Dynamic Service Migration and Request Routing for Microservice in Multi-cell Mobile Edge Computing

Xiangyi Chen, Yuanguo Bi, *Member, IEEE,* Xueping Chen, Hai Zhao, Nan Cheng, *Member, IEEE,*
Fuliang Li, *Member, IEEE,* and Wenlin Cheng

***Abstract*—Mobile Edge Computing (MEC) sinks computation and storage capacities to network edge, where it is close to users to support delay-sensitive services. However, due to the dynamic and stochastic properties of MEC networks, the deployed services may be frequently migrated among edge servers to follow the mobility of users, which greatly increases the network operational cost. In this paper, considering the service migration cost brought by user mobility, we study the joint optimization problem of service deployment and request routing decisions to maximize the long-term network utility of MEC networks. Firstly, we propose a Lyapunov optimization-based online service migration algorithm to decompose the continuous optimization problem into a number of one-slot online optimization problems. Then, to address the NP-hard issue of one-slot optimization, we use a randomized rounding technique to implement service migration and request routing. Furthermore, through a closed-form theoretical analysis, we prove that the proposed algorithm not only greatly meets the local user requests and enables approximate performance guarantees, but also adaptively balances the service migration cost and system performance online. Finally, extensive simulations are conducted, which demonstrate that our algorithm can efficiently utilize the storage and computation resources of edge servers, and maximize the long-term network utility while ensuring the stability of service migration cost.**

***Index Terms*—Mobile edge computing, randomized rounding, request routing, service migration.**

I. Introduction

THE explosive growth of Internet of Things (IoT) devices [1] and the emergence of advanced mobile services have driven increased demands for computation-intensive and delay-sensitive services, such as online interactive game, autonomous driving [2], augmented reality [3], etc. Cisco predicts that there will be 12.3 billion IoT devices by 2022 [1]. The rapid increase of mobile services and their requirements of low latency have brought great challenges to effectively processing user requests in the centralized cloud center [4]. In order to handle a large number of real-time computational tasks and satisfy diverse Quality of Service (QoS) requirements, Mobile Edge Computing (MEC) [5] expands computation and storage resources from the cloud center to network edge. Service requests offloaded by users are run at edge servers directly connecting to base stations, eliminating the non-negligible communication delay between base stations and the cloud center in wired networks [6], [7]. With the developments of 5G/B5G and 6G technologies, MEC will be promoted more extensively and play an important role in future mobile networks [7], [8].

On the other hand, as an emerging service architecture, microservice decouples an application into multiple lightweight services and reuses functions [9], [10], which has been adopted by many large enterprises. With the characteristics of low cost, flexibility, and scalability, microservice can be deployed in MEC networks to achieve rapid response and dynamic deployments of delay-sensitive services [11], [12].

Although MEC and microservice can rapidly respond to user requests at the network edge, the dynamic and stochastic properties of MEC networks lead to frequent handovers in the edge access networks, such as unpredictable user mobility, dynamically changing service requests, time-varying wireless channels, etc. These lead to high service delay, instability, and even service interruption, seriously degrading the QoS of time-critical MEC services [13]. To follow the mobility of users, frequent migrations of services between different MEC servers can provide users with seamless real-time services when users move across base stations, but it also greatly increases the operational cost of the network, such as Wide Area Network (WAN) bandwidth and system energy consumptions [14]–[16].

In addition, an edge server has limited storage and computation capacities that are incommensurable with those of the cloud center, which means that the edge server may only install a part of the services, and cannot handle all service requests during peak periods. Moreover, with the advance of 5G technology and the widespread deployment of base stations, the density of base stations is increasing and expected to reach up to 50 base stations per square kilometer [17]. This creates a multi-cell MEC environment where users may

Manuscript received June 29, 2021; revised November 14, 2021; accepted December 20, 2021; date of current version December 31, 2021. This work was supported in part by the National Key Research and Development Program of China under Grant 2017YFE0125300, in part by the National Natural Science Foundation of China under Grant U1808207 and Grant 62171113, in part by the Fundamental Research Funds for the Central Universities of China under Grant N2116002, Grant N2124006-1, and Grant N2116014. (*Corresponding author: Yuanguo Bi.*)

Xiangyi Chen, Yuanguo Bi, Xueping Chen, Hai Zhao, Fuliang Li, and Wenlin Cheng are with the School of Computer Science and Engineering, Northeastern University, Shenyang 110819, China (e-mail: xiangyichen3746@gmail.com; biyuanguo@mail.neu.edu.cn; chenxueping996@gmail.com; haizhao310@163.com; lifuliang@cse.neu.edu.cn; wlcheng12@gmail.com).

Nan Cheng is with the State Key Lab of ISN, School of Telecommunication Engineering, Xidian University, Xi'an 710126, China (e-mail: nancheng@xidian.edu.cn).

be within the overlapped coverage of multiple base stations simultaneously. In this case, users can send their service requests to any covering base station for processing [18]. Intensive deployments and limited resources of base stations make it more difficult to properly deploy services and route edge requests.

MEC has been widely investigated recently, and the researches mainly include: i) energy-centric computing offloading policy [19], [20]; ii) resource management for improving resource utilization [21]–[23]; and iii) service optimization to improve user experience [12], [14], [24], [25]. Furthermore, to meet diverse QoS requirements of users, the network optimization issues related to edge services in MEC networks have attracted wide attention from researchers. The researches are mainly categorized into: i) service deployment in edge clouds [14], [24]–[30]; ii) service coordination among multiple edge clouds [12], [31]–[33]; and iii) joint optimization of service deployment and request routing [34]–[36]. Although the aforementioned solutions have been proposed for service optimization in MEC networks, there are still some challenging issues to be addressed. Specifically, the dynamic and stochastic properties of MEC networks, such as user mobility, time-varying service requests, bring challenges to reasonable service deployment and request routing. Besides, an edge server has limited storage and computation resources, which greatly affects network decisions and is ignored in some related works [14]. Furthermore, how to balance the network operational cost brought by service migration and network performance under resource constraints, and optimize the long-term network utility needs to be investigated further.

In this paper, we study the joint issues of service deployment and request routing at edge servers with limited storage and computation capacities, considering the cost of service migration brought by user mobility. Specifically, we propose a dynamic service migration framework to maximize the long-term system utility of MEC networks and access the cloud center as little as possible to reduce the high cost of bandwidth and delay. In the presented framework, the network operator makes two important decisions: i) service deployment decision. As the storage capacity of an edge server is limited in a multi-cell MEC network, how to make a trade-off between network utility and migration cost in a low-cost way, and migrate services reasonably to meet the diverse QoS requirements of mobile users over time? ii) request routing decision. As the computation capacity of the edge server is limited in a multi-cell MEC network, how to route service requests to maximize the system utility of an MEC network, while accessing the cloud center as little as possible?

To our best knowledge, this is the first study on long-term optimization of a multi-cell MEC network considering the storage and computation constraints of edge servers and the dynamic and stochastic properties of MEC networks. Our contributions are fourfold, and summarized as follows:

1) We develop a dynamic service migration model considering the actual characteristics of a multi-cell MEC network, under the strict constraints including service migration cost, storage capacity, and computation capacity of edge servers.
2) We propose a Lyapunov optimization-based Online Service Migration (LOSM) scheme that decomposes the continuous optimization problem into a number of one-slot online optimization problems. It not only works without any prior information such as the path of user movement and the arrival of service requests, but also achieves an adaptive balance between service migration cost and system performance.
3) To address the NP-hard issue of the one-slot optimization problem, we propose a low-complexity Randomized Rounding Optimization considering Migration cost (RROM) algorithm to solve two important decision variables including service deployment and request routing simultaneously. Furthermore, we analyze the RROM algorithm performance through Chernoff Bound Theory.
4) Through a closed-form theoretical analysis, we prove that the LOSM algorithm enables approximate performance guarantees and approaches the optimal time average utility. It not only ensures the stability of service migration cost, but also maximizes the long-term system utility of an MEC network.

The rest of this paper is organized as follows. In Section II, we briefly review the related works. The system model and problem formulation are described in Section III. In Section IV, two algorithms are proposed, and we prove that the presented algorithms achieve performance guarantees through theoretical analysis in Section V. In Section VI, we evaluate the system performance through extensive simulations, followed by the conclusions of our work in Section VII.

II. Related Work

In recent years, MEC network optimization related to mobile edge service has attracted considerable attention. Specifically, related researches mainly include service deployment [14], [24]–[30], service coordination [12], [31]–[33], and joint optimization of service deployment and request routing [34]–[36].

In terms of service deployment, a key challenge is to follow user mobility and provide seamless real-time services. For the problem, the existing works mainly include offline optimization approaches under the assumption of known future information [26], prediction-dependent optimization approaches [27], [28], and online service optimization approaches [14], [24]. In [26], a task offloading and service migration scheme under the assumption of the pre-knowledge of user mobility information is proposed to reduce energy consumption or perceived delay of mobile users. Nadembega *et al.* [27] proposed a mobility-based service migration prediction to address the trade-off between system overhead and quality of experience. Wang *et al.* [28] investigated the cost-based prediction scheme, which finds the optimal placement position of service instances by predicting future cost parameters, and minimizes the average cost over time. In [14], the performance optimization of mobile services under long-term cost budget constraints is studied. This work mainly focuses on the trade-off between service migration cost and delay, without considering the constraints of storage and computation capacities of edge

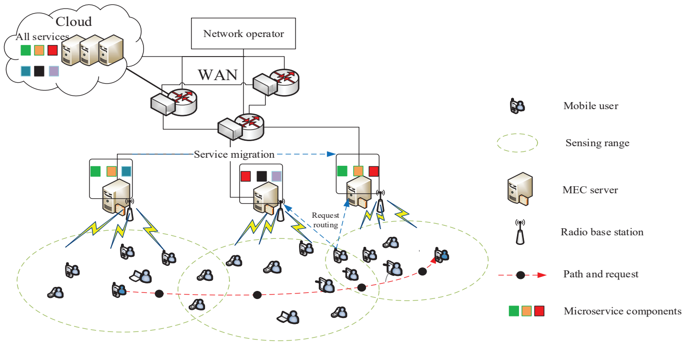


Fig. 1. Illustrations of dynamic service migration and request routing for microservices in a multi-cell mobile edge computing network.

servers. Wang *et al.* [29] formulated the problem of service migration with user mobility as a Markov decision process and approximated the state space by the distances between user and service locations. Aissioui *et al.* [30] presented an MEC architecture to support the mobility requirements of 5G vehicles.

There are some proposals working on service coordination among multiple edge clouds to handle service requests and improve user experience considering the locations of mobile users and the deployments of services in edge clouds. In [12], the problem of microservice coordination with the processing capacities of edge servers, channel conditions and the availability of service request information is investigated, and a reinforcement learning-based method is proposed to select the optimal edge cloud to reduce the overall migration cost and service delay when users move. Ma *et al.* [31] studied the demand of mobile users for virtual network functional services in MEC, and discussed the problem of user request admission considering user mobility and service delay requirements in two cases. Schneider *et al.* [33] proposed an autonomous service provisioning and coordination approach based on Deep Reinforcement Learning (DRL), which provides services by a centralized DRL agent.

Some related works focus on the researches of joint optimization of service deployment and request routing. In [34], the optimal edge services configuration that can meet both sharable (storage) and non-sharable (communication, computation) resource requirements is studied. In order to reduce resource usage and end-to-end response time, Yu *et al.* [35] studied the optimization problem of microservice instance placement and request routing by an efficient heuristic algorithm. Another similar work [36] also investigated the joint optimization problem of service deployment and request routing in MEC networks. This paper takes multi-dimensional constraints into account to minimize the number of requests routing to the cloud center, and proves that the algorithm can obtain performance approximation guarantees while violating resource constraints. However, the mobility of users over time and the long-term optimization of system performance are not considered.

Although existing solutions have been proposed to reduce user-aware latency and optimize edge services in MEC networks, there are still some challenging issues to be resolved. Firstly, the dynamic and stochastic properties of MEC networks have not been fully considered, such as unpredictable mobility of users and time-varying service requests. Secondly, considering the limited resources in MEC networks, the long-term optimization issue of system utility under the trade-off between migration cost and system performance is also ignored, which requires further investigation. In order to address these problems, we propose the dynamic service migration scheme, which aims to maximize long-term system utility of the MEC network under resource constraints of edge servers and long-term migration cost constraint. The proposed LOSM decomposes the continuous optimization problem into a number of one-slot online optimization problems by minimizing the upper bound of Drift-Minus-Utility function, and it works without any prior information such as the path of user movement and the arrival of service requests. While ensuring the stability of the system, our scheme approaches the optimal value of the time average utility, and adaptively balances the cost of service migration and system performance.

III. System Model

A. Network Model

A multi-cell MEC network is shown in Fig. 1, where users are constantly moving and may be located in the overlapped area of multiple base stations. The network model is described as $G = (B, U)$, where B and U represent the set of base stations and the set of mobile users, respectively. Specifically, $b_j \in B$, indexed by $j \in J = \{1, 2, \cdots, m\}$, and an MEC server with limited storage and computation capabilities is directly connected to a base station. Since base stations and MEC servers are closely deployed or co-located, we use $b_j \in B$ to denote the base station and its connected edge server uniformly. User $u_i \in U$, indexed by $i \in I = \{1, 2, \cdots, n\}$. Furthermore, a user delivers a service request when the service is initialized. Service $s^k = (\varsigma^k, w^k, q^k, \varpi^k, \rho^k)$, $s^k \in S$, and indexed by $k \in K = \{1, 2, \cdots, \ell\}$, where ς^k is the service size of occupying storage capacity (in bits), w^k is the required calculation intensity of service request of s^k, q^k is the size of the service request of s^k (in bits), ϖ^k is the total computational load of s^k (total CPU cycles), ρ^k is the delay requirement of the service s^k (in second), and S represents the system service library. Time slot is denoted by $t \in T = \{0,1,\cdots,\tau-1\}$. In each time slot, a user may randomly generate one of the ℓ service requests, or do not generate any service request. The service request of u_i in time slot t is expressed as $s_i^{k(t)} = (\varsigma_i^{k(t)}, w_i^{k(t)}, q_i^{k(t)}, \varpi_i^{k(t)}, \rho_i^{k(t)})$, $s_i^{k(t)} \in S(t)$, where $S(t)$ represents the set of service requests in time slot t. All service requests are routed to edge servers or the cloud center for execution. Note that the service request of s^k can be executed at an edge server only when the execution environment of s^k is configured.

