Dynamic Multi-time Scale User Admission and Resource Allocation for Semantic Extraction in MEC Systems

Dynamic Multi-time Scale User Admission and Resource Allocation for Semantic Extraction in MEC Systems

Yuanpeng Zheng, *Student Member, IEEE,* Tiankui Zhang, *Senior Member, IEEE,* Jonathan Loo

***Abstract*—This paper investigates the semantic extraction task-oriented dynamic multi-time scale user admission and resource allocation in mobile edge computing (MEC) systems. Amid prevalence artificial intelligence applications in various industries, the offloading of semantic extraction tasks which are mainly composed of convolutional neural networks of computer vision is a great challenge for communication bandwidth and computing capacity allocation in MEC systems. Considering the stochastic nature of the semantic extraction tasks, we formulate a stochastic optimization problem by modeling it as the dynamic arrival of tasks in the temporal domain. We jointly optimize the system revenue and cost which are represented as user admission in the long term and resource allocation in the short term respectively. To handle the proposed stochastic optimization problem, we decompose it into short-time-scale subproblems and a long-time-scale subproblem by using the Lyapunov optimization technique. After that, the short-time-scale optimization variables of resource allocation, including user association, bandwidth allocation, and computing capacity allocation are obtained in closed form. The user admission optimization on long-time scales is solved by a heuristic iteration method. Then, the multi-time scale user admission and resource allocation algorithm is proposed for dynamic semantic extraction task computing in MEC systems. Simulation results demonstrate that, compared with the benchmarks, the proposed algorithm improves the performance of user admission and resource allocation efficiently and achieves a flexible trade-off between system revenue and cost at multi-time scales and considering semantic extraction tasks.**

***Index Terms*—Semantic extraction task, resource allocation, MEC, dynamic optimization.**

I. Introduction

In recent years, mobile edge computing (MEC), which supports not only computing but also communications and storage, has become a key technology to solve many related problems with specific requirements [1], [2]. By being closer to the edge of network than traditional cloud computing systems, MEC can obviously improve the quality of user experience, including optimization of delay and energy consumption [3]. Devices can significantly reduce their response times and energy consumption by offloading the computing tasks to nearby edge network, hence resource capacity and scheduling of MEC systems become a very important issue, especially in the context of the increasing number of intelligent tasks. The rapid development of network edge applications such as the Internet of Things (IoT) indicates change of service requirements and diversity of tasks, nevertheless, few existing works consider the various performance requirements of these dynamics applications [4] and the characteristics of computing tasks [5]. Therefore, how to efficiently allocate resource to support the dynamics demand of services is still an unaddressed problem.

Hence, in the context of massive IoT devices deployment, limited terminal computing capacity and battery capacity, and increasingly complex computing tasks, existing works on resource allocation in MEC systems has become specific and multidimensional [6]–[11]. S. Zarandi *et al.* [6] investigated a way of combining MEC and network slicing, and proposed the optimization of the weighted sum of the difference between the observed delay and the delay requirement. By consider edge users and large data volume, a power consumption and delay optimization problem in unmanned aerial vehicle (UAV) assisted MEC systems was addressed by G. Faraci *et al.* [7]. As a key scenario, an efficient method of MEC and network slice integration which is deployed on the IoT platform was proposed by J. Y. Hwang *et al.* [8] to maximize the effect of decreasing delay and traffic prioritization. X. Cao *et al.* [9] introduced a new MEC setup where an UAV is served by cellular ground base stations for computation offloading to minimize the UAV's task completion time considering computing capacity. Considering computing tasks, the MEC techniques combined with network slicing and non-orthogonal multiple access was leveraged by M. A. Hossain *et al.* [10] to minimize the total latency of the computing tasks with energy constraints. T. Zhang *et al.* [11] considered that a UAV equipped with an MEC server is deployed to serve a number of terminal devices of Internet of Things in a finite period, which aims to minimize the total energy consumption including communication-related energy, computation-related energy and UAV's flight energy by optimizing the bits allocation. Obviously, the above works do not consider the influence of the stochastic nature and specific computing capacity consumption of computing tasks on user admission and resource allocation. The research on specific tasks have become important in MEC systems considering communication bandwidth and computing capacity allocation. Ignoring the characteristics will result in

This work was supported by National Natural Science Foundation of China under Grants 61971060. (Corresponding author: Tiankui Zhang)

Yuanpeng Zheng, Tiankui Zhang are with the School of Information and Communication Engineering, Beijing University of Posts and Telecommunications, Beijing 100876, China (e-mail: {zhangtiankui, zhengyuanpeng}@bupt.edu.cn).

Jonathan Loo is with the School of Computing and Engineering, University of West London, London W5 5RF, U.K. (e-mail: jonathan.loo@uwl.ac.uk).

some errors in real scenario and hence can not satisfy delay requirements well.

