* (pp. 1728–1736). https://doi.org/10.1109/INFCOM.2011.5934971.
31. Patel, M., Naughton, B., Chan, C., Sprecher, N., Abeta, S., Neal, A., et al. (2014). Mobile-edge computing introductory technical white paper, White paper, Mobile-edge computing (MEC) industry initiative.
32. Samanta, A., & Chang, Z. (2019). Adaptive service offloading for revenue maximization in mobile edge computing with delay-constraint. *IEEE Internet of Things Journal*,. https://doi.org/10.1109/JIOT.2019.2892398.
33. Satyanarayanan, M. (2017). The emergence of edge computing. *Computer*, *50*(1), 30–39. https://doi.org/10.1109/MC.2017.9.
34. Satyanarayanan, M., Bahl, P., Caceres, R., & Davies, N. (2009). The case for vm-based cloudlets in mobile computing. *IEEE Pervasive Computing*, *8*(4), 14–23. https://doi.org/10.1109/MPRV.2009.82.
35. Shi, C., Habak, K., Pandurangan, P., Ammar, M., Naik, M., & Zegura, E. (2014). Cosmos: Computation offloading as a service for mobile devices. In *Proceedings of the 15th ACM international symposium on mobile ad hoc networking and computing, MobiHoc '14* (pp. 287–296). New York, NY: ACM. https://doi.org/10.1145/2632951.2632958.
36. Sun, X., Ansari, N., & Fan, Q. (2015). Green energy aware avatar migration strategy in green cloudlet networks. In *IEEE 7th international conference on cloud computing technology and science (CloudCom)* (pp. 139–146). https://doi.org/10.1109/CloudCom.2015.23.
37. Tran, T. X., Pandey, P., Hajisami, A., & Pompili, D. (2017). Collaborative multi-bitrate video caching and processing in mobile-edge computing networks. In *13th Annual conference on wireless on-demand network systems and services (WONS)* (pp. 165–172). https://doi.org/10.1109/WONS.2017.7888772.
38. Urgaonkar, R., Urgaonkar, B., Neely, M. J., & Sivasubramaniam, A. (2011). Optimal power cost management using stored energy in data centers. In *Proceedings of the ACM SIGMETRICS joint international conference on measurement and modeling of computer systems, SIGMETRICS '11* (pp. 221–232). New York, NY: ACM. https://doi.org/10.1145/1993744.1993766.
39. Wang, Y., Sheng, M., Wang, X., Wang, L., & Li, J. (2016). Mobile-edge computing: Partial computation offloading using dynamic voltage scaling. *IEEE Transactions on Communications*, *64*(10), 4268–4282. https://doi.org/10.1109/TCOMM.2016.2599530.
40. Wu, H., Knottenbelt, W., Wolter, K., & Sun, Y. (2016). *An optimal offloading partitioning algorithm in mobile cloud computing* (pp. 311–328). Cham: Springer. https://doi.org/10.1007/978-3-319-43425-4_21.
41. You, C., Huang, K., Chae, H., & Kim, B. H. (2017). Energy-efficient resource allocation for mobile-edge computation offloading. *IEEE Transactions on Wireless Communications*, *16*(3), 1397–1411. https://doi.org/10.1109/TWC.2016.2633522.
42. Yu, Y., Zhang, J., & Letaief, K. B. (2016). Joint subcarrier and cpu time allocation for mobile edge computing. In *IEEE global communications conference (GLOBECOM)* (pp. 1–6). https://doi.org/10.1109/GLOCOM.2016.7841937.
43. Zhang, K., Mao, Y., Leng, S., Zhao, Q., Li, L., Peng, X., et al. (2016). Energy-efficient offloading for mobile edge computing in 5G heterogeneous networks. *IEEE Access*, *4*, 5896–5907. https://doi.org/10.1109/ACCESS.2016.2597169.

Weiwei Fang received the B.S. degree from Hefei University of Technology, Hefei, China, and the Ph.D. degree from Beihang University, Beijing, China, in 2003 and 2010, respectively. He is currently an Associate Professor with the School of Computer and Information Technology, Beijing Jiaotong University, Beijing, China. His research interests include mobile edge computing, cloud computing, wireless communication and Internet of Things. He has published over 60 papers in journals, international conferences/workshops.

Shuai Ding is currently a B.S. candidate at the School of Electronic and Information Engineering, Beijing Jiaotong University, Beijing, China. His research interests include mobile edge computing and communication protocols.



Yangyang Li received the B.S. degree from Nanjing University of Information Science and Technology, Nanjing, China, and the Ph.D. degree from Beijing University of Posts and Telecommunications, Beijing, China, in 2009 and 2015, respectively. He is currently a Senior Engineer in National Engineering Laboratory for Public Safety Risk Perception and Control, Beijing, China. His research interests include mobile edge computing, cloud computing, and network security.

Wenchen Zhou received the B.S. degree from Beijing University of Technology, China, in 2016. She is currently a M.S. candidate at the School of Computer and Information Technology, Beijing Jiaotong University, Beijing, China. Her research interests include mobile edge computing and wireless communications.

Naixue Xiong is currently an Associate Professor at the Department of Mathematics and Computer Science of Northeastern State University, OK, USA. He has received his Ph.D. from Wuhan University (on sensor system engineering) and the Japan Advanced Institute of Science and Technology (on dependable sensor networks). Before he attended Northeastern State University, he worked at Georgia State University, Wentworth Technology Institution, and Colorado Technical University (as a full professor for 5 years) for approximately 10 years. His research interests include cloud computing, security and dependability, parallel and distributed computing, networks, and optimization theory. Dr. Xiong has published over 300 papers in international journal and over 100 papers presented at the international conference.