The location of user u_i in time slot t is represented as $l_i(t) = (u_{i,x}(t), u_{i,y}(t))$, and base station b_j is fixed at location $l_j = (r_{j,x}, r_{j,y})$. If $\|l_j - l_i(t)\| \leqslant D_j$, the user u_i is within the coverage of base station b_j, where D_j is the covering radius of b_j. In this case, b_j is the covering base station of u_i, $b_j \in \Omega_i(t)$, where $\Omega_i(t)$ is the set of covering base stations for user u_i in time slot t. Therefore, the maximum uplink transmission rate between u_i and b_j can be calculated by

$$r_{ij}^{k(t)} = W\log_2\left[1 + \frac{P_i h_{ij}(t)}{\sigma^2 + I_i + I_j}\right] \tag{1}$$

where P_i is the transmission power of u_i, σ^2 is the noise power, and W is channel bandwidth. $h_{ij}(t) = g_{ij} d_{ij}^{-\alpha}(t)$ is the channel power gain between u_i and b_j in time slot t and is proportional to $d_{ij}^{-\alpha}(t)$, α is the path loss exponent, $d_{ij}(t) = \|l_j - l_i(t)\|$ is the distance between u_i and b_j, and g_{ij} is the coefficient of the effective channel power gain [37], [38]. In this paper, Non-Orthogonal Multiple Access (NOMA) [39] is utilized to support multiple users by sharing spectrum resources in MEC networks. While NOMA improves spectral efficiency, it also brings interference problem. Specifically, due to the difference in uploading delay of users, the completion of task uploading may be inconsistent with the decoding order of NOMA, which complicates co-channel interference [40]. As shown in (1), denote $I_i = \sum_{e\in U_j(t)\setminus\{i\}} P_e h_{ej}(t)\mathcal{I}\left(h_{ej} \geqslant h_{ij}\right)$ and $I_j = \sum_{f\in J\setminus\{j\}} \sum_{e\in U_f(t)} P_e h_{ej}(t)$ as intra-cell interference and inter-cell interference respectively, where $U_j(t)$ is the set of users served by base station b_j in time slot t [39], [41]. A 0-1 indicator function $\mathcal{I}\left(h_{ej} \geqslant h_{ij}\right)$ is utilized to indicate whether there is intra-cell interference, which is 1 only if the inequation in $\mathcal{I}\left(\right)$ is true. In NOMA, the Successive Interference Cancellation (SIC) technology is used to distinguish different users by decoding the users' signals according to the signal difference on power domain [40]. The notations and variables commonly used in our formulations are shown in Table I.

B. Decision Model

To solve the cost-utility optimization problem and follow user mobility in a multi-cell MEC network, the network operator needs to make two important decisions including request routing and service deployment. The request routing decision in time slot t is expressed as $x_{ij}(t) \in \{0,1\}$, and $x_{i\phi}(t) \in \{0,1\}$. Specifically, when the edge network accepts the service request of u_i and routes it to b_j for processing, $x_{ij}(t) = 1$. Otherwise, it is routed to the cloud center, and $x_{i\phi}(t) = 1$. On the other hand, the service deployment decision is expressed as $y_j^k(t) \in \{0,1\}$, which indicates whether to deploy the service s^k at the edge server b_j in time slot t, $b_j \in B, s^k \in S$. Due to the unpredictable mobility of users, the network operator needs to quickly deploy microservices at edge servers near mobile users to provide seamless and real-time services. However, dynamic deployment and migration of services require extra operational cost. The service migration cost of b_j is expressed as

$$c_j(t) = \sum_{k\in K} c_j^{k(t)} \Phi\left(y_j^k(t) > y_j^k(t-1)\right), \tag{2}$$

where $c_j^{k(t)}$ represents migration cost of service s^k, and the 0-1 indicator function $\Phi()$ indicates whether s^k needs to be migrated to b_j in time slot t. Only if $y_j^k(t) = 1$, $y_j^k(t-1) = 0$, which means that service s^k is not deployed in b_j in time slot $t-1$, but needs to be deployed in b_j in time slot t. Now $y_j^k(t) > y_j^k(t-1)$ is true, $\Phi\left(y_j^k(t) > y_j^k(t-1)\right) = 1$, service s^k needs to be migrated. Consequently, the edge system will migrate the service to b_j from a nearby edge server where the service is located, otherwise from the cloud center. Since local popular services are deployed in the MEC network in most cases, migration cost $c_j^{k(t)}$ is mainly proportional to the size of service s^k. The total migration cost of all base stations in time slot t is expressed as $c(t) = \sum\limits_{j\in J} c_j(t)$.

C. Problem Formulation

Denote the user set with service requests in time slot t by $I_{s(t)}$, and the request routing decision $x_{ij}(t)$ and the service deployment decision $y_j^k(t)$ for processing these service requests need to satisfy the following constraints

$$x_{ij}(t) \in \{0,1\}, \forall i \in I_{s(t)}, j \in J \cup \{\phi\}, \tag{3}$$

$$y_j^k(t) \in \{0,1\}, \forall k \in K, j \in J. \tag{4}$$

Note that our work focuses on the processing decisions in current time slot, including $x_{ij}(t)$ and $y_j^k(t)$. If the task is not completed in the current slot, its remaining part will be regarded as a new task in the next slot. The system can process it according to the service deployment decisions and request routing decisions in a new slot. The service request is only offloaded to a covering base station or the cloud center for processing, and we have

$$\sum_{j\in J\bigcup\{\phi\}} x_{ij}(t) = 1, \forall i \in I_{s(t)}. \tag{5}$$

Non-covering base stations cannot accept service requests, which can be expressed as

$$x_{ij}(t) = 0, \forall i \in I_{s(t)}, b_j \notin \Omega_i(t). \tag{6}$$

To route the service request $s_i^{k(t)}$ of user u_i to b_j, the corresponding service must be deployed in b_j, hence we have

$$x_{ij}(t) \leqslant y_j^k(t), \forall j \in J. \tag{7}$$

The total computational intensity of user requests routing to the base station b_j must not exceed its computation capacity

W_j, which is expressed as

$$\sum_{i \in I_{s(t)}} x_{ij}(t) w_i^{k(t)} \leqslant W_j, \forall j \in J. \tag{8}$$

The total size of the services stored at the base station b_j cannot exceed its storage capacity R_j, and it is given by

$$\sum_{k \in K} y_j^k(t) \varsigma_i^{k(t)} \leqslant R_j, \forall j \in J. \tag{9}$$

The execution of service request must meet the delay constraint including communication delay and computation delay, that is

$$\left(\frac{q_i^{k(t)}}{r_{ij}^{k(t)}} + \frac{\varpi_i^{k(t)}}{w_{ij}^{k(t)}} \right) x_{ij}(t) \leqslant \rho^k, \tag{10}$$

where $\frac{q_i^{k(t)}}{r_{ij}^{k(t)}}$ is communication delay of transmitting the service request $s_i^k(t)$ of user u_i to server b_j, and $\frac{\varpi_i^{k(t)}}{w_{ij}^{k(t)}}$ is the computation delay of $s_i^k(t)$ at b_j.

On the other hand, to cope with the dynamic and stochastic properties of the MEC network and provide mobile users with seamless and real-time services, the MEC operator needs to rapidly deploy and migrate microservices to an edge server nearby to follow user mobility, which evidently causes the cost of service migration. Since user mobility and service requests are time-varying and unpredictable, in actual situations, the network operator usually considers optimizing long-term system utility within a long-term cost budget. Denote the time average of long-term cost budget by $\tilde{C}$, and the service migration cost needs to satisfy the constraint

$$\lim_{\tau \to \infty} \frac{1}{\tau} \sum_{t \in T} \mathbb{E} \left\{ c_j(t) \right\} \leqslant \tilde{C}. \tag{11}$$

For an MEC network, one of the most important system performance metrics is the number of serving requests at edge servers, while meeting the constraints such as storage capacity and computation capacity. Therefore, the system utility of an MEC network can be expressed as the number of serving requests at edge servers, which represents the ability of an MEC network to process service requests, and expressed as

$$Z(x_{ij}(t)) = \sum_{j \in J} \sum_{i \in I} x_{ij}(t). \tag{12}$$

Note that network utility does not include service requests processed by the cloud center. The main goal of the MEC operator is to maximize the system utility $Z(x_{ij}(t))$ of the MEC network by making request routing decision $x_{ij}(t)$ and service deployment decision $y_j^k(t)$, that is, $\max\limits_{x,y} Z(x_{ij}(t))$. As mentioned above, there is a constraint between $x_{ij}(t)$ and $y_j^k(t)$, that is, the corresponding service must be configured before the request can be processed.

TABLE I
COMMONLY USED NOTATIONS AND VARIABLES

Notation	Description
b_j,B	Base station and the set of base stations
u_i,U	Mobile user and the set of mobile users
s^k,S	Service and the set of services
$\Omega_i(t)$	The set of covering base stations of user u_i in time slot t
$x_{ij}(t)$	Binary variable of whether the service request of user u_i is routed to the edge server b_j in time slot t
$x_{i\phi}(t)$	Binary variable of whether the service request of user u_i is routed to the cloud center in time slot t
$y_j^k(t)$	Binary variable of whether the service s^k is deployed at the edge server b_j in time slot t
$c_j^{k(t)}$	Migration cost of the service s^k to the edge server b_j
$c_j(t)$	Total migration cost of edge server b_j
$\tilde{C}$	Time average budget of migration cost
$Z(x_{ij}(t))$	System utility of servicing requests
ς^k	Service size of s^k
w^k	Required calculation intensity of s^k
q^k	Size of the service request of s^k
ϖ^k	Total computational load of s^k
ρ^k	Delay requirement of the service s^k
W_j	Total computation capacity of edge server b_j
R_j	Total storage capacity of edge server b_j
$M_j(t)$	Virtual queue of service migration cost of edge server b_j

Taking into account the mobility of users and the randomness of service requests, the long-term system utility can be characterized by the time average expectation of system utility. Therefore, the long-term system utility optimization of the MEC network is formulated as

$$\begin{aligned} \mathcal{P}1: & \max_{x,y} \lim_{\tau \to \infty} \frac{1}{\tau} \sum_{t \in T} \mathbb{E} \left\{ Z \left(x_{ij}(t) \right) \right\}, \\ & \text{s.t. (3) - (11) .} \end{aligned} \tag{13}$$

Next, there are two key issues in solving the above-mentioned problem formulation (13): i) User mobility and service requests are time-varying and unpredictable, which makes it difficult to accurately obtain some key parameters such as the set of covering base stations $\Omega_i(t)$ for user u_i, the generated service request $s_i^k(t)$ in time slot t, etc. Furthermore, long-term utility optimization of an MEC network requires constant adjustment of strategies to adapt to the dynamic changes of the network. ii) Due to strict delivery delay requirements of service requests, it is impractical to optimize the model offline. However, the online problem for a large number of mobile users and joint request routing and service deployment has high computational complexity, which brings great challenges. To address the above-mentioned issues, we propose the LOSM algorithm and RROM algorithm, which are described in detail in Section IV.

IV. Dynamic Service Migration and Request Routing

The user mobility and dynamically changing service requests have brought challenges to the operation of an MEC network. In this paper, the proposed dynamic service migration scheme aims to maximize long-term system utility of the MEC network under long-term migration cost constraint. Specifically, we define the Drift-Minus-Utility function and decompose the continuous optimization problem into a number of one-slot online optimization problems by minimizing the function upper bound. Furthermore, to address the NP-hard issue of the one-slot optimization problem, we propose a low-complexity RROM algorithm based on the optimal fractional solutions to obtain feasible solutions for service deployment and request routing. Finally, we prove that the solution returned by RROM is equal to the optimal fractional solution on the mathematical expectation, and analyze the proposed scheme through Stochastic Network Optimization [42] and Chernoff Bound Theory [43]. The proposed scheme not only ensures the stability of service migration cost, but also approaches the optimal time average utility.

A. Lyapunov Optimization-Based Online Service Migration Algorithm

To control long-term migration cost and optimize network utility, the MEC operator constructs and maintains a virtual queue $M_j(t)$ of service migration cost at each base station. It represents the excess cost of service migration at b_j by the end of time slot t. The virtual queue of migration cost is updated as

$$M_j(t+1) = \max\left[M_j(t) + c_j(t) - \tilde{C}, 0\right], \tag{14}$$

where the initial value of the queue $M_j(0) = 0$, and each b_j maintains a virtual queue.

Lemma 1: If the virtual queue of migration cost is mean rate stable, that is

$$\lim_{\tau\to\infty} \frac{1}{\tau}\mathbb{E}\left\{M_j(\tau)\right\} = 0, \tag{15}$$

then the constraint (11) can be satisfied.

Therefore, the stability of the virtual queue can ensure that the time average migration cost does not exceed the cost budget.

Proof. The virtual queue of migration cost for a base station satisfies the inequation

$$M_j(t+1) \geqslant M_j(t) + c_j(t) - \tilde{C}. \tag{16}$$

Summing all time slots $t \in T = \{0, \cdots, \tau-1\}$ from both sides of (16) and based on the Law of Telescoping Sum, we have

$$\left(M_j(\tau) - M_j(\tau-1) + M_j(\tau-1) - \cdots - M_j(0)\right) + \sum_{t\in T}\tilde{C} \geqslant \sum_{t\in T} c_j(t). \tag{17}$$

Then dividing from both sides by τ, and we have

$$\frac{1}{\tau}\left(M_j(\tau) - M_j(0)\right) + \frac{1}{\tau}\sum_{t\in T}\tilde{C} \geqslant \frac{1}{\tau}\sum_{t\in T} c_j(t). \tag{18}$$

Taking expectations of both sides and taking limit of $\tau \to \infty$ yield

$$\lim_{\tau\to\infty}\frac{1}{\tau}\mathbb{E}\left\{M_j(\tau)\right\} + \tilde{C} \geqslant \lim_{\tau\to\infty}\frac{1}{\tau}\sum_{t\in T}\mathbb{E}\left\{c_j(t)\right\}. \tag{19}$$

If $\lim_{\tau\to\infty}\frac{1}{\tau}\mathbb{E}\left\{M_j(\tau)\right\} = 0$, that is, the virtual queue is mean rate stable, the constraint (11) can be satisfied. The stability of virtual queues is proved in Theorem 2 in Section V. □

If there are m MEC servers in the multi-cell MEC network, that is, the system contains m queues, and the queue backlog vector of migration cost is expressed as

$$\mathbf{\Theta}(t) = \left([M_j(t)]_{j\in J}\right). \tag{20}$$

A quadratic Lyapunov function is given by

$$L(\mathbf{\Theta}(t)) = \frac{1}{2}\sum_{j\in J}\left[M_j(t)\right]^2. \tag{21}$$

$L(\mathbf{\Theta}(t))$ indicates queue backlog, and a small queue backlog means that the migration cost is stable. Due to the randomness of service requests and the mobility of users, we define the one-slot conditional Lyapunov drift as the conditional expectation of the difference between the Lyapunov quadratic functions of adjacent time slots, which is the expected change of the quadratic Lyapunov function in time slot t and expressed as

$$\Delta(\mathbf{\Theta}(t)) = \mathbb{E}\{L(\mathbf{\Theta}(t+1)) - L(\mathbf{\Theta}(t)) \mid \mathbf{\Theta}(t)\}. \tag{22}$$

Lemma 2: The one-slot conditional Lyapunov drift of the migration cost queue backlog has an upper bound function:

$$\Delta(\mathbf{\Theta}(t)) \leqslant \beta + \sum_{j\in J} M_j(t)\mathbb{E}\left\{[c_j(t) - \tilde{C}]|\mathbf{\Theta}(t)\right\}, \tag{23}$$

where $\beta = \frac{1}{2}m(\eta^2 + \tilde{C}^2)$ is a constant, which is independent of routing request and service deployment decisions. η is the maximum migration cost of the base station.