The rise of semantic communication research in recent years has brought more requirements to the MEC field. Meanwhile, it proposes more scenarios of specific tasks in MEC systems and a few studies on quantification of computational complexity of intelligent tasks has been discussed [12]–[17]. Some works proposed practical schemes to deploy semantic communication to MEC systems. H. Xie *et al.* [12] proposed a brand new framework of semantic communication where a deep learning based semantic communication system for text transmission combined with deep learning, natural language processing and semantic layer communication was constructed. After that, H. Xie *et al.* [13] considered a semantic communication system which is constructed between edge and IoT devices where MEC servers trains and updates the semantic communication model based on deep learning, and the IoT devices collect and transmit data based on the training model. H. Qi *et al.* [14] investigated a model named PALEO which was applied to analyse performance of deep neural network (DNN). Nevertheless, D. Justus *et al.* [15] indicated that the presentation of computational complexity of PALEO was not accurate because of many other influence factors, and proposed an alternative strategy which predicted execution time by training a deep learning network including network features and hardware features. An approximation strategy of optimization of DNN training was proposed by D. Bienstock *et al.* [16], which modelled DNN as a directed graph to control approximation error of computational complexity. M. Bianchini *et al.* [17] proposed a new approach to study how the depth of feedforward neural networks impacts on their ability in implementing high complexity functions and indicated how the complexity depends on the number of hidden units and the used activation function. It is shown that the resource allocation problems for the semantic communication including the communication bandwidth and the computing capacity with the dynamic arrival of task in MEC systems has not been fully researched and it is hard but important to present characteristics and complexity of intelligent computing tasks.

There are still some studies consider the dynamic of mobile network and the stochastic nature of computing tasks [4], [18]–[20]. F. Guo *et al.* [18] designed the framework where the service requirements of some IoT applications have been changing. In [19], Y. Xiao *et al.* considered the dynamic of fog computing networks to maximize the utilization efficiency of available resources while balancing the workloads among fog nodes. The real-time dynamics of the network resource requests have been discussed in [20] by N. Van Huynh*et al.* and obtain the optimal resource allocation policy under the dynamics of the frequency of request. J. Feng et al. [4] considered the stochastic nature of tasks and proposed an architecture that maximizing revenue of network provider in MEC systems, where a multi-time scale scheme was adopted to increase revenue on the basis of QoS guarantee. However, it is necessary to integrate the stochastic nature of tasks and quantification of complexity of computing tasks in MEC systems. Semantic extraction tasks which are mainly composed of convolutional neural networks (CNN) of computer vision gradually become the mainstream on dynamic resource allocation. In conclusion, modelling dynamic and computing capacity of semantic extraction tasks in MEC systems has not been considered yet according to the above works.

A. Motivation and Contribution

As mentioned above, the combination of dynamic multi-time admission and resource allocation in MEC systems with specific computing tasks, i.e., semantic extraction tasks, is still an unaddressed research area, which motivate this contribution. In this paper, we formulate a stochastic optimization problem by modelling it as dynamic arrival of tasks in temporal domain considering the stochastic nature of the semantic extraction tasks. In order to investigate the dynamic arrival and stochastic nature of tasks, we adopt multi-time scale to represent traffic variations. Based on the dynamic model, we optimize the average utility over time that consists of the system revenue and cost which depends on user admission in the long term and resource allocation in the short term. We also model computing characteristic of semantic extraction task as a formula based on the structure of CNN. The primary contributions of this paper are as follows:

- We formulate a stochastic optimization problem for dynamic user admission and resource allocation considering the stochastic nature of semantic extraction tasks in MEC systems. We set up a queue model to represent dynamic of semantic extraction tasks and define the operator's utility which consists of long-time-scale revenue depending on the number of users and short-time-scale cost depending on power consumption in order to achieve continuous revenue in temporal domain with as little cost as possible at each time slot. For this study, we adopt a formula based on the structure of CNN to quantify the relationship between input data and computational complexity of semantic extraction tasks.
- We solve the highly coupled problem without any prior knowledge of traffic distributions or channel information with the assistance of the Lyapunov optimization with maximisation of the number of users and minimisation of power consumption. We decoupled manifold optimization variables on the dimension of time scale and propose a multi-time scale user admission and resource allocation algorithm for semantic extraction tasks where the dynamic user admission subproblem is in the long term and user association subproblem, bandwidth allocation subproblem and computing capacity allocation subproblem are in the short term. The dynamic user admission subproblem in long term is settled by a heuristic iteration method and resouce allocation subproblems in short term are solved in closed forms.
- We demonstrate the simulation results which verify that our framework is applicable to semantic scenario in MEC systems and the proposed algorithm has significant effect for the multi-time scale problem solving. It is shown that, compared with the benchmarks, the proposed algorithm improves the performance of user admission and resource allocation efficiently and achieves a flexible trade-off

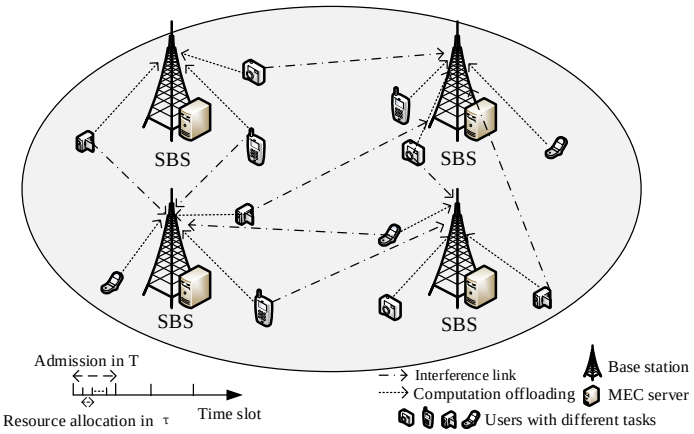

Fig. 1. The system scenario.

between system revenue and cost at multi-time scales and considering semantic extraction tasks.