Proof. Squaring the two sides of (14) to get the following inequation

$$\begin{aligned}[M_j(t+1)]^2 &= \left\{\max\left[M_j(t) + c_j(t) - \tilde{C}, 0\right]\right\}^2 \\ &\leqslant [M_j(t)]^2 + [c_j(t)]^2 + \tilde{C}^2 + 2M_j(t)[c_j(t) - \tilde{C}].\end{aligned} \tag{24}$$

Therefore, according to the definition of the one-slot conditional drift of the migration cost queue backlog, the following inequation can be obtained

$$\begin{aligned}\Delta(\mathbf{\Theta}(t)) &= \mathbb{E}\left\{L(\mathbf{\Theta}(t+1)) - L(\mathbf{\Theta}(t))|\mathbf{\Theta}(t)\right\} \\ &\leqslant \mathbb{E}\left\{\beta + \sum_{j\in J}\left\{M_j(t)[c_j(t) - \tilde{C}]\right\}|\mathbf{\Theta}(t)\right\} \\ &= \beta + \sum_{j\in J} M_j(t)\mathbb{E}\left\{[c_j(t) - \tilde{C}]|\mathbf{\Theta}(t)\right\}.\end{aligned} \tag{25}$$

□

In this paper, we define the optimization objective of request routing decision and service deployment decision as the Drift-Minus-Utility function in time slot t, which is expressed as

$$\Delta(\mathbf{\Theta}(t)) - V\mathbb{E}\left\{Z\left(x_{ij}(t)\right)|\mathbf{\Theta}(t)\right\}, \tag{26}$$

where V is the control parameter of migration cost queue stability and network utility. The MEC operator can flexibly adjust V to achieve trade-off between utility and system stability according to the current queue backlog condition. And we have

$$\begin{aligned}&\Delta(\mathbf{\Theta}(t)) - V\mathbb{E}\left\{Z\left(x_{ij}(t)\right) \mid \mathbf{\Theta}(t)\right\} \\ &\quad\leqslant \beta + \sum_{j\in J} M_j(t)\mathbb{E}\left\{[c_j(t) - \tilde{C}]|\mathbf{\Theta}(t)\right\} \\ &\quad\quad - V\mathbb{E}\left\{Z\left(x_{ij}(t)\right) \mid \mathbf{\Theta}(t)\right\}.\end{aligned} \tag{27}$$

Simplify the above formula and scale the constant term, the one-slot optimization objective is simplified to minimize the expected upper bound function. The problem can be solved

Algorithm 1 Lyapunov Optimization-Based Online Service Migration

```
Input: Migration cost budget $\tilde{C}$, storage capacity constraint
    $R_j$, computation capacity constraint $W_j$;
1: Initialize $M_j(0) = 0,\ j \in J$;
2: for each time slot $t \in \{0, 1, \cdots, \tau - 1\}$ do
3:     Get the service request $S(t)$, virtual queues of migration
       cost $M_j(t)$;
4:     Solve the decision problem:
       $\min_{x,y} \sum_{j\in J} M_j(t)\{c_j(t) - \tilde{C}\} - V\{Z(x_{ij}(t))\}$ , and get
       the optimal request routing decision $x_{ij}(t)$ and service
       deployment decision $y_j^k(t)$ in the current time slot;
5:     Update the virtual queues of migration cost according
       to the definition of $M_j(t+1)$;
6: end for
Output: The virtual queues of migration cost $M_j(t+1)$ in
    time slot $t+1$;
```

with a solution that minimizes the upper bound function without expectation according to the framework of opportunistically minimizing a conditional expectation [42]. Thus, the one-slot optimization is expressed as

$$\begin{aligned}\mathcal{P}2: \min_{x,y} &\sum_{j\in J} M_j(t)\left\{c_j(t) - \tilde{C}\right\} - VZ\left(x_{ij}(t)\right), \\ &\text{s.t. } (3) \text{ - } (10).\end{aligned} \tag{28}$$

The pseudo code of LOSM is shown in Algorithm 1. The LOSM algorithm can be divided into two main steps. In step 1, LOSM constructs and maintains a virtual queue $M_j(t)$ of service migration cost at each base station to control long-term service migration cost (line 1 in Algorithm 1). In step 2, the continuous-time optimization problem is decomposed into a number of one-slot online optimization problems by defining the Drift-Minus-Utility function. Then our objective function is transformed from maximizing the total utility of the MEC network to minimizing Drift-Minus-Utility upper bound (lines 2-6 in Algorithm 1). Now our goal is to get the optimal request routing decision $x_{ij}(t)$ and service deployment decision $y_j^k(t)$ in the current time slot for the problem of $\mathcal{P}2$.

B. Randomized Rounding Optimization Considering Migration Cost

The objective of one-slot optimization contains two decision variables that are not independent. It is a mixed-integer programming problem, which has been proved to be NP-hard, and can be solved by a randomized rounding technique [36], [44]. Therefore, we propose the RROM algorithm that can be proved to achieve approximate performance guarantees.

The solution set of request routing decision variables and service deployment decision variables is represented as $(\mathbf{X}, \mathbf{Y})$. First, relax the value constraints of decision variables $x_{ij}(t)$ and $y_j^k(t)$

$$x_{ij}(t) \in \{0, 1\} \longrightarrow x_{ij}(t) \in [0, 1], \tag{29}$$

$$y_j^k(t) \in \{0, 1\} \longrightarrow y_j^k(t) \in [0, 1]. \tag{30}$$

Then, we have the expansion of the one-slot objective function (28), which is expressed as

$$\begin{aligned}&\sum_{j\in J} M_j(t)\left\{c_j(t) - \tilde{C}\right\} - VZ\left(x_{ij}(t)\right) \\ &= \Gamma + \sum_{k\in K}\sum_{j\in J}\left\{M_j(t)c_j^{k(t)}\Phi\left(y_j^k(t) > y_j^k(t-1)\right)\right\} \\ &- V\sum_{i\in I}\sum_{j\in J} x_{ij}(t),\end{aligned} \tag{31}$$

where $\Gamma = -\sum_{j\in J} M_j(t)\tilde{C}$ is a constant in time slot t.

The issue of one-slot optimization is solved in Algorithm 2, which is mainly divided into four main steps. Specifically, in step 1, the covering base station set $\Omega_i(t)$ is obtained (line 1 in Algorithm 2). In step 2, linear relaxation based on the optimal linear fitting of $\Phi\left(y_j^k(t) > y_j^k(t-1)\right)$ is solved by minimizing mean square error, which can be linearized to Φ'. Then, the optimal fractional solution set of request routing decision $x_{ij}(t)$ and service deployment decision $y_j^k(t)$ can be obtained through linear programming while satisfying the constraints (5)-(10), (29), and (30), expressed as $(\tilde{\mathbf{X}}, \tilde{\mathbf{Y}})$, where $\tilde{x}_{ij}(t) \in \tilde{\mathbf{X}}$ and $\tilde{y}_j^k(t) \in \tilde{\mathbf{Y}}$ (lines 2-3 in Algorithm 2). In step 3, the service deployment variable $y_j^k(t)$ is rounded with the probability of $\tilde{y}_j^k(t)$ (lines 4-6 in Algorithm 2). In step 4, the algorithm aims to obtain feasible solution for request routing variable $x_{ij}(t)$ by randomized rounding technique in a probabilistic approach (lines 7-21 in Algorithm 2).

We randomly round the solutions calculated by linear programming in a probabilistic approach to obtain integer values for service deployment and request routing. Our aim is to make the rounded values equal to the optimal fractional solutions on the mathematical expectation. Specifically, if the covering base stations of u_i are not deployed with the corresponding service, the request will be routed to the cloud center with probability $p_\phi^{i'}(t)$. Otherwise, the service request is routed to any edge server in $\tilde{\Omega}_i(t)$ according to the corresponding probability, where $\tilde{\Omega}_i(t)$ is the set of base stations that can handle request s_i^k. RROM first iterates through the set of base stations that can handle request. Then, according to the constraint (7) that $x_{ij}(t)$ depends on $y_j^k(t)$, the algorithm rounds the routing request variables for each base station with the probability $\frac{\tilde{x}_{ij}(t)}{\tilde{y}_j^k(t)}$. Whereas the probability of routing the service request to the cloud center is set to $p_\phi^{i''}(t)$ to improve the robustness of the system. Finally, if there are several routing variables rounded to 1 for one request, RROM randomly selects one base station to handle requests. Otherwise, for user u_i, if all routing decision variables $x_{ij}(t)$ are rounded to 0, now the request is only possibly routed to the cloud center, that is, the value of $x_{i\phi}(t)$ is taken.

When $\tilde{\Omega}_i(t) = \emptyset$, the probability of routing the request to the cloud center is expressed as

$$p_\phi^{i'}(t) = \begin{cases} 1, & \text{if } \tilde{x}_{i\phi}(t) \geqslant \prod_{b_j\in\Omega_i(t)}(1 - \tilde{y}_j^k(t)), \\ \dfrac{\tilde{x}_{i\phi}(t)}{\prod_{b_j\in\Omega_i(t)}(1 - \tilde{y}_j^k(t))}, & \text{otherwise} . \end{cases} \tag{32}$$

Algorithm 2 Randomized Rounding Optimization Considering Migration Cost

```
Input: Migration cost budget $\tilde{C}$, storage capacity constraint $R_j$, computation capacity constraint $W_j$;
 1: Get the service request $s_i^k(t)$ in time slot $t$, the covering base station $\Omega_i(t)$;
 2: Linear relaxation of request routing decision variables and service deployment decision variables $x_{ij}(t) \in \{0,1\} \rightarrow x_{ij}(t) \in [0,1]$, $y_j^k(t) \in \{0,1\} \rightarrow y_j^k(t) \in [0,1]$, and linearize $\Phi(y_j^k(t) > y_j^k(t-1))$ into $\Phi' \triangleq \left(y_j^k(t) - y_j^k(t-1)\right)/2 + 1/4$
    ;
 3: Solve the request routing and service migration model satisfying the constraints (5)-(10), (29), and (30) by linear programming, and get solution set $(\tilde{\mathbf{X}}, \tilde{\mathbf{Y}})$;
 4: for $b_j \in B, s^k \in S$ do
 5:     Set $y_j^k(t) = 1$ with the probability $\tilde{y}_j^k(t)$;
 6: end for
 7: for $u_i \in U$ do
 8:     $\tilde{\Omega}_i(t) = \{b_j \in \Omega_i(t)\} \bigcap \{b_j \mid y_j^k = 1, s^k = s_i^k(t)\}$;
 9:     if $\tilde{\Omega}_i(t) = \emptyset$ then
10:         Set $x_{i\phi}(t) = 1$ with the probability $p_{\phi}^{i'}(t)$;
11:     else
12:         Set $x_{i\phi}(t) = 1$ with the probability $p_{\phi}^{i''}(t)$;
13:         for $b_j \in \tilde{\Omega}_i(t)$ do
14:             Set $x_{ij}(t) = 1$ with the probability $\frac{\tilde{x}_{ij}(t)}{\tilde{y}_j^k(t)}$;
15:         end for
16:         All $x_{ij}(t) = 1$, $b_j \in \tilde{\Omega}_i(t)$, randomly and uniformly select one to execute;
17:         if all $x_{ij}(t) = 0$, $b_j \in \tilde{\Omega}_i(t)$ then
18:             Get the value of $x_{i\phi}(t)$;
19:         end if
20:     end if
21: end for
Output: Solution set $(\mathbf{X}, \mathbf{Y})$;
```

However, when $\tilde{\Omega}_i(t) \neq \emptyset$, the probability of routing the request to the cloud center depends on $\tilde{x}_{ij}(t)$ and $\tilde{y}_j^k(t)$. It is larger than 0 and expressed as

$$p_{\phi}^{i''}(t) = \left[\frac{\tilde{x}_{i\phi}(t) - \prod_{b_j \in \Omega_i(t)} (1 - \tilde{y}_j^k(t))}{1 - \prod_{b_j \in \Omega_i(t)} (1 - \tilde{y}_j^k(t))}\right]_+ . \tag{33}$$

Suppose the solution obtained by the one-slot linear programming is denoted by $\tilde{P}2\left(\tilde{x}_{ij}(t), \tilde{y}_j^k(t)\right)$, the solution obtained by RROM is denoted by $\hat{P}2\left(\hat{x}_{ij}(t), \hat{y}_j^k(t)\right)$. Therefore, we can have Lemma 3.

Lemma 3: The solution returned by RROM algorithm is equal to the optimal fractional solution on the mathematical expectation.

The detailed proof of Lemma 3 is given in Appendix A. Based on aforementioned descriptions about RROM, we analyze the performance of the algorithm through Lemma 3, which is inspired by the application of a randomized rounding technique to these issues [36]. Furthermore, our main contribution is to extend the one-slot problem to the long-term optimization model of network performance. We prove that LOSM has the approximate performance guarantees and approaches optimal time average utility by rigorous theoretical analysis. Moreover, under ensuring the stability of service migration cost, LOSM can adaptively balance system performance and service migration cost in an online manner over time. The analysis process is described in detail in Section V.

We utilize asymptotic computation time complexity notation $O(*)$ to analyze the complexity of different execution steps and sub-steps in Algorithm 1 and 2. Algorithm 1 mainly consists of two steps. In step 1, the construction of virtual queues has a time complexity of $O(m)$. In step 2, the continuous optimization problem is decomposed and its complexity depends on the computational complexity of one-slot optimization.

The issue of one-slot optimization in Algorithm 1 is solved in Algorithm 2, which is mainly divided into four main steps. Specifically, in step 1, obtaining the covering base station set $\Omega_i(t)$ has a time complexity of $O(mn)$. In step 2, linear relaxation and solving the request routing and service migration model by linear programming solving tools of Interior Point Method have a time complexity of $O((mn + m\ell + n)^{2.055})$ [45], where $mn + m\ell + n$ is the number of variables in our algorithm. The step 3 of Algorithm 2 is to update service deployment decisions and has a complexity of $O(m\ell)$. In step 4, the feasible solutions for request routing are obtained by random rounding, which has a time complexity of $O(mn)$. Therefore, the total computational complexity of Algorithm 2 is reduced to $O((\max(mn, m\ell))^{2.055})$, and the total computational complexity of Algorithm 1 is $O(\tau(\max(mn, m\ell))^{2.055})$.

Next, we analyze the convergence of the algorithm. According to the convergence theory of Lyapunov optimization [42], [46] and the Supporting Hyperplane Theorem [47], we can give the following convergence of LOSM. Specifically, define $\varepsilon = 1/V(\varepsilon > 0)$, then for all $\tau \geqslant 1/\varepsilon^2$, we have

$$\begin{aligned} &\bar{Z}(\tau) \geqslant \xi^* - O(\varepsilon), \\ &\bar{c}_j(\tau) \leqslant \tilde{C} + O(\varepsilon), \quad \forall j \in J, \end{aligned} \tag{34}$$

where $\bar{Z}(\tau) = \frac{1}{\tau} \sum\limits_{t \in T} \mathbb{E}\{Z(t)\}$ and $\bar{c}_j(\tau) = \frac{1}{\tau} \sum\limits_{t \in T} \mathbb{E}\{c_j(t)\}$. Therefore, LOSM provides an $O(\varepsilon)$ approximation with convergence time $O(1/\varepsilon^2)$. The detailed proof of the convergence of LOSM is given in Appendix B.

V. Performance Analysis

We discuss the performance of the proposed framework through a closed-form theoretical analysis in this section. Firstly, we analyze the gap between the solution returned by RROM based on random rounding and the optimal solution in a time slot. Then, the performance gap between the solution returned by LOSM and the optimal solution with long-term system utility is analyzed. Theoretical analysis proves that LOSM has an approximate performance guarantees while ensuring the stability of service migration cost.

A. Randomized Rounding

Through Lemma 3, we prove that the solution returned by RROM is equal to the optimal fractional solution on the mathematical expectation. The optimal solution of $\mathcal{P}2$ is denoted by $P2^*\left(x_{ij}^*(t), y_j^{k*}(t)\right)$. Then we specifically discuss the gap between the solution returned by RROM algorithm and the optimal solution through Chernoff Bound Theory [43].

Theorem 1: The gap between the solution returned by RROM algorithm $\hat{P}2$ and the optimal solution $P2^*$ can be expressed as

$$\begin{aligned}\hat{P}2 - P2^* \leqslant & \left(1 + \frac{3\ln(r)}{\mu} + \sqrt{\frac{6\ln(r)}{\mu}}\right) \\ & \cdot \frac{m}{2}\left(M_j(t)\right)_{\max}\left(R_j\right)_{\max}\frac{c_{\max}}{\left(\varsigma^k\right)_{\min}} \\ & + mV\left(W_j\right)_{\max}\frac{1}{\left(w^k\right)_{\min}} \triangleq \Lambda,\end{aligned} \tag{35}$$

where μ is the minimum of $\sum\limits_{k\in K}\tilde{y}_j^k(t)c_j^{k(t)}$, r is the number of service requests, $c_{\max}$ is the maximum of $c_j^{k(t)}$. The detailed proof of Theorem 1 is given in Appendix C.

B. Online Service Migration

The goal of the proposed framework is to optimize the long-term system utility of the MEC network. Based on the performance gap in a time slot in Theorem 1, we give the infimum bound of the time-average network utility and the supremum bound of the time-average migration cost queue backlogs of LOSM in Theorem 2. We use a Λ-additive approximation in LOSM in every slot t, then the performance and stability guarantees can be provided in Theorem 2.

Theorem 2: **(a) Performance Guarantees:** The gap between the solution returned by LOSM and the optimal solution can be expressed as

$$\liminf_{\tau\to\infty}\frac{1}{\tau}\sum_{t\in T}\mathbb{E}\left\{Z\left(x_{ij}(t)\right)\right\} \geqslant \xi^* - \frac{\beta+\Lambda}{V}. \tag{36}$$

(b) Stability Guarantee

LOSM has system stability guarantees, including mean rate stability and strong stability of migration cost queue.

(i) Mean rate stability of migration cost queue: All queues of $M_j(t)$ are mean rate stable, that is, satisfy (15) in Lemma 1.

The mean rate stability ensures that the increasing rate of queue backlog expectation will not exceed the increasing rate of time. It shows that the queue backlog increases at a low rate, and guarantees the constraint (11).

(ii) Strong stability of migration cost queue: The queue of migration cost is strongly stable, and satisfies the following inequation

$$\limsup_{\tau\to\infty}\frac{1}{\tau}\sum_{t\in T}\sum_{j\in J}\mathbb{E}\left\{M_j(t)\right\} \leqslant \frac{\beta+\Lambda+V[\xi^*-\psi(\delta)]}{\delta}, \tag{37}$$

where β, V, Λ, δ are positive constants, $\psi(\delta)$ is the solution between the minimum value and the maximum value of the original objective function (expectation of system utility), and ξ^* is the optimal time average utility.

The strong stability indicates that the time-average total queue backlog has an upper bound, which strictly limits the increasing rate of total migration cost.

TABLE II
SIMULATION PARAMETERS

Parameter	Value	Parameter	Value
m	16	W_j	1-100 GHz
n	200-1600	R_j	5-300 GB
ℓ	500	$\tilde{C}$	45
α	4	W	4 MHz
P_i	1 W	τ	1000

For Theorem 2, we prove that the time-average expected utility satisfies (36), which indicates that the Drift-Minus-Utility method of LOSM can approach the optimal control of the system utility with the increase of parameter V. Besides, the time-average expected value of queue backlog bound increases linearly with V, as shown in (37). This presents a utility-migration cost trade-off of $[O(1/V), O(V)]$, hence the network operator can set an appropriate value of V to balance the long-term utility and migration cost. The detailed proof of Theorem 2 is given in Appendix D.

VI. Performance Evaluation

To evaluate algorithm performance of LOSM, we evaluate network utility under long-term migration cost constraints by conducting extensive simulations and varying some important simulation parameters. Based on the solution of RROM for solving the one-slot problem, we conduct a comprehensive performance evaluation of LOSM under long-term circumstances.

A. Simulation Setup

In the simulations, 16 MEC edge servers are regularly deployed, and there are up to 1600 mobile users randomly distributed and constantly moving in the simulated environment. Supposing that each mobile user is in the coverage of at least one base station in a time slot. The benchmark of storage capacity is 150 GB, and the benchmark computation capacity is 40 GHz. In order to fully evaluate the performance of LOSM, we conduct simulations under different storage capacities and computation capacities, and each MEC server is equipped with multiple CPU cores [48]. Considering the diversity of service types, we set up 500 microservice components that contain different types of services, including delay-sensitive and computation-intensive services, such as online video game, autonomous driving, face recognition and augmented reality. Specifically, services such as online video game and augmented reality require more storage capacity, while autonomous driving services require more computation intensity. The computation intensity is uniformly distributed within [0.1, 2.5] GHz. Moreover, to simulate actual situations, we use ONE simulator [49] to generate users' moving trajectories based on two widely-used movement models, including map-based movement model (70% of users) and random walk model (30% of users) to comprehensively evaluate algorithm performance [14], [50]. In our simulations, we set up 1000 edge users, where 30% of them are vehicles with a speed of 2.7-13.9 m/s and 70% of them are pedestrians with a speed of 0.5-1.5 m/s. The number of time slot we simulated is 1000, in which users will not move from the coverage of one base station to that of another base station in a time slot. Users randomly generate service requests, and the important simulation parameters are shown in Table II. To evaluate the performance of LOSM, we compare it with the existing algorithms.

1) Optimal Fractional Solution: it uses the optimal fractional solution obtained by the solver linprog with linear programming model [51].
2) Greedy-Based Utility Maximize (GBUM) [52]: GBUM sequentially deploys services to meet the maximum number of local user requests. Then, a service request is sent to the nearest edge server for processing even though multiple edge servers that can provide service.
3) No Service Migration (NSM): NSM randomly deploys services and leaves the initial allocation policy unchanged. It has no migration cost, but at the cost of routing more service requests to the cloud center.
4) Low Cost Priority Migration (LCPM): LCPM sequentially deploys the lowest-cost services to satisfy local user requests. Meanwhile, service requests are randomly routed to an edge server that can provide service for processing.
5) Maximize Volume of Data Offloaded (MVDO): MVDO deploys services in the sequence which can bring maximum volume of data offloaded to the MEC network. Similarly, a service request is sent to the nearest edge server for processing.

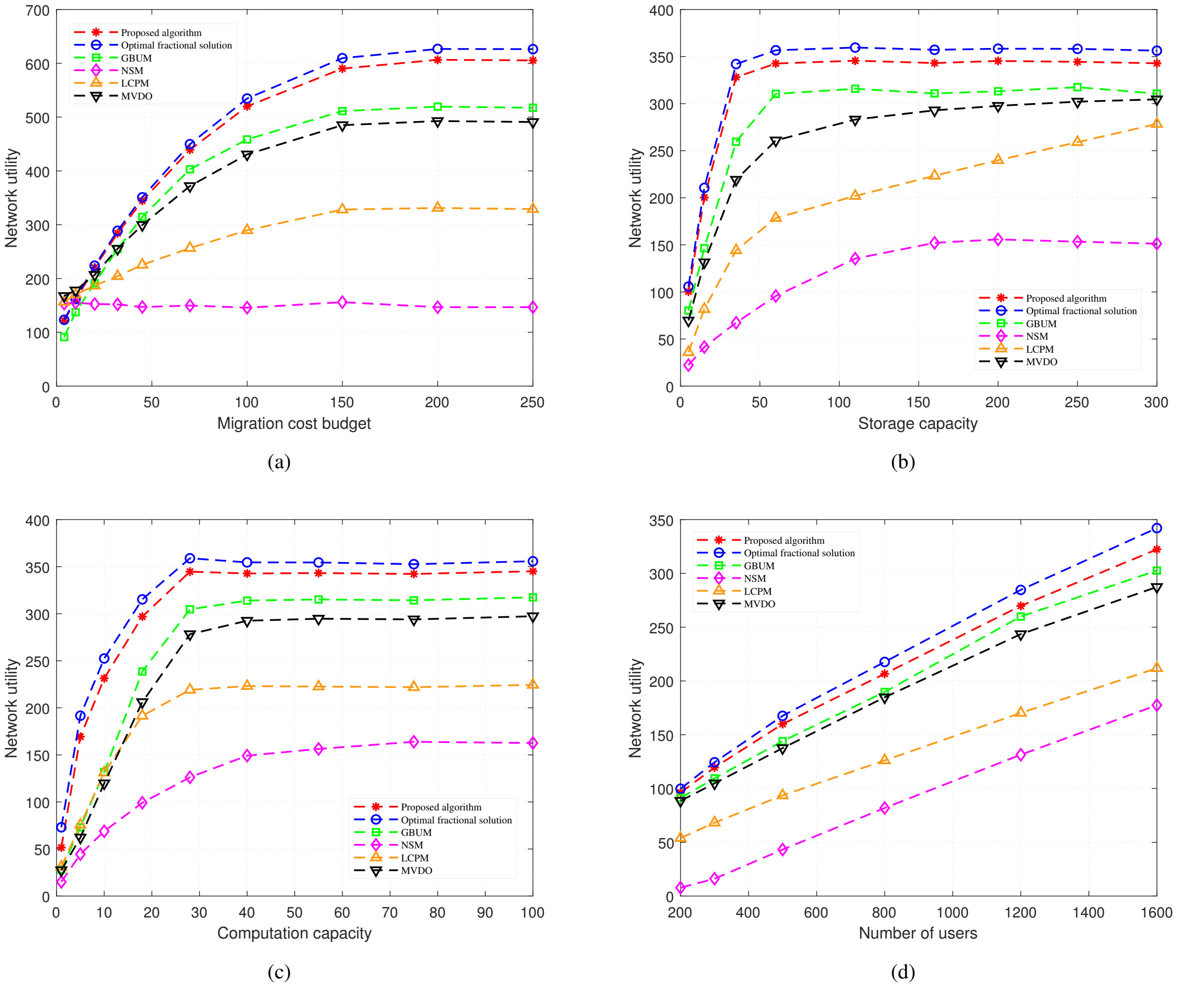
Note that the solution obtained by the optimal fractional solution is not feasible in practice, because it may not meet the variable constraint of request routing decision $x_{ij}(t) \in \{0,1\}$, or the variable constraint of service deployment decision $y_j^k(t) \in \{0,1\}$. However, it provides the reference value of the integer solution, which is also the target value that our algorithm needs to approach. Therefore, we would not elaborate on the optimal fractional solution in the following performance comparisons.

B. Simulation Results

To evaluate the performance of LOSM, we select multiple performance metrics including network utility and the amount of data offloaded. Furthermore, we study the impact of control parameter V on network utility and migration cost. Besides, the feasibility of LOSM is also evaluated through the statistics of resource utilization and running time.

1) Network utility: For an MEC network, a key performance metric is the network utility of serving requests at edge servers. Network utility is characterized as the average number of serving requests at edge servers, which is also essential for the network operator to deploy and operate MEC networks. Fig. 2 shows the network utility of the MEC network with different values of migration cost budget, storage capacity, computation capacity, and number of users. We can observe that the network utility of LOSM is higher than those of the comparison algorithms under different migration cost budgets from Fig. 2(a). When the migration cost is 200, the network utility of LOSM is 606.55, which achieves 23.11% gains than MVDO, and 16.76% gains than GBUM. Furthermore, our algorithm achieves the best storage capacity adaptability over the comparison algorithms, as shown in Fig. 2(b). Besides, when the computation capacity is 40 GHz, LOSM gains 9.18% over GBUM, and gains 17.17% over MVDO in Fig. 2(c). The results of network utility versus the number of users are shown in Fig. 2(d). With the increasing number of users, the network utility of each algorithm increases since it efficiently makes use of MEC network resources. Among 16 edge servers deployed in the simulations, when the number of users increases from 500 to 1200, the network utility of LOSM increases by 68.48%, and achieves the highest network utility compared with the comparison algorithms. The above simulation results demonstrate that our algorithm obtains the highest network utility compared to the comparison algorithms. This is because LOSM operates Lyapunov optimization, which decomposes the continuous optimization problem into a number of one-slot online optimization problems by minimizing Drift-Minus-Utility upper bound. LOSM achieves the long-term optimization of network utility by controlling migration cost queue. Furthermore, LOSM uses a randomized rounding method based on the optimal fractional solution, and the returned solution is equal to the optimal fractional solution on the mathematical expectation. Therefore, the returned solution by our scheme approaches the optimal fractional solution and achieves a high network utility.