B. Organization

The rest of this paper is organized as follows. In Section II, we introduce system model and problem formulation. In Section III, we decompose the coupling problem into resource allocation in the short-time slot and user admission in the long-time slot. The performance of the proposed algorithm is evaluated by the simulation in Section IV, which is followed by our conclusions in Section V.

II. Stochastic Optimisation Problem Formulation

We consider that fog radio access network (F-RAN) is built on MEC systems, and communication and computing between terminals and MEC are for specific semantic extraction tasks, as shown in Fig. 1. We equip MEC servers on small base stations (SBS) to form MEC systems, which is assembled as $K^S = \{1,...,k,...,K\}$. In order to consider dynamic allocation of resources in temporal domain to dynamically meet the demands of multiple task slices, we design two types of time slots based on the time-slotted system where one is a long time slot (LTS) and the other is a short time slot (STS). In this paper, our system contains multiple LTSs which are dedicated to user admission and the length of the LTS is *T*. We assume that each LTS contains *p* STSs which are dedicated to resource allocation and the length of STS is τ, i.e., $T = p\tau$. At LTS *l*, we denote the set of users by $U^S = \{1,...,u,...,U\}$, and the set of specific tasks by $M^S = \{1,...,m,...,M\}$. At the beginning of each LTS the network operator can decide user admission and at the beginning of each STS resource allocation policies is given. Let the admission control variable of the user *u* accessing MEC systems be $y_u(l) \in \{0,1\}$, where $y_u(l) = 1$ denotes user *u* is admitted by MEC systems and $y_u(l) = 0$ means the opposite. The multi-time scale system will be discussed in detail in the following sections of this section. Let the bandwidth resource of each SBS be W_k, computing capacity of each MEC be F_k. The delay limit for semantic extraction tasks is set to $\tilde{t}_m$.

TABLE I
Main Symbol and Variable List

Notation	Description
K	Number of SBSs
U	Number of users
M	Number of tasks
T, τ	The long and short time slot
W_k	Bandwidth resource of each SBS
F_k	Computing capacity of each MEC
$\tilde{t}_m$	The delay limit for semantic extraction tasks
B_{bus}	The bus bandwidth of the hardware devices within the SBS
$\mathcal{K}$	The effective switched capacitance of the MEC server
η	The parameter for adjusting the overhead weights
y_u	Indicator of whether user u admitted by MEC systems
x_{uk}	Indicator of whether user u accessing SBS k
z_{um}	Indicator of whether task m requested by user u
r_{uk}	The uplink transmission rate of user u accessing SBS k
w_{uk}	The uplink bandwidth resource obtained by user u accessing SBS k
f_{uk}	The computing capacity that allocated by SBS k for user u
t_u^{comm}	The transmission delay of the raw data collected by user u in the wireless link
t_u^{comp}	The computing latency incurred by user u to perform tasks on SBS
t_u^{bus}	The latency generated by the bus transfer of data between the hardware within the system
a_u	The raw data that user u connected to SBS k collects
$A = (A_u)$	The random arrival process of tasks of user u
$\mathbf{Q_I}$	The amount of tasks in the current queue that need to be unloaded
$\mathbf{Q_{II}}$	The number of tasks currently in the bus transfer
$\mathbf{\Phi}$	The number of tasks offloaded but unprocessed in cache
F_{um}	The needed computing resource of task m of user u
P_{uk}	The computational power consumption of SBS k to handle the user u offloading task
v_u	The access control weighting parameter determined by the task delay limit
G_L, G_S	The long-term revenue and short-term cost

A. Communication Model

In our system, we adopt a more convenient communication model [21] which can be easily modified to other general models to complete our design. At STS *t*, the indicator variable for user *u* accessing to SBS *k* is denoted by $x_{uk} \in \{0,1\}$. Assume that a user can only access one SBS in a short time slot, then the uplink transmission rate of user *u* accessing SBS *k* is given by

$$r_{uk}(t) = w_{uk}(t) log_2 \left(1 + \frac{p_u g_{uk}(t)}{I_{uk}(t) + \sigma^2}\right), \quad (1)$$

where $w_{uk}(t)$ is uplink bandwidth resource obtained by user u, $g_{uk}(t)$ is channel gain between SBS k and user u, p_u is the transmit power from user u to SBS k, and $I_{uk}(t)$ is the co-channel interference from users connected to other SBS, i.e. $I_{uk} = \sum_{i \in K^S, i \neq k} g_{ui}(t) p_u$. σ^2 is the noise power. We denote the raw data that user u connected to SBS k collects as $a_u(t)$, such as pictures of the industrial environment that need to be semantically segmented and given instructions. The raw data is transmitted to SBS through the uplink channel. Therefore the transmission delay of the raw data collected by user u in the wireless link is

$$t_u^{comm}(t) = \frac{a_u(t)}{r_u(t)}, \tag{2}$$

where $r_u(t) = \sum_{k=1}^{K} x_{uk}(t) r_{uk}(t)$.

In our model, computing tasks arrive randomly in each STS t and form a queue, we let $A(t) = (A_u(t))$ denote the random arrival process of tasks of user u. For processing convenience, we assume that $A(t)$ is independently and identically distributed between STS t and has arrival rate λ, so that $\mathbb{E}\{A(t)\} = \lambda$ for all STSs. Let $\mathbf{Q_I}(t_s)$ denote the amount of tasks in the current queue that need to be unloaded. The dynamics of the task offloading queue can be given by

$$\begin{aligned} \mathbf{Q_I}(t+1) =& \max\left\{ \mathbf{Q_I}(t) - \sum_{u=1}^{U} y_u(l) r_u(t) \tau, 0 \right\} \\ &+ \sum_{u=1}^{U} y_u(l) A_u(t), \end{aligned} \tag{3}$$

where max represents queue accumulation of $\mathbf{Q_I}$ exists only when the queue arrival is greater than the queue departure, otherwise it is 0.

At STS t, user u transmits its collected raw data $a_u(t)$ such as various captured images, etc., to the SBS for a specific semantic extraction task m to generate semantic extracted feature data. Then the MEC feeds the semantic data back to the user for further operation via the downlink channel. In the context of the image semantic segmentation task in our scenario, the feature data is extremely small compared to raw data $a_u(t)$, therefore the downlink transmission delay can be neglected.

B. Semantic Extraction Task

The design of semantic extraction task oriented MEC systems becomes increasingly important as intelligent tasks become mainstream, especially lightweight semantic communication network combined with IoT [13]. In this paper, we consider some image recognition applications of industrial Internet where the image semantic extraction algorithm based on CNN is mainly used. The computational complexity of those applications primarily depends on not only input raw data but also CNN. Semantic extraction tasks in scenario of MEC systems needs to be considered separately from the general task to this extent.

In our system, we design a computing model for semantic extraction tasks which is specific to CNN. The needed computing resource of CNN is determined by the amount of data and model parameters associated with the input of the convolutional layer, and the network model parameters are task-specific [14]. We denote the model parameter of task m as n_m, and the specific value is determined by the number of filters of CNN. Therefore, for ease of representation, the basic computing model for semantic extraction tasks at STS t is expressed as

$$\begin{aligned} F_{um}(a_u(t)) =& log\left(\frac{a_u(t)}{3N}\right) \cdot \left(\frac{n_m a_u(t)}{N} + a_u(t) \right. \\ &\left. + \frac{n_m a_u(t)}{3}\right) + a_u(t) n_m \ (Gigacycle), \end{aligned} \tag{4}$$

where the constant 3 is the number of channels and N is the number of input feature maps. The above equation represents the approximate function of raw data volume and needed computing resource. Therefore, we set needed computing resource of user u as

$$F_u(a_u(t)) = \sum_{m=1}^{M} z_{um}(t) F_{um}(a_u(t)) \ (Gigacycle), \tag{5}$$

which means the computing amount needed processing of user u.

C. Computing Model

We propose a complete system-level computing model for semantic extraction tasks. Let the indicator variable for task m of user u at STS t be $z_{um}(t) \in \{0, 1\}$, where $z_{um}(t) = 1$ the task of user u is m and $z_{um}(t) = 0$ is the opposite. $z_{um}(t)$ is known as the content of the user's request and $\sum_{m=1}^{M} z_{um}(t) = 1$. We set the computing capacity that is allocated by SBS k for user u as $f_{uk}(t)$. Then the computing latency incurred by user u to perform tasks on SBS is given by

$$t_u^{comp}(t) = \frac{F_u(a_u(t))}{f_u(t)}, \tag{6}$$

where $f_u(t) = \sum_{k=1}^{K} x_{uk}(t) f_{uk}(t)$ (Gigacycle/s), which represents the computing capacity that MEC allocate to user u to proccess offloading tasks.

At STS t, the processing of computing tasks also requires consideration of the latency generated by the bus transfer of data between the hardware within the system [14], [15], which is denoted as

$$t_u^{bus}(t) = \frac{a_u(t)}{B_{bus}}, \tag{7}$$

where B_{bus} represents the bus bandwidth of the hardware devices within the SBS. Therefore, the total delay for user u to access SBS k to complete the task processing is given by

$$t_u(t) = t_u^{comm}(t) + t_u^{comp}(t) + t_u^{bus}(t). \tag{8}$$

We consider the computational power consumption of SBS k to handle the user u offloading task, which is expressed as

$$P_{uk}(t) = \mathcal{K}_{esc} f_{uk}^3(t), \tag{9}$$

where $\mathcal{K}_{esc}$ is the effective switched capacitance of the MEC server. Then the total power consumption of system is

$$P(t) = \sum_{u,k} x_{uk}(t) P_{uk}(t). \tag{10}$$

In our model, we consider transmission between multiple hardware connected by bus inside the MEC server. This type of transmission can also have an impact on the task queue, therefore we model this impact as the bus transfer queue. The bus transfer queue is after the task offloading queue and is independent of it. Let $\mathbf{Q_{II}}(t)$ denote the number of tasks currently in the bus transfer, the dynamics of bus transfer queue is expressed as

$$\begin{aligned}\mathbf{Q_{II}}(t+1) = &\max\left\{\mathbf{Q_{II}}(t) - \sum_{u=1}^{U} y_u(l) B_{bus}\tau, 0\right\} \\ &+ \min\left\{\sum_{u=1}^{U} y_u(l) r_u(t), \mathbf{Q_I}(t)\right\},\end{aligned} \tag{11}$$

where max represents queue accumulation of $\mathbf{Q_{II}}$ exists only when the queue arrival is greater than the queue departure, otherwise it is 0, and min represents the effect of queue arrival and queue accumulation of $\mathbf{Q_I}$ on accumulation of $\mathbf{Q_{II}}$ in tandem queue.