2) Amount of data offloaded: On the other hand, the amount of data offloaded is also one of the essential metrics. It also represents the ability of an MEC network to process service requests. Therefore, we investigate the amount of data offloaded with different values of migration cost budget, storage capacity, computation capacity, and number of users, as shown in Fig. 3. From Fig. 3(a), with the increase of migration cost budget, LOSM performs better than other algorithms on the amount of data offloaded. Specifically, when migration cost is 200, LOSM achieves 22.26% gains over GBUM and 13.55% gains over MVDO on the the amount of data offloaded. Besides, as shown in Fig. 3(b), LOSM achieves gains more than 9.03% over GBUM, LCPM, and NSM when storage capacity is 250 GB. When the computation capacity is 28 GHz, the amount of data offloaded to edge servers of LOSM is 418.50, and it is 56.52% higher than LCPM in Fig. 3(c). Moreover, in Fig. 3(d) when the number of users reaches 1200, LOSM approaches MVDO and achieves 438.29 of the amount of data offloaded, which achieves gains more than 8.34% over other three comparison algorithms including LCPM, GBUM, and NSM. Overall, MVDO achieves the best performance results on the amount of data offloaded, because MVDO performs service deployment and request routing decisions based on the idea of maximizing the amount of data offloaded. Nevertheless, compared with the other three algorithms, LOSM has obvious advantages. This is because LOSM achieves the long-term optimization of network and performs randomized rounding according to the optimal fractional solution. It approaches the optimal solution and reasonably optimizes network resources, and greatly increases the amount of data offloaded to edge servers.



(a)

(b)

(c)

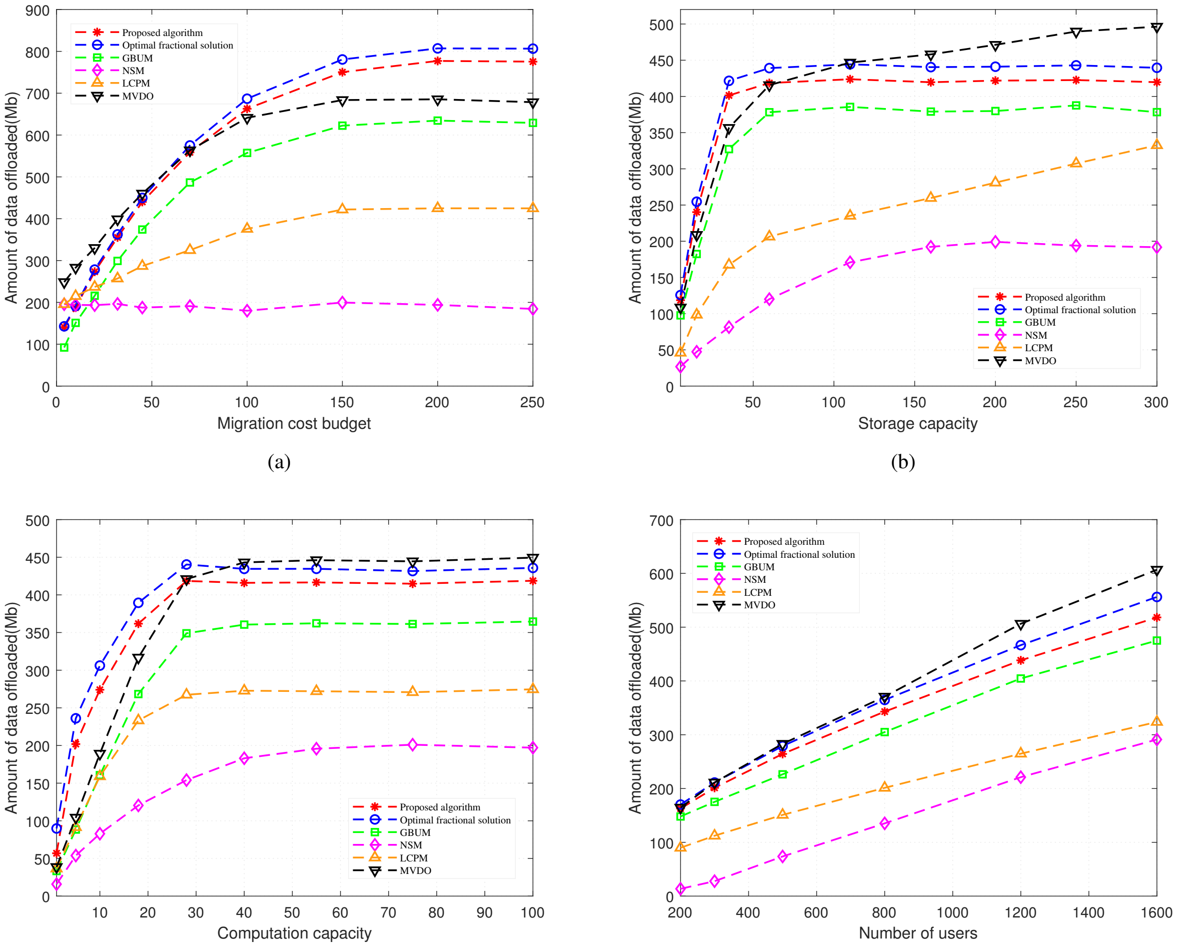
(d)

Fig. 2. (a) Network utility under different migration cost budgets, (b) Network utility under different storage capacities, (c) Network utility under different computation capacities, (d) Network utility under different number of users.

3) Control parameter: Control parameter V is a key parameter for the network operator to operate MEC networks, and we evaluate the impact of different values of V on system performance and migration cost. Firstly, as shown in Fig. 4, we verify the convergence of network utility and migration cost with time slot t under different V. From Fig. 4(a), we can observe that the network utility gradually decreases and stabilizes with the increase of time slots, the system convergence time also goes up with the increase of V. Specifically, when V is 2500, the network utility reaches the steady state with the value of 346, and the time slot is around 75. Furthermore, the results of migration cost with time slots under different V are shown in Fig. 4(b). Similarly, migration cost gradually decreases and converges to the migration cost budget. Specifically, when V is 5000, the time slot of the migration cost reaching the steady state is around 120. The main reason for violating migration cost constraint is insufficient time slots and the system does not reach the stable state. Moreover, the convergence of migration cost also proves the strong stability of migration cost queues in Theorem 2 (b), which indicates that the time-average total queue backlog has an upper bound and limits the increasing rate of total migration cost.

The network utility versus different values of V is shown in Fig. 5(a). With the increase of V, the network utility of LOSM also goes up and tends to be stable. When V is 0.5, the network utility is 159.21, while V increases to 140, the network utility increases to 309.13. Similarly, in terms of the amount of data offloaded, LOSM approximates the relaxed Optimal fractional solution algorithm in Fig. 5(b). With control parameter V, LOSM can adaptively adjust the migration cost and network performance according to the current network state. Simulation results show that increasing the value of V could optimize network performance, which is one of the most important parameters in our system design. Generally, simulation results are consistent with Theorem 2 in Section V.

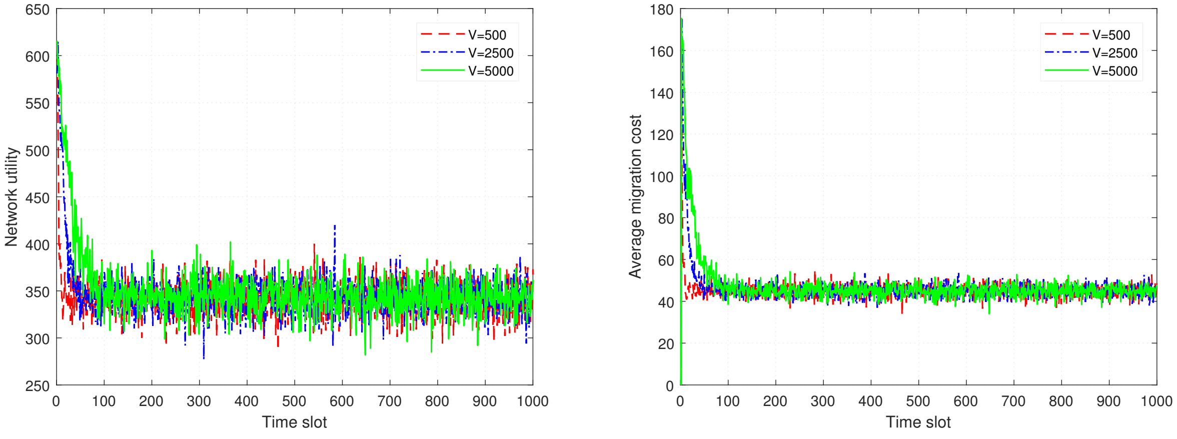
Simulation results of the average queue backlog versus different values of V are shown in Fig. 6(a). We can see that the average queue backlog increases linearly with the increase of V, which is consistent with the strong stability of the migration cost queue in Theorem 2. Specifically, (37) shows the upper bound of the time-average expected value of queue backlog increases linearly with V. Combined with Fig. 5(a), simulation results in Fig. 6(a) confirm the theoretical analysis in Section



(a) (b)

(c) (d)

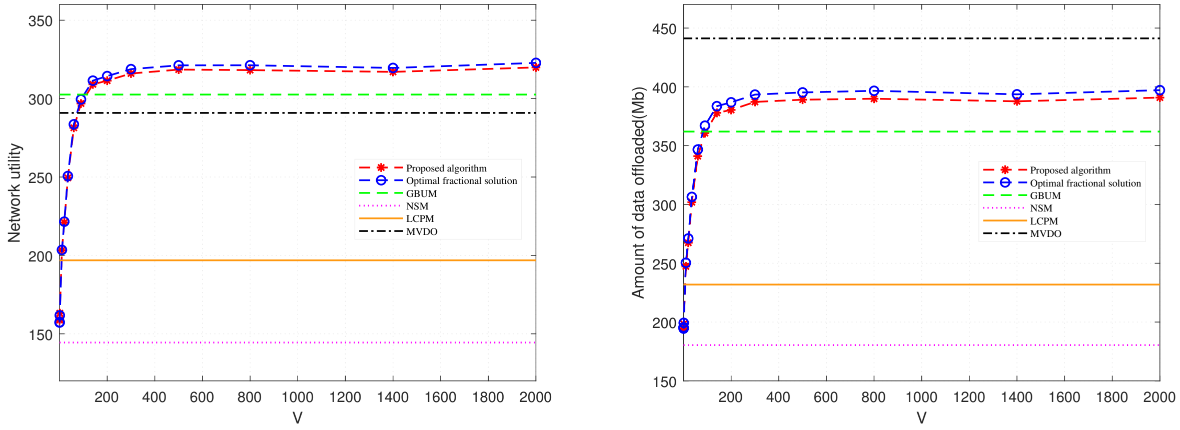
Fig. 3. (a) Amount of data offloaded under different migration cost budgets, (b) Amount of data offloaded under different storage capacities, (c) Amount of data offloaded under different computation capacities, (d) Amount of data offloaded under different number of users.



(a) (b)

Fig. 4. (a) Network utility with time slot, (b) Average migration cost with time slot.

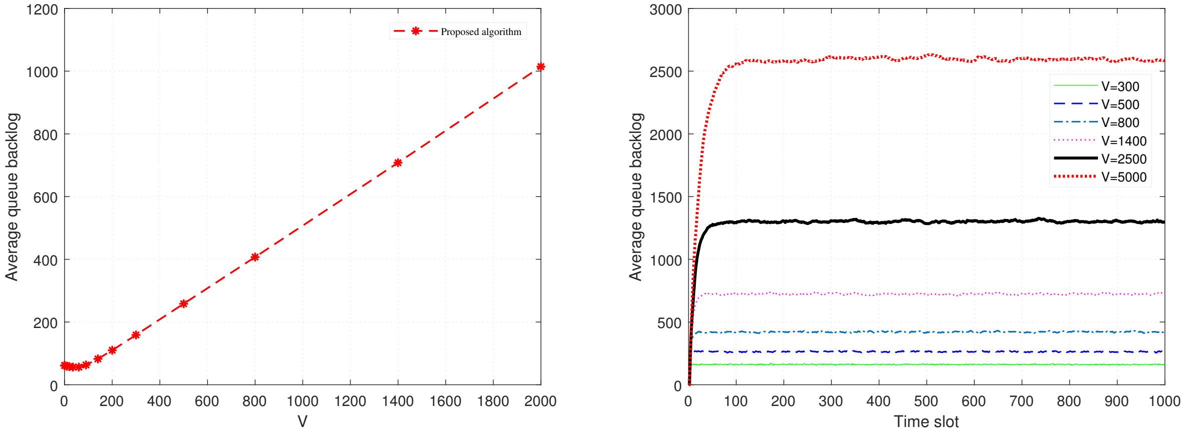
V, that is, the control parameter V can adjust the trade-off between utility and migration cost in $[O(1/V), O(V)]$. Fig. 6(b) shows the variation curve of average queue backlogs with time slots versus different values of V. Although the different values of V cause the difference of queue backlog, the queue backlog fluctuates around a certain value, which reflects the adjustment of the migration cost budget and the stability of cost queues. This is because the strong stability



(a)

(b)

Fig. 5. (a) Network utility under different values of V, (b) Number of serving requests at edge servers under different values of V.



(a)

(b)

Fig. 6. (a) Average queue backlog under different values of V, (b) Average queue backlog with time slot.

ensures that there is an upper bound for the time-average total queue backlog in Theorem 2 (b).