Thereafter computing tasks are offloaded to the MEC for processing. Assuming that there is sufficient cache in MEC systems to store offloaded but unprocessed tasks, the dynamics of the computational processing queue is given by

$$\begin{aligned}\mathbf{\Phi}(t+1) = &\max\left\{\mathbf{\Phi}(t) - \sum_{u=1}^{U} y_u(l) f_u(t), 0\right\} + \min\left\{\sum_{u} \right. \\ &\left. F_u(y_u(l) B_{bus})(t), \sum_{u} F_u(\mathbf{Q_{II}}(t))\right\},\end{aligned} \tag{12}$$

where max represents queue accumulation of $\mathbf{\Phi}$ exists only when the queue arrival is greater than the queue departure, otherwise it is 0, and min represents the effect of queue arrival and queue accumulation of $\mathbf{Q_{II}}$ on accumulation of $\mathbf{\Phi}$ in tandem queue.

D. Utility Model and Problem Formulation

In this paper, we consider the trade-off between the revenue and the cost of the optimization system, where the revenue depends on the number of admitted users associated with the admitted control weighting parameter and the cost depends on the computational energy consumption. The admitted control weighting parameter determined by the importance of users to revenue is expressed as v_u, which is given by

$$v_u(l) = \frac{\sum_{t=pl}^{p(l+1)-1} \sum_{m=1}^{M} z_{um}(t)\tilde{t}_m}{T}, \tag{13}$$

which can be seen as the importance distribution for users at LTS l determined by the average task delay limit. Then the revenue expression is

$$G_L(l) = \sum_{u=1}^{U} v_u(l) y_u(l), \tag{14}$$

where $G_L(l)$ express that the influence of admitted users to revenue is determined by weighting parameter v_u. We investigate that the computing energy cost of the system in long time slot T is

$$G_S(l) = \sum_{t=pl}^{p(l+1)-1} P(t). \tag{15}$$

In that case the system utility is expressed as

$$G(l) = G_L(l) - \eta G_S(l), \tag{16}$$

Remark 1. *From (14)(15), we notice that η is the parameter for adjusting the revenue and cost weights. Hence, different values of η may effect the trade-off between the revenue in LTS and the cost in STS and stabilize the system utility. However, other comparison algorithms do not have this characteristic of balancing revenue and cost because of different treatment methods of admission and resource allocation. Therefore, our proposed algorithm can perform well under different values of η in (16).*

Furthermore, we denote the average utility as

$$\overline{G} = \lim_{Z\to\infty} \frac{1}{Z} \sum_{l=0}^{Z-1} \mathbb{E}\{G(l)\}, \tag{17}$$

where $\overline{G}$ represent the average utility of system on all time slots and is for constructing Lyapunov stochastic optimization problem in the following.

According to our model above, we investigate the operator's utility maximization problem in MEC systems by jointly controlling system admission $\boldsymbol{y}(l)$, user association $\boldsymbol{x}(t)$, bandwidth allocation $\boldsymbol{w}(t)$ and computing capacity allocation $\boldsymbol{f}(t)$. In particular, we formulate it as the following stochastic optimization problem.

$$\begin{aligned}&\max_{\boldsymbol{y}(l),\boldsymbol{x}(t),\boldsymbol{w}(t),\boldsymbol{f}(t)} \overline{G} \\ &s.t.\quad (\mathrm{C1}): y_u(l) \in \{0,1\}, \forall u, l, \\ &\quad (\mathrm{C2}): x_{uk}(t) \in \{0,1\}, \forall u, k, t, \\ &\quad (\mathrm{C3}): \sum_{k=1}^{K} x_{uk} \leqslant 1, \forall u, t, \\ &\quad (\mathrm{C4}): y_u(l) t_u(t) \leqslant \sum_{m=1}^{M} z_{um}(t)\tilde{t}_m, \forall u, t, \\ &\quad (\mathrm{C5}): \sum_{u=1}^{U} x_{uk}(t) w_{uk}(t) \leqslant W_k, \forall k, t, \\ &\quad (\mathrm{C6}): \sum_{u=1}^{U} x_{uk}(t) f_{uk}(t) \leqslant F_k, \forall k, t, \\ &\quad (\mathrm{C7}): \overline{Q}_I < \infty, \overline{Q}_{II} < \infty, \overline{\mathbf{\Phi}} < \infty, \forall t.\end{aligned} \tag{18}$$

In (18), (C2) and (C3) indicate that a user can only access one SBS. (C4) is the delay requirements of tasks. (C5) and (C6) denote the limit of bandwidth and computing capacity of each SBS. (C7) represents that the data rate should be greater than or equal to the arrival rate of all data queues and processing queues, i.e. the mean rate stability [22].

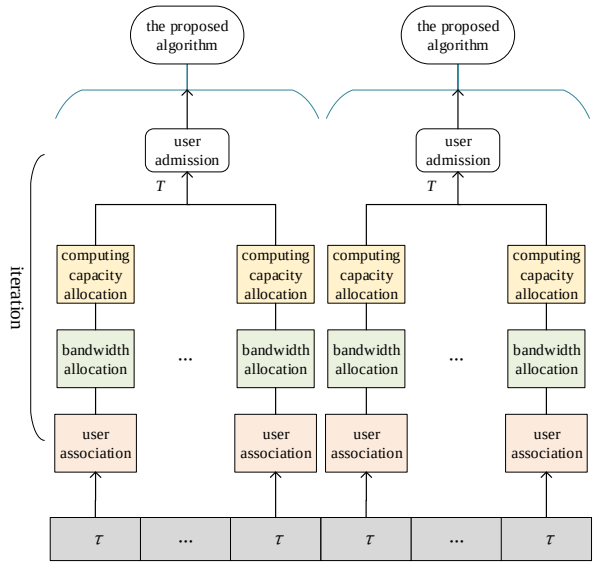

Fig. 2. Framework of the two time-slot algorithm.

III. Problem Solution and Algorithm Design

Since our optimization problem (18) is stochastic and complex in temporal domain and mixed in multi-time scale, we need to solve it by decomposing the two time-slot problem into many single time-slot subproblems with the help of Lyapunov framework as illustrated in Fig. 2. Besides, an algorithm for the user admission problem will be designed. Next, we will show the proposed algorithm is capable of achieving the revenue-cost trade-off in MEC systems.

A. The Lyapunov Optimization-Based Algorithm

We propose a algorithm based on the Lyapunov optimization and substitute the queue formula proposed in Section II into Lyapunov framework. Let $\Theta(t) = [\mathbf{Q_I}(t), \mathbf{Q_{II}}(t), \mathbf{\Phi}(t)]$ be a concatenated vector, and we define the Lyapunov function as

$$L(\Theta(t)) = \frac{1}{2}\left[\mathbf{Q_I}^2(t) + \mathbf{Q_{II}}^2(t) + \mathbf{\Phi}^2(t)\right]. \tag{19}$$

Then the LTS conditional Lyapunov drift $\Delta_T(\Theta(l))$ is given by

$$\Delta_T(\Theta(l)) = E[L(\Theta(l+T) - \Theta(l))|\Theta(l)], \tag{20}$$

where $\Theta(l) = \{\mathbf{Q_I}(t), \mathbf{Q_{II}}(t), \mathbf{\Phi}(t), t \in [l, l+T-1]\}$. Then, the drift-plus-penalty expression of (19) can be expressed as

$$\Delta_T(\Theta(l)) - V\mathbb{E}\{G(l)|\Theta(l)\}. \tag{21}$$

Remark 2. *From (21), we notice that the control parameter $V > 0$ which is from normalized form of Lyapunov optimization represents the extent of drift-plus-penalty and controls the weight of penalty. In our proposed algorithm, the larger parameter V increases the weight of penalty, i.e., system utility and causes a increase in the final optimization value. Hence, adjusting parameter V can balance the importance of queue stability and system utility and acquire an ideal result we want.*

We derive the following theorem to provide an upper bound on the above drift-plus-penalty expression.

Theorem 1: Suppose $G(t)$ is i.i.d. over slots. For arbitrary $\boldsymbol{y}(l)$, $\boldsymbol{x}(t)$, $\boldsymbol{w}(t)$, $\boldsymbol{f}(t)$, all parameters $V > 0$, and all possible values of $\Theta(l)$, $\Delta_T(\Theta(l)) - V\mathbb{E}\{G(l)|\Theta(l)\}$ is upper bounded by

$$\begin{aligned}
&\Delta_T(\Theta(l)) - V\mathbb{E}\{G(l)|\Theta(l)\} \\
&\leqslant C - \sum_{t=pl}^{p(l+1)-1} \mathbf{Q_I}(t)\mathbb{E}\left\{\left[\sum_{u=1}^{U} y_u(l) r_u(t)\tau - \sum_{u=1}^{U} y_u(l) A_u(t)\right]\right. \\
&\left.\Big|\Theta(l)\right\} - \sum_{t=1}^{l+T-1} \mathbf{Q_{II}}(t)\mathbb{E}\left\{\left[\sum_{u=1}^{U} y_u(l) B_{bus}\tau - \sum_{u=1}^{U} y_u(l) r_u(t)\right]\right. \\
&\left.\Big|\Theta(l)\right\} - \sum_{t=1}^{l+T-1} \mathbf{\Phi}(t)\mathbb{E}\left\{\left[\sum_{u=1}^{U} y_u(l) f_u(t) - \sum_{u} F_u(y_u(l)B_{bus})\right.\right. \\
&\left.\left.(t)\right]\Big|\Theta(l)\right\} - V\mathbb{E}\left\{\left[G_L(l) - \eta \sum_{t=pl}^{p(l+1)-1} P(t)\right]\Big|\Theta(l)\right\},
\end{aligned}$$

where

$$\begin{aligned}
C \geqslant \frac{1}{2}\Bigg\{&\left[\sum_{u=1}^{U} y_u(l) \sum_{t=pl}^{p(l+1)-1} r_u(t)\tau\right]^2 + \left[\sum_{u=1}^{U} y_u(l) \sum_{t=pl}^{p(l+1)-1}\right. \\
&\left. A_u(t)\right]^2 + \left[\sum_{u=1}^{U} y_u(l) \sum_{t=pl}^{p(l+1)-1} B_{bus}\tau\right]^2 + \max_{u\in U^S(l)}\left[y_u(l) \sum_{t=pl}^{p(l+1)-1}\right. \\
&\left. r_u(t)\right]^2 + \left[\sum_{u=1}^{U} y_u(l) \sum_{t=pl}^{p(l+1)-1} f_u(t)\right]^2 + \max_{u\in U^S(l)}\left[\sum_{t=pl}^{p(l+1)-1}\right. \\
&\left. F_u(y_u(l)B_{bus})(t)\right]^2\Bigg\}
\end{aligned} \tag{22}$$

Proof: Please refer to Appendix A.