4) Resource utilization and running time: Furthermore, we evaluate the resource utilization of different algorithms, which reflects whether the algorithms fully make use of storage resources and computation resources of edge servers. The storage capacity is 150 GB and the computation capacity is 40 GHz, respectively. Simulations are run on Intel(R) Xeon(R) Gold 6140 CPU@2.3 GHz processor with 192.0 GB of RAM. The storage and computation resource utilizations of 16 edge servers are shown in Fig. 7. Specifically, the average storage resource utilization of LOSM is 84.74%, and the average computation resource utilization is as high as 70.89%. The average utilization of computation resource in LOSM is significantly higher than MVDO and LCPM with gains 10.96% and 53.61%, respectively. Simulation results demonstrate that LOSM fully makes use of the resources of edge servers. This is because LOSM maximizes the network utility of MEC processing requests, and greatly reduces requests routed to the cloud center. It reasonably optimizes network resources and improves the resource utilization of edge servers. The average running time of each algorithm is shown in Table III. When the number of users is 1000, the average running time of LOSM is 3.88 s, while the average running time of MVDO is as high as 88.27 s. This is because LOSM randomly rounds the optimal fractional solution obtained by solving linear programming, reducing iterations of base stations or services that must be executed in a greedy-based algorithm. Therefore, the proposed algorithm greatly reduces running time, which is critical in the MEC network with real-time requirements.

VII. Conclusion

In this paper, we have studied the long-term utility maximization issue in MEC networks considering service migration cost, where edge servers have limited storage capacity and computation capacity. Lyapunov optimization has been utilized to decompose the continuous optimization problem, and a randomized rounding technique has been used to solve the service deployment and request routing problems of a single time slot. Furthermore, we have analyzed the performance of

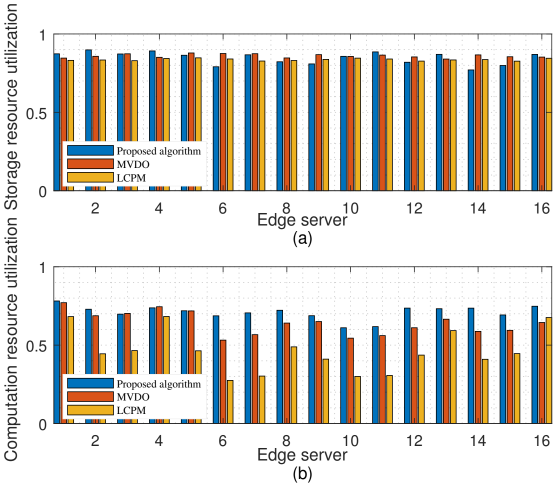


Fig. 7. (a) Storage resource utilization, (b) Computation resource utilization.

TABLE III
RUNNING TIME (S)

Number of Users	LOSM	OFS	GBUM	LCPM	MVDO
200	0.28	0.25	13.58	1.36	16.91
500	0.83	0.82	38.02	3.72	41.64
800	1.74	1.67	56.60	7.83	62.59
1000	3.88	3.87	69.38	17.29	88.27
1200	5.87	5.84	81.11	26.35	92.64

our algorithm theoretically, and have proved that the algorithm enables approximate performance guarantees. Finally, through sufficient simulations, the performance of our algorithm has been comprehensively evaluated in terms of network utility, the amount of data offloaded to edge servers, resource utilization, etc. Simulation results have demonstrated that LOSM can not only efficiently utilize the storage and computation resources of edge servers, but also optimize the long-term system utility of the MEC network while ensuring the stability of service migration cost. For future research directions, we will consider load balancing of multiple edge servers and the computing paradigm considering community relations in MEC.

APPENDIX A
PROOF OF LEMMA 3

Proof. The expected value of the solution returned by the RROM algorithm can be expressed as

$$\begin{aligned}&\mathbb{E}\left\{\hat{P}2\left(\hat{x}_{ij}(t),\hat{y}_j^k(t)\right)\right\}=\Gamma+\frac{1}{2}\cdot\\&\sum_{k\in K}\sum_{j\in J}\left(M_j(t)c_j^{k(t)}\left[\left(\Pr\left[\hat{y}_j^k(t)=1\right]\right)-y_j^k(t-1)+\frac{1}{2}\right]\right)\\&-\sum_{i\in I}\sum_{j\in J}V\Pr\left[\hat{x}_{ij}(t)=1\right].\end{aligned}\tag{38}$$

For $\Pr\left[\hat{y}_j^k(t)=1\right]$ in (38), according to line 5 in Algorithm 2, we can get

$$\Pr\left[\hat{y}_j^k(t)=1\right]=\tilde{y}_j^k(t).\tag{39}$$

Then, for $\Pr\left[\hat{x}_{ij}(t)=1\right]$ in (38), we can infer it under two situations. In the first case, the request is routed to the base station for processing. Now, when $y_j^k(t)=1$, $\Pr[y_j^k(t)=1]=\tilde{y}_j^k(t)$, and the conditional probability $\Pr[\hat{x}_{ij}(t)=1|y_j^k(t)=1]=\frac{\tilde{x}_{ij}(t)}{\tilde{y}_j^k(t)}$ according to line 14 in Algorithm 2. Otherwise, when $y_j^k(t)=0$, $\Pr[\hat{x}_{ij}(t)=1|y_j^k(t)=0]=0$. Based on the Total Probability Theorem, we have

$$\Pr[\hat{x}_{ij}(t)=1]=\frac{\tilde{x}_{ij}(t)}{\tilde{y}_j^k(t)}\tilde{y}_j^k(t)=\tilde{x}_{ij}(t).\tag{40}$$

In the second case, the request is routed to the cloud center for processing. When $\tilde{\Omega}_i(t)=\emptyset$, $\Pr[\tilde{\Omega}_i(t)=\emptyset]=\prod_{b_j\in\Omega_i(t)}(1-\tilde{y}_j^k(t))$ and $\Pr[\hat{x}_{i\phi}(t)=1|\tilde{\Omega}_i(t)=\emptyset]=p_\phi^{i'}(t)$ according to line 10 in Algorithm 2. Otherwise, when $\tilde{\Omega}_i(t)\neq\emptyset$, $\Pr[\tilde{\Omega}_i(t)\neq\emptyset]=1-\prod_{b_j\in\Omega_i(t)}(1-\tilde{y}_j^k(t))$ and $\Pr[\hat{x}_{i\phi}(t)=1|\tilde{\Omega}_i(t)\neq\emptyset]=p_\phi^{i''}(t)$ according to line 12 in Algorithm 2. Based on the Total Probability Theorem, we have

$$\begin{aligned}\Pr[\hat{x}_{i\phi}(t)=1]=p_\phi^{i'}(t)\prod_{b_j\in\Omega_i(t)}(1-\tilde{y}_j^k(t))+\\p_\phi^{i''}(t)\left(1-\prod_{b_j\in\Omega_i(t)}(1-\tilde{y}_j^k(t))\right)=\tilde{x}_{ij}(t).\end{aligned}\tag{41}$$

Combining (40) and (41) yields $\Pr[\hat{x}_{ij}=1]=\tilde{x}_{ij}(t)$. Next, (38) can be simplified as

$$\begin{aligned}\Gamma+\frac{1}{2}\sum_{k\in K}\sum_{j\in J}\left(M_j(t)c_j^{k(t)}\left[\left(\tilde{y}_j^k(t)\right)-y_j^k(t-1)+\frac{1}{2}\right]\right)\\-\sum_{i\in I}\sum_{j\in J}V\tilde{x}_{ij}(t)=\tilde{P}2\left(\tilde{x}_{ij}(t),\tilde{y}_j^k(t)\right).\end{aligned}\tag{42}$$

Hence, Lemma 3 is proved. □

APPENDIX B
PROOF OF CONVERGENCE

Proof. According to the convergence theory of Lyapunov optimization [42], [46], we construct an equivalent minimization problem

$$\begin{aligned}\min\quad&-Z,\\\text{s.t.}\quad&c_j\leqslant\tilde{C},\forall j\in J,\\&(-Z,c_1,c_2,\ldots,c_m)\in\bar{\mathcal{R}},\end{aligned}\tag{43}$$

where $Z=\mathbb{E}\{Z(t)\}$, $c_j=\mathbb{E}\{c_j(t)\}$, and $\bar{\mathcal{R}}$ is the closure of the set of all expectation vectors $(-Z,c_1,c_2,\ldots,c_m)$ that can be obtained. According to the Supporting Hyperplane Theorem [47], there exists a supporting hyperplane that passes through the closure point $(-\xi^*,\tilde{C},\ldots,\tilde{C})$ of $\bar{\mathcal{R}}$, and a non-negative vector $\Upsilon=(\gamma_0,\gamma_1,\ldots,\gamma_m)$ (i.e. the normal vector of the supporting hyperplane), which satisfy the following formula

$$\gamma_0(-Z)+\sum_{j\in J}\gamma_jc_j\geqslant\gamma_0(-\xi^*)+\sum_{j\in J}\gamma_j\tilde{C}.\tag{44}$$

The supporting hyperplane is non-vertical if $\gamma_0 \neq 0$ [47], define $\mu_k = \gamma_k / \gamma_0$ and we have

$$-Z + \sum_{j \in J} \mu_j c_j \geqslant -\xi^* + \sum_{j \in J} \mu_j \tilde{C}, \tag{45}$$

where the non-negative vector $(\mu_1, \mu_2, \ldots, \mu_m)$ is called a Lagrange multiplier vector [46]. $\bar{Z}(\tau) = \frac{1}{\tau} \sum_{t \in T} \mathbb{E}\{Z(t)\}$ is the time average of Z, $\bar{c}_j(\tau) = \frac{1}{\tau} \sum_{t \in T} \mathbb{E}\{c_j(t)\}$ is the time average of c_j, and $\bar{\mathcal{R}}$ is a closure convex set. So $\left(\bar{Z}(\tau), \bar{c}_1(\tau), \ldots, \bar{c}_m(\tau)\right) \in \bar{\mathcal{R}}$ satisfy (45), that is

$$-\bar{Z}(\tau) + \sum_{j \in J} \mu_j \bar{c}_j(\tau) \geqslant -\xi^* + \sum_{j \in J} \mu_j \tilde{C}. \tag{46}$$

Rearrange (46), and combine (18), then we have

$$\frac{1}{2\tau} \left\| \mathbb{E}\left\{\Theta(\tau)\right\} \right\|^2 \leqslant \beta + \Lambda + \frac{V}{\tau} \left\| \mu \right\| \cdot \left\| \mathbb{E}\{\Theta(\tau)\} \right\|, \tag{47}$$

where $\|*\|$ is the 2-norm of vector. Solve the quadratic inequation above with respect to $\left\| \mathbb{E}\left\{\Theta(\tau)\right\} \right\|$ and substitute (18), we have

$$\begin{aligned} \bar{c}_j(\tau) &\leqslant \tilde{C} + \frac{\mathbb{E}\{M_j(\tau)\}}{\tau} \leqslant \tilde{C} + \frac{\left\| \mathbb{E}\left\{\Theta(\tau)\right\} \right\|}{\tau} \\ &\leqslant \tilde{C} + \frac{V \left\| \mu \right\| + \sqrt{V^2 \left\| \mu \right\|^2 + 2(\beta + \Lambda)\tau}}{\tau}. \end{aligned} \tag{48}$$

Furthermore, take $\varepsilon = 1/V$, and when $\tau \geqslant 1/\varepsilon^2$, (48) can be further derived as

$$\bar{c}_j(\tau) \leqslant \tilde{C} + \left\| \mu \right\| \varepsilon + \sqrt{\left\| \mu \right\|^2 \varepsilon^2 + 2(\beta + \Lambda)\varepsilon^2} \leqslant \tilde{C} + O(\varepsilon). \tag{49}$$

On the other hand, we have,

$$\bar{Z}(\tau) \geqslant \xi^* - (\beta + \Lambda)\varepsilon \geqslant \xi^* - O(\varepsilon). \tag{50}$$

Therefore, when $\tau \geqslant 1/\varepsilon^2$, our algorithm can meet the convergence constraints of (34), and the convergence of LOSM is proved. □

Appendix C
Proof of Theorem 1

Proof. We have the following inequation,

$$\begin{aligned} &\hat{P}2 - P2^* \\ &= \frac{1}{2} \sum_{k \in K} \sum_{j \in J} \left(M_j(t) c_j^{k(t)} \hat{y}_j^k(t) \right) - \sum_{i \in I} \sum_{j \in J} V \hat{x}_{ij}(t) \\ &- \frac{1}{2} \sum_{k \in K} \sum_{j \in J} \left(M_j(t) c_j^{k(t)} y_j^{k^*}(t) \right) + \sum_{i \in I} \sum_{j \in J} V x_{ij}^*(t) \\ &\leqslant \frac{1}{2} \sum_{k \in K} \sum_{j \in J} \left(M_j(t) c_j^{k(t)} \hat{y}_j^k(t) \right) + \sum_{i \in I} \sum_{j \in J} V x_{ij}^*(t). \end{aligned} \tag{51}$$

We scale the gap between the solution obtained by RROM and the optimal solution to obtain the above inequation. Next, we will determine the bound of the inequation on the right side. For the first item, we use Chernoff Bound Theory [43] to scale $\sum_{k \in K} c_j^{k(t)} \hat{y}_j^k(t)$. First, according to the process of one-slot solving and constraint (9), we have

$$\begin{aligned} &\mathbb{E}\left\{ \sum_{k \in K} \hat{y}_j^k(t) c_j^{k(t)} \right\} = \sum_{k \in K} \mathbb{E}\left\{ \hat{y}_j^k(t) c_j^{k(t)} \right\} \\ &= \sum_{k \in K} \Pr\left[\hat{y}_j^k(t) = 1 \right] c_j^{k(t)} = \sum_{k \in K} \tilde{y}_j^k(t) c_j^{k(t)} \\ &= \sum_{k \in K} \tilde{y}_j^k(t) \varsigma_i^{k(t)} \frac{c_j^{k(t)}}{\varsigma_i^{k(t)}} \leqslant R_j \frac{c_{\max}}{\left(\varsigma^k\right)_{\min}}. \end{aligned} \tag{52}$$

Then, normalize $c_j^{k(t)}$ and $\sum_{k \in K} \tilde{y}_j^k(t) c_j^{k(t)}$ to let $\hat{y}_j^k(t) c_j^{k(t)} \in [0, 1]$, and since $\hat{y}_j^k(t) c_j^{k(t)}$ is independent of each other, $k \in \{1, 2, \cdots, \ell\}$. According to Chernoff Bound Theory [43], $\varepsilon > 0$, we have

$$\begin{aligned} &\Pr\left[\sum_{k \in K} \hat{y}_j^k(t) c_j^{k(t)} \geqslant (1 + \varepsilon) \sum_{k \in K} \tilde{y}_j^k(t) c_j^{k(t)} \right] \\ &\leqslant \exp\left(\frac{-\varepsilon^2 \sum_{k \in K} \tilde{y}_j^k(t) c_j^{k(t)}}{2 + \varepsilon} \right), \end{aligned} \tag{53}$$