The stochastic optimization theory indicates that a stochastic optimization problem can be solved by minimizing the upper bound of its drift-plus-penalty expression subject to the same constraints except the stability one in [22]. Therefore, we need to minimize right-hand-side of (III-A) to solve (18) subject to (C1)-(C6), because (C7) is a stability constraint. Therefore, the original multi-time scale optimization problem for long-term revenue can be equivalently transformed into an optimization of revenue on multiple LTSs after the above process. Then, we can get the following optimization problem which is expressed by

$$\begin{aligned}
&\max_{\boldsymbol{y}(l),\boldsymbol{x}(t),\boldsymbol{w}(t),\boldsymbol{f}(t)} [\mathbf{Q_{II}}(t) - \mathbf{Q_I}(\mathbf{t})] \sum_{u=1}^{U} y_u(l) r_u(t) - \mathbf{Q_{II}}(t) \\
&\sum_{u,k} y_u(l) B_{bus}\tau - \mathbf{\Phi}(t) \sum_{u=1}^{U} y_u(l) f_u(t) + \mathbf{\Phi}(t) \sum_{u} F_u(y_u(l) \\
&B_{bus})(t) + V\eta P(t) \\
&s.t. (\text{C1}) - (\text{C6}).
\end{aligned} \tag{23}$$

From the principle of opportunistically minimizing an expectation, minimizing $\boldsymbol{f}(t)$ can ensure that $\mathbb{E}\{\boldsymbol{f}(t)|\Theta(t)\}$ is minimized. Therefore for the objective function in (23), we

can obtain by ignoring constant C, $\mathbf{Q_I}(t)\sum_{u=1}^{U} y_u(l)A_u(t)$ and $G_L(l)$ in (III-A) and removing the conditional expectations in (III-A). Since user admission is in long time scale, we use subproblem separation to separate the long and short time scale problems for ease of processing. The subproblems in long and short time scale will be solved later by iterative integration. Obviously, user association, bandwidth allocation and computing capacity allocation are highly coupled with each other in (23). We further decompose these optimization variables to develop low-complexity algorithms in the following subsections.

B. Solution of Resource Allocation Subproblem in Short Time Scale

We obtain the solution of the coupled optimization problem (23) by integrating the algorithms through iterative optimization. Under given user association $\boldsymbol{x}(t)$ and bandwidth allocation $\boldsymbol{w}(t)$, the computing resource allocation subproblem can be expressed by

$$\begin{aligned}
&\max_{\boldsymbol{f}(t)} \boldsymbol{\Phi}(t)\sum_{u=1}^{U} y_u f_u(t) - V\eta\sum_{u,k} x_{uk}(t)\kappa_{esc} f_{uk}^3(t)\\
&s.t.\ \ (\mathrm{C4}): f_u(t) \geqslant \frac{y_u F_u(a_u(t))}{\sum_{u=1}^{U} z_{um}\tilde{t}_m - y_u\frac{a_u(t)}{B_{bus}} - y_u\frac{a_u(t)}{r_U(t)}}, \forall u,t,\\
&\qquad (\mathrm{C6}): \sum_{u=1}^{U} x_{uk}(t) f_{uk}(t) \leqslant F_k, \forall k,t.
\end{aligned} \tag{24}$$

where the other terms of equation (23) are constants under the above condition. Obviously, the objective function of (24) is concave and its constraints are linear, so it is a convex optimization problem. Therefore we can obtain the optimized solution $f^*(t)$ directly through the convex optimization method [23] in polynomial time using standard CVX tools [24].

Under given $\boldsymbol{w}(t)$ and $\boldsymbol{f}(t)$ we can obtain the user association problem which is denoted as

$$\begin{aligned}
&\min_{\boldsymbol{x}(t)} [\mathbf{Q_{II}}(t) - \mathbf{Q_I}(t)]\sum_{u=1}^{U} y_u r_u(t) + V\eta\sum_{u,k} x_{uk}(t)\kappa_{esc} f_{uk}^3(t)\\
&s.t.\ \ (\mathrm{C2}) - (\mathrm{C3}).
\end{aligned} \tag{25}$$

It can be converted to

$$\begin{aligned}
&\min_{\boldsymbol{x}(t)} \left[(\mathbf{Q_{II}}(t) - \mathbf{Q_I}(t))\sum_{u=1}^{U} y_u r_{uk}(t) + V\eta\sum_{u,k}\kappa_{esc} f_{uk}^3(t)\right] x_{uk}(t)\\
&s.t.\ \ (\mathrm{C2}) - (\mathrm{C3}),
\end{aligned} \tag{26}$$

whose optimal solution can be expressed as

$$x_{uk} = \begin{cases} 1, & k = k^*, \\ 0, & k \neq k^*, \end{cases} \tag{27}$$

where

$$k^* = \arg\min_{k\in K^S} \left\{(\mathbf{Q_{II}}(t) - \mathbf{Q_I}(t)) y_u r_{uk}(t) + V\eta\kappa_{esc} f_{uk}^3(t)\right\} \tag{28}$$