Since $\sum_{k \in K} \tilde{y}_j^k(t) c_j^{k(t)} \leqslant R_j \frac{c_{\max}}{\left(\varsigma^k\right)_{\min}}$, it can be derived

$$\begin{aligned} &\Pr\left[\sum_{k \in K} \hat{y}_j^k(t) c_j^{k(t)} \geqslant (1 + \varepsilon) R_j \frac{c_{\max}}{\left(\varsigma^k\right)_{\min}} \right] \\ &\leqslant \Pr\left[\sum_{k \in K} \hat{y}_j^k(t) c_j^{k(t)} \geqslant (1 + \varepsilon) \sum_{k \in K} \tilde{y}_j^k(t) c_j^{k(t)} \right]. \end{aligned} \tag{54}$$

By the defination of μ, we have $\mu \leqslant \sum_{k \in K} \tilde{y}_j^k(t) c_j^{k(t)}$. And according to the monotonicity of negative exponential function, then we have

$$\exp\left(\frac{-\varepsilon^2 \sum_{k \in K} \tilde{y}_j^k(t) c_j^{k(t)}}{2 + \varepsilon} \right) \leqslant \exp\left(\frac{-\varepsilon^2 \mu}{2 + \varepsilon} \right). \tag{55}$$

Therefore, we can have the following inequation with (53), (54) and (55)

$$\Pr\left[\sum_{k \in K} \hat{y}_j^k(t) c_j^{k(t)} \geqslant (1 + \varepsilon) R_j \frac{c_{\max}}{\left(\varsigma^k\right)_{\min}} \right] \leqslant \exp\left(\frac{-\varepsilon^2 \mu}{2 + \varepsilon} \right). \tag{56}$$

Next, we relax the upper bound of the right term of the inequation (56) as $\frac{1}{r^3}$, that is, $\exp\left(\frac{-\varepsilon^2 \mu}{2 + \varepsilon} \right) \leqslant \frac{1}{r^3}$, and it ensures that the probability value of the left term of (56) is small enough [44]. Therefore, we can solve the value range of ε, as the following inequation

$$\varepsilon \geqslant \frac{3 \ln(r)}{2\mu} + \sqrt{\frac{9 \ln^2(r)}{4\mu^2} + \frac{6 \ln(r)}{\mu}}. \tag{57}$$

We pick a value of ε that satisfies the above inequation (57)

$$\varepsilon = \frac{3 \ln(r)}{\mu} + \sqrt{\frac{6 \ln(r)}{\mu}}. \tag{58}$$

Therefore, when the number of services is large, the following formula holds

$$\Pr\left[\sum_{k\in K}\hat{y}_j^k(t)c_j^{k(t)} \geqslant (1+\varepsilon)R_j\frac{c_{\max}}{\left(\varsigma^k\right)_{\min}}\right] \leqslant \frac{1}{r^3}\to 0. \tag{59}$$

Since (59) indicates that the probability of $\sum\limits_{k\in K}\hat{y}_j^k(t)c_j^{k(t)} \geqslant (1+\varepsilon)R_j\frac{c_{\max}}{(\varsigma^k)_{\min}}$ tends to 0, hence the following inequation holds

$$\sum_{k\in K}\hat{y}_j^k(t)c_j^{k(t)} < R_j\frac{c_{\max}}{\left(\varsigma^k\right)_{\min}}\left(1+\frac{3\ln(r)}{\mu}+\sqrt{\frac{6\ln(r)}{\mu}}\right). \tag{60}$$

For the second term of the inequation (51) on the right side, from the computation capacity constraint (8) we have the following inequation

$$\begin{aligned}\sum_{i\in I}\sum_{j\in J}Vx_{ij}^*(t) &= V\sum_{i\in I}\sum_{j\in J}x_{ij}^*(t)w_i^{k(t)}\frac{1}{w_i^{k(t)}}\\ &\leqslant V\sum_{i\in I}\sum_{j\in J}x_{ij}^*(t)w_i^{k(t)}\frac{1}{\left(w^k\right)_{\min}}\\ &\leqslant mV\left(W_j\right)_{\max}\frac{1}{\left(w^k\right)_{\min}}.\end{aligned} \tag{61}$$

Therefore, taking (60) and (61) into (51), Theorem 1 is proved. □

Appendix D
Proof of Theorem 2

Proof. First, we prove Theorem 2(a). For the problem of service deployment and request routing, there exists a solution $x_{ij}^+(t)$, $y_j^{k+}(t)$, $\varepsilon>0$, that is independent of the current queue backlogs $\mathbf{\Theta}(t)$ and i.i.d over all time slots. For example, it is obtained by a stationary solution search independent of $\mathbf{\Theta}(t)$, and the solution satisfies

$$\begin{aligned}\mathbb{E}\left\{Z^+\left(x_{ij}^+(t)\right)\right\} &\geqslant \xi^*-\varepsilon,\\ \mathbb{E}\left\{c_j^+(t)\right\} &\leqslant \tilde{C}+\varepsilon.\end{aligned} \tag{62}$$

For (27), according to the relationship between the solution and the optimal solution, and the independence between the solution and $\mathbf{\Theta}(t)$, taking $\varepsilon\to 0$, then taking expectations and using the law of iterated expectations, the optimization goal of (27) can be derived as

$$\begin{aligned}&\mathbb{E}\left\{L(\mathbf{\Theta}(t+1))\right\}-\mathbb{E}\left\{L(\mathbf{\Theta}(t))\right\}-V\mathbb{E}\left\{Z\left(x_{ij}(t)\right)\right\}\\ &\quad\leqslant \beta+\Lambda-V\mathbb{E}\left\{Z^+\left(x_{ij}^+(t)\right)\right\}+\sum_{j\in J}M_j(t)\mathbb{E}\left\{[c_j^+(t)-\tilde{C}]\right\}\\ &\quad\leqslant \beta+\Lambda-V(\xi^*-\varepsilon)+\varepsilon\sum_{j\in J}M_j(t)\\ &\quad\leqslant \beta+\Lambda-V\xi^*.\end{aligned} \tag{63}$$

For the above inequation, summing the telescoping series over $t\in\{0,1,\cdots,\tau-1\}$, dividing by $V\tau$, and rearranging terms, we have

$$\begin{aligned}&\frac{1}{\tau}\sum_{t\in T}\mathbb{E}\left\{Z\left(x_{ij}(t)\right)\right\}\\ &\geqslant \xi^*-\frac{\beta+\Lambda}{V}+\frac{\mathbb{E}\left\{L(\mathbf{\Theta}(\tau))\right\}-\mathbb{E}\left\{L(\mathbf{\Theta}(0))\right\}}{V\tau},\end{aligned} \tag{64}$$

where $L(\mathbf{\Theta}(0))<\infty$ is a finite number, and $\mathbb{E}\{L(\mathbf{\Theta}(t))\}\geqslant 0$. Taking a limit inferior as $\tau\to\infty$, hence we get (36) to prove the performance guarantees in Theorem 2(a).

Next, we prove that all queues are mean rate stable in Theorem 2(b). Given in Lemma 1, the mean rate stable of queues ensures that the migration cost constraint (11) is satisfied.

For (63), summing the telescoping series over $t\in\{0,1,\cdots,\tau-1\}$. Denote $\xi_{\min}$ and $\xi_{\max}$ as the minimum and maximum values of $\mathbb{E}\left\{Z\left(x_{ij}(t)\right)\right\}$ respectively. $\xi_{\max}\geqslant\mathbb{E}\left\{Z\left(x_{ij}(t)\right)\right\}$, and rearranging terms yields

$$\frac{1}{2}\sum_{j\in J}\mathbb{E}\left\{[M_j(\tau)]^2\right\}\leqslant\mathbb{E}\left\{L(\mathbf{\Theta}(0))\right\}+(\beta+\Lambda+V(\xi_{\max}-\xi^*))\tau. \tag{65}$$

According to (65), for all $j\in J$, we have

$$\mathbb{E}\left\{[M_j(\tau)]^2\right\}\leqslant 2\mathbb{E}\left\{L(\mathbf{\Theta}(0))\right\}+2(\beta+\Lambda+V(\xi_{\max}-\xi^*))\tau. \tag{66}$$

Since $\mathbb{E}\{[M_j(t)]^2\}\geqslant\mathbb{E}\{|M_j(t)|\}^2$ and $M_j(t)\geqslant 0$, taking the square root of (66) and dividing by τ, and taking a limit as $\tau\to\infty$, we have

$$\begin{aligned}&0\leqslant\lim_{\tau\to\infty}\frac{1}{\tau}\mathbb{E}\left\{M_j(\tau)\right\}\leqslant\\ &\lim_{\tau\to\infty}\sqrt{\frac{2\mathbb{E}\left\{L(\mathbf{\Theta}(0))\right\}}{\tau^2}+\frac{2(\beta+\Lambda+V(\xi_{\max}-\xi^*))}{\tau}}=0.\end{aligned} \tag{67}$$

Hence we have proved the mean rate stability in Theorem 2(b).

Next, we prove the strong stability of migration cost. There exists a solution $x_{ij}^+(t)$, $y_j^{k+}(t)$, $\delta>0$, $\psi(\delta)$, and $\xi_{\min}\leqslant\psi(\delta)\leqslant\xi_{\max}$. This solution is independent of the current queue backlogs $\mathbf{\Theta}(t)$ and i.i.d over all time slots, which satisfies

$$\begin{aligned}\mathbb{E}\left\{Z^+\left(x_{ij}^+(t)\right)\right\} &= \psi(\delta),\\ \mathbb{E}\left\{c_j^+(t)\right\} &\leqslant \tilde{C}-\delta.\end{aligned} \tag{68}$$

For (27), since the solution is independent of $\mathbf{\Theta}(t)$ and time slot, taking iterated expectations and it can be derived as

$$\begin{aligned}&\mathbb{E}\left\{L(\mathbf{\Theta}(t+1))\right\}-\mathbb{E}\left\{L(\mathbf{\Theta}(t))\right\}-V\mathbb{E}\left\{Z\left(x_{ij}(t)\right)\right\}\\ &\quad\leqslant\beta+\Lambda-V\psi(\delta)-\delta\sum_{j\in J}\mathbb{E}\left\{M_j(t)\right\}.\end{aligned} \tag{69}$$

Summing all migration costs over $t\in\{0,1,\ldots,\tau-1\}$ by telescoping sums, dividing both sides by τ and rearranging terms, we have

$$\begin{aligned}&\frac{\delta}{\tau}\sum_{t\in T}\sum_{j\in J}\mathbb{E}\left\{M_j(t)\right\}\leqslant\beta+\Lambda+V\frac{1}{\tau}\sum_{t\in T}\mathbb{E}\left\{Z\left(x_{ij}(t)\right)\right\}\\ &\quad-V\psi(\delta)-\frac{\mathbb{E}\left\{L(\mathbf{\Theta}(\tau))\right\}-\mathbb{E}\left\{L(\mathbf{\Theta}(0))\right\}}{\tau}.\end{aligned} \tag{70}$$

Since $\frac{1}{\tau}\sum\limits_{t\in T}\mathbb{E}\left\{Z\left(x_{ij}(t)\right)\right\}\leqslant\xi^*$, dividing the above by δ, then taking a limit superior as $\tau \rightarrow \infty$, we have

$$\limsup_{\tau\rightarrow\infty} \frac{1}{\tau}\sum_{t\in T}\sum_{j\in J}\mathbb{E}\left\{M_j(t)\right\} \leqslant \frac{\beta+\Lambda+V[\xi^*-\psi(\delta)]}{\delta}. \tag{71}$$

Hence, the strong stability of migration cost in Theorem 2(b) is proved.

□

References

[1] G. M. D. T. Forecast, "Cisco visual networking index: Global mobile data traffic forecast update, 2017–2022," *Cisco white paper*, vol. 2017, pp. 1–36, Feb. 2019.

[2] S. Liu, L. Liu, J. Tang, B. Yu, Y. Wang, and W. Shi, "Edge computing for autonomous driving: Opportunities and challenges," *Proc. IEEE*, vol. 107, no. 8, pp. 1697–1716, Jun. 2019.

[3] J. Ahn, J. Lee, S. Yoon, and J. K. Choi, "A novel resolution and power control scheme for energy-efficient mobile augmented reality applications in mobile edge computing," *IEEE Wirel. Commun. Lett.*, vol. 9, no. 6, pp. 750–754, Jun. 2020.

[4] J. Ren, D. Zhang, S. He, Y. Zhang, and T. Li, "A survey on end-edge-cloud orchestrated network computing paradigms: Transparent computing, mobile edge computing, fog computing, and cloudlet," *ACM Comput. Surv.*, vol. 52, no. 6, pp. 1–36, Oct. 2020.

[5] N. Abbas, Y. Zhang, A. Taherkordi, and T. Skeie, "Mobile edge computing: A survey," *IEEE Internet Things J.*, vol. 5, no. 1, pp. 450–465, Feb. 2018.

[6] G. Peng, H. Wu, H. Wu, and K. Wolter, "Constrained multi-objective optimization for IoT-enabled computation offloading in collaborative edge and cloud computing," *IEEE Internet Things J.*, vol. 8, no. 17, pp. 13 723–13 736, Sept. 2021.

[7] T. Taleb, K. Samdanis, B. Mada, H. Flinck, S. Dutta, and D. Sabella, "On multi-access edge computing: A survey of the emerging 5G network edge cloud architecture and orchestration," *IEEE Commun. Surv. Tutor.*, vol. 19, no. 3, pp. 1657–1681, May 2017.

[8] J. Xia, C. Li, X. Lai, S. Lai, F. Zhu, D. Deng, and L. Fan, "Cache-aided mobile edge computing for B5G wireless communication networks," *EURASIP J. Wirel. Commun. Netw.*, vol. 2020, no. 1, pp. 1–15, Jan. 2020.

[9] A. Sill, "The design and architecture of microservices," *IEEE Cloud Comput.*, vol. 3, no. 5, pp. 76–80, Nov. 2016.

[10] S. Taherizadeh, V. Stankovski, and M. Grobelnik, "A capillary computing architecture for dynamic internet of things: Orchestration of microservices from edge devices to fog and cloud providers," *Sensors*, vol. 18, no. 9, pp. 1–23, Sept. 2018.

[11] M. Alam, J. Rufino, J. Ferreira, S. H. Ahmed, N. Shah, and Y. Chen, "Orchestration of microservices for IoT using docker and edge computing," *IEEE Commun. Mag.*, vol. 56, no. 9, pp. 118–123, Sept. 2018.

[12] S. Wang, Y. Guo, N. Zhang, P. Yang, A. Zhou, and X. Shen, "Delay-aware microservice coordination in mobile edge computing: A reinforcement learning approach," *IEEE Trans. Mob. Comput.*, vol. 20, no. 3, pp. 939–951, Mar. 2021.

[13] Z. Rejiba, X. Masip-Bruin, and E. Marín-Tordera, "A survey on mobility-induced service migration in the fog, edge, and related computing paradigms," *ACM Comput. Surv.*, vol. 52, no. 5, pp. 90:1–90:33, Sept. 2019.

[14] T. Ouyang, Z. Zhou, and X. Chen, "Follow me at the edge: Mobility-aware dynamic service placement for mobile edge computing," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 10, pp. 2333–2345, Oct. 2018.

[15] H. Wei, H. Luo, and Y. Sun, "Mobility-aware service caching in mobile edge computing for internet of things," *Sensors*, vol. 20, no. 3, pp. 1–20, Jan. 2020.

[16] F. Lyu, J. Ren, P. Yang, N. Cheng, Y. Zhang, and X. S. Shen, "SoSA: Socializing static APs for edge resource pooling in large-scale WiFi system," in *Proc. 39th IEEE Conf. Comput. Commun. (INFOCOM)*, Aug. 2020, pp. 1181–1190.

[17] X. Ge, S. Tu, G. Mao, C. Wang, and T. Han, "5G ultra-dense cellular networks," *IEEE Wirel. Commun.*, vol. 23, no. 1, pp. 72–79, Feb. 2016.

[18] D. Calabuig, S. Barmpounakis, S. Gimenez, A. Kousaridas, T. R. Lakshmana, J. Lorca, P. Lundén, Z. Ren *et al.*, "Resource and mobility management in the network layer of 5G cellular ultra-dense networks," *IEEE Commun. Mag.*, vol. 55, no. 6, pp. 162–169, Jun. 2017.

[19] Y. Mao, J. Zhang, and K. B. Letaief, "Dynamic computation offloading for mobile-edge computing with energy harvesting devices," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 12, pp. 3590–3605, Dec. 2016.

[20] S. Josilo and G. Dán, "Computation offloading scheduling for periodic tasks in mobile edge computing," *IEEE/ACM Trans. Netw.*, vol. 28, no. 2, pp. 667–680, Apr. 2020.

[21] X. Chen, Y. Cai, Q. Shi, M. Zhao, B. Champagne, and L. Hanzo, "Efficient resource allocation for relay-assisted computation offloading in mobile-edge computing," *IEEE Internet Things J.*, vol. 7, no. 3, pp. 2452–2468, Mar. 2020.

[22] J. Feng, Q. Pei, F. R. Yu, X. Chu, J. Du, and L. Zhu, "Dynamic network slicing and resource allocation in mobile edge computing systems," *IEEE Trans. Veh. Technol.*, vol. 69, no. 7, pp. 7863–7878, Jul. 2020.

[23] G. Castellano, F. Esposito, and F. Risso, "A distributed orchestration algorithm for edge computing resources with guarantees," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, Jun. 2019, pp. 2548–2556.

[24] T. Taleb, A. Ksentini, and P. A. Frangoudis, "Follow-me cloud: When cloud services follow mobile users," *IEEE Trans. Cloud Comput.*, vol. 7, no. 2, pp. 369–382, Feb. 2019.

[25] Q. Yuan, J. Li, H. Zhou, T. Lin, G. Luo, and X. Shen, "A joint service migration and mobility optimization approach for vehicular edge computing," *IEEE Trans. Veh. Technol.*, vol. 69, no. 8, pp. 9041–9052, Aug. 2020.

[26] Y. Ding, C. Liu, K. Li, Z. Tang, and K. Li, "Task offloading and service migration strategies for user equipments with mobility consideration in mobile edge computing," in *Proc. IEEE Int. Conf. Parallel Distrib. Process. Appl., Big Data Cloud Comput., Sustain. Comput. Commun., Social Comput. Netw. (ISPA/BDCloud/SocialCom/SustainCom)*, Mar. 2019, pp. 176–183.

[27] A. Nadembega, A. S. Hafid, and R. Brisebois, "Mobility prediction model-based service migration procedure for follow me cloud to support QoS and QoE," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jul. 2016, pp. 1–6.

[28] S. Wang, R. Urgaonkar, T. He, K. Chan, M. Zafer, and K. K. Leung, "Dynamic service placement for mobile micro-clouds with predicted future costs," *IEEE Trans. Parallel Distrib. Syst.*, vol. 28, no. 4, pp. 1002–1016, Apr. 2017.

[29] S. Wang, R. Urgaonkar, M. Zafer, T. He, K. Chan, and K. K. Leung, "Dynamic service migration in mobile edge computing based on markov decision process," *IEEE/ACM Trans. Netw.*, vol. 27, no. 3, pp. 1272–1288, Jun. 2019.

[30] A. Aissioui, A. Ksentini, A. M. Guéroui, and T. Taleb, "On enabling 5G automotive systems using follow me edge-cloud concept," *IEEE Trans. Veh. Technol.*, vol. 67, no. 6, pp. 5302–5316, Jun. 2018.

[31] Y. Ma, W. Liang, and S. Guo, "Mobility-aware delay-sensitive service provisioning for mobile edge computing," in *Proc. IEEE Conf. Comput. Commun. Workshops (INFOCOM Workshops)*, Sept. 2019, pp. 270–276.

[32] S. Zhang, P. He, K. Suto, P. Yang, L. Zhao, and X. Shen, "Cooperative edge caching in user-centric clustered mobile networks," *IEEE Trans. Mobile Comput.*, vol. 17, no. 8, pp. 1791–1805, Aug. 2018.

[33] S. Schneider, R. Khalili, A. Manzoor, H. Qarawlus, R. Schellenberg, H. Karl, and A. Hecker, "Self-learning multi-objective service coordination using deep reinforcement learning," *IEEE Trans. Netw. Serv. Manag.*, vol. 18, no. 3, pp. 3829–3842, Mar. 2021.

[34] T. He, H. Khamfroush, S. Wang, T. L. Porta, and S. Stein, "It's hard to share: Joint service placement and request scheduling in edge clouds with sharable and non-sharable resources," in *Proc. 38th IEEE Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Jul. 2018, pp. 365–375.

[35] Y. Yu, J. Yang, C. Guo, H. Zheng, and J. He, "Joint optimization of service request routing and instance placement in the microservice system," *J. Netw. Comput. Appl.*, vol. 147, Dec. 2019.

[36] K. Poularakis, J. Llorca, A. M. Tulino, I. J. Taylor, and L. Tassiulas, "Service placement and request routing in MEC networks with storage, computation, and communication constraints," *IEEE/ACM Trans. Netw.*, vol. 28, no. 3, pp. 1047–1060, Jun. 2020.

[37] X. Yang, Z. Fei, J. Zheng, N. Zhang, and A. Anpalagan, "Joint multi-user computation offloading and data caching for hybrid mobile cloud/edge computing," *IEEE Trans. Veh. Technol.*, vol. 68, no. 11, pp. 11 018–11 030, Nov. 2019.

[38] S. Josilo and G. Dán, "Selfish decentralized computation offloading for mobile cloud computing in dense wireless networks," *IEEE Trans. Mob. Comput.*, vol. 18, no. 1, pp. 207–220, Jan. 2019.

[39] N. Nouri, J. Abouei, M. Jaseemuddin, and A. Anpalagan, "Joint access and resource allocation in ultradense mmwave NOMA networks with mobile edge computing," *IEEE Internet Things J.*, vol. 7, no. 2, pp. 1531–1547, Feb. 2020.

[40] M. Sheng, Y. Dai, J. Liu, N. Cheng, X. Shen, and Q. Yang, "Delay-aware computation offloading in NOMA MEC under differentiated uploading delay," *IEEE Trans. Wirel. Commun.*, vol. 19, no. 4, pp. 2813–2826, Apr. 2020.

[41] K. Wang, Y. Liu, Z. Ding, A. Nallanathan, and M. Peng, “User association and power allocation for multi-cell non-orthogonal multiple access networks,” *IEEE Trans. Wirel. Commun.*, vol. 18, no. 11, pp. 5284–5298, Nov. 2019.

[42] M. J. Neely, *Stochastic Network Optimization with Application to Communication and Queueing Systems*, ser. Synthesis Lectures on Communication Networks. Morgan & Claypool, Jun. 2010.

[43] M. Mitzenmacher and E. Upfal, *Probability and Computing: Randomized Algorithms and Probabilistic Analysis*. Cambridge Univ. Press, Jun. 2005.

[44] A. Srinivasan, “Approximation algorithms via randomized rounding: A survey,” *Ser. Adv. Topics Math., Polish Scientific Publishers PWN*, pp. 9–71, 1999.

[45] S. Jiang, Z. Song, O. Weinstein, and H. Zhang, “A faster algorithm for solving general LPs,” in *Proc. 53rd Annu. ACM Symp. Theory Comput. (STOC)*, Jun. 2021, pp. 823–832.

[46] M. J. Neely, “A simple convergence time analysis of drift-plus-penalty for stochastic optimization and convex programs,” *arXiv preprint arXiv:1412.0791*, 2014.

[47] D. Bertsekas, A. Nedic, and A. Ozdaglar, *Convex analysis and optimization*. Athena Scientific, Apr. 2003.

[48] M. Li, F. R. Yu, P. Si, and Y. Zhang, “Green machine-to-machine communications with mobile edge computing and wireless network virtualization,” *IEEE Commun. Mag.*, vol. 56, no. 5, pp. 148–154, May 2018.

[49] A. Keränen, J. Ott, and T. Kärkkäinen, “The ONE simulator for DTN protocol evaluation,” in *Proc. 2nd Int. Conf. Simulat. Tools Techn. (ICST)*, Mar. 2009, pp. 50–55.

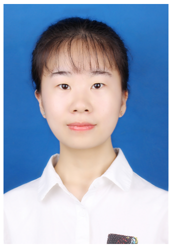
[50] Q. Peng, Y. Xia, F. Zeng, J. Lee, C. Wu, X. Luo, W. Zheng, H. Liu *et al.*, “Mobility-aware and migration-enabled online edge user allocation in mobile edge computing,” in *Proc. IEEE Int. Conf. Web Services. (ICWS)*, Jul. 2019, pp. 91–98.

[51] V. Lazaridis, K. Paparrizos, N. Samaras, and A. Sifaleras, “Visual linProg: A web-based educational software for linear programming,” *Comput. Appl. Eng. Educ.*, vol. 15, no. 1, pp. 1–14, Apr. 2007.

[52] K. Shanmugam, N. Golrezaei, A. G. Dimakis, A. F. Molisch, and G. Caire, “Femtocaching: Wireless content delivery through distributed caching helpers,” *IEEE Trans. Inf. Theory*, vol. 59, no. 12, pp. 8402–8413, Dec. 2013.

Xiangyi Chen received the M.S. degree in computer science from Northeastern University, Shenyang, China, in 2019. She is currently pursuing the Ph.D. degree in computer science at Northeastern University, Shenyang, China. Her research interests include mobile edge computing, network function virtualization, and software-defined networking, etc.

Yuanguo Bi (M’11) received the Ph.D. degree in computer science from Northeastern University, Shenyang, China, in 2010. He was a Visiting Ph.D. Student with the BroadBand Communications Research (BBCR) lab, Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada from 2007 to 2009. He is currently a Professor with the School of Computer Science and Engineering, Northeastern University. He has authored/coauthored more than 50 journal/conference papers, including high quality journal papers, such as IEEE JSAC, IEEE TWC, IEEE TITS, IEEE TVT, IEEE IoT Journal, IEEE Communications Magazine, IEEE Wireless Communications, IEEE Network, and mainstream conferences, such as IEEE Global Communications Conference, IEEE International Conference on Communications. His research interests include medium access control, QoS routing, multihop broadcast, and mobility management in vehicular networks, software-defined networking, and mobile edge computing. Dr. Bi has served as an Editor/Guest Editor for IEEE Communications Magazine, IEEE Wireless Communications, IEEE ACCESS. He has also served as the Technical Program Committee member for many IEEE conferences.



Xueping Chen received the B.S. degree from Northeast Petroleum University, Daqing, China, in 2019. She is currently pursuing the master’s degree in computer science at Northeastern University, Shenyang, China. Her research interests include software-defined networking and mobile edge computing.

Hai Zhao received the B.S. degree in electrical engineering from Dalian Maritime University, Dalian, China, in 1982, and the M.S. and Ph.D. degrees in computer science from Northeastern University, China, in 1987 and 1995, respectively. He is currently a professor with the school of Computer Science and Engineering, Northeastern University, Shenyang, China. He is also the Director of the Liaoning Provincial Key Laboratory of Embedded Technology. His current research interests include embedded Internet technology, wireless sensor networks, vehicular ad hoc networks, body area networks, pervasive computing, operating systems, data and information fusion, computer simulation, and virtual reality.

Nan Cheng (M’16) received the Ph.D. degree from the Department of Electrical and Computer Engineering, University of Waterloo in 2016, and B.E. degree and the M.S. degree from the Department of Electronics and Information Engineering, Tongji University, Shanghai, China, in 2009 and 2012, respectively. He worked as a Post-doctoral fellow with the Department of Electrical and Computer Engineering, University of Toronto, from 2017 to 2019. He is currently a professor with State Key Lab of ISN and with School of Telecommunication Engineering, Xidian University, Shaanxi, China. His current research focuses on B5G/6G, space-air-ground integrated network, big data in vehicular networks, and self-driving system. His research interests also include performance analysis, MAC, opportunistic communication, and application of AI for vehicular networks.

Fuliang Li received the B.S. degree in computer science from the Northeastern University, Shenyang, China in 2009, and the Ph.D. degree in computer science from the Tsinghua University, Beijing, China in 2015. He is currently an associate professor at the School of Computer Science and Engineering, Northeastern University, Shenyang, China. He was a postdoctoral fellow with Department of Computing at Hong Kong Polytechnic University, Hong Kong during 2016-2017. He published 50 Journal/conference papers, including journal papers such as IEEE/ACM TON, IEEE TCC, Computer Networks, Computer Communications, Journal of Network and Computer Applications, and mainstream conferences such as IEEE INFOCOM, IEEE ICDCS, IEEE/ACM IWQoS, IEEE GLOBECOM, IEEE LCN, IEEE CLOUD, and IFIP/IEEE IM, etc. His research interests include network management and measurement, mobile computing, software defined networking and network security. He is a member of the IEEE.

Wenlin Cheng received the B.S. degree in mathematics and applied mathematics in 2017 and the M.S. degree in computer science from Northeastern University, Shenyang, China, in 2019. He is currently pursuing the Ph.D. degree in computer science at Northeastern University, Shenyang, China. His research interests include software-defined networking, network function virtualization, and 5G/B5G wireless communication, etc.