Next, if user association $\boldsymbol{x}(t)$ and computing capacity allocation $\boldsymbol{f}(t)$ are known, then the bandwidth allocation subproblem can be given by

$$\begin{aligned}
&\min_{\boldsymbol{w}(t)} \sum_{u,k}[(\mathbf{Q_{II}}(t) - \mathbf{Q_I}(t)]x_{uk}(t)\log_2\left(1 + \frac{g_{uk}P_u}{I_{uk}(t)+\sigma^2}\right) w_{uk}(t)\\
&s.t.\ \ (\mathrm{C4})': \sum_{k} x_{uk}(t) w_{uk}(t)\log_2\left(1 + \frac{g_{uk}P_u}{I_{uk}(t)+\sigma^2}\right) \geqslant \frac{a_u(t)}{\tilde{t}_m - t_u^{comp}(t) - t_u^{bus}(t)}, \forall u,t,\\
&\qquad (\mathrm{C5}): \sum_{u\in U} x_{uk}(t) w_{uk}(t) \leqslant W_k, \forall k,t.
\end{aligned} \tag{29}$$

It can be seen that the objective function of the problem is a linear function, and the constraints are linear, so it is a linear programming problem, and the solution $\boldsymbol{w}(t)$ can be directly obtained by optimization methods such as the interior point method [25] by using standard CVX tools.

We use the idea of the greedy algorithm to iterate the above three solutions for the subproblem and arrive at the suboptimal solution, which is summarized in Algorithm 1.

Algorithm 1 Iterative Algorithm for Resource Allocation for Given Admission.

repeat
  Set initial STS $t = l$ and obtain the current queue state $\mathbf{Q_I}(t)$, $\mathbf{Q_{II}}(t)$, and $\boldsymbol{\Phi}(t)$.
  Set $q = 0$, user association $x^0(t)$ and computing capacity allocation $f^0(t)$.
  Set $l = 0$ and the iteration constraints $\varepsilon > 0$.
  Obtain $w^0(t)$ by solving bandwidth allocation subproblem through $x^0(t)$ and $f^0(t)$.
  Obtain the value of optimization problem (23), i.e. $N^0(t)$.
  repeat
    $q = q + 1$.
    Obtain $x^q(t)$ by solving user association subproblem through $w^{q-1}(t)$ and $f^{q-1}(t)$.
    Obtain $f^q(t)$ by solving computing capacity allocation subproblem through $w^{q-1}(t)$ and $x^q(t)$.
    Obtain $w^q(t)$ by solving bandwidth allocation subproblem through $x^q(t)$ and $f^q(t)$.
    Obtain the value of $N^q(t)$.
  until $\left|N^q(t) - N^{q-1}(t)\right| \leqslant \varepsilon$.
  $t = t + 1$.
  Update $\mathbf{Q_I}(t)$, $\mathbf{Q_{II}}(t)$, and $\boldsymbol{\Phi}(t)$.
until $t = l + T - 1$.

C. Solution of User Admission Subproblem in Long Time Scale

The original problem (18) can be decomposed into two subproblems on the time scale, one is (23) and the other can be denoted as

$$\begin{aligned} &\max_{\boldsymbol{y}(l)} G_L(l) - \eta G_S(l) \\ &s.t. \quad (\mathrm{C1}) : y_u(l) \in \{0,1\}, \forall u, \\ &\qquad (\mathrm{C8}) : \text{feasibility of problem (23).} \end{aligned} \tag{30}$$

Obviously, this problem is a nonlinear 0-1 integer programming problem, and the constraint (C8) needs to determine the solvability of (23). Therefore it cannot be solved by conventional methods. According to the above condition that the random arrival process of tasks of user u is independently and identically distributed between STSs, the constraint (C8) can be transformed into (C4) and (C7). The Lyapunov architecture makes the above solution , i.e. Algorithm 2, satisfy (C7), so we solely need to transform (C8) into (C4) there. Then, (30) can be expressed as

$$\begin{aligned} &\max_{\boldsymbol{y}(l)} \sum_{u=1}^{U} v_u(l) y_u(l) - \eta \sum_{t=pl}^{p(l+1)-1} P(t) \\ &s.t. \quad (\mathrm{C1}) : y_u(l) \in \{0,1\}, \forall u, \\ &\qquad (\mathrm{C4}) : y_u(l) t_u(t) \leqslant \sum_{m=1}^{M} z_{um}(t) \tilde{t}_m, \forall u, t \in [l, l+T-1]. \end{aligned} \tag{31}$$

The optimization problem (31) is a linear 0-1 integer programming problem and can be solved by iterating with (23). The whole procedure is shown in Algorithm 2.

Algorithm 2 User Admission Algorithm.

1: Set initial LTS l.
2: **repeat**
3: Set $j = 0$, and determine the initial $y_u^0(l)$ that satisfies the condition.
4: Solve (23) through Algorithm 1 and obtain $\boldsymbol{x}(t)$, $\boldsymbol{w}(t)$ and $\boldsymbol{f}(t)$.
5: $j = j + 1$.
6: Obtain the iteration consequence $y_u^j(l)$ by solving (31).
7: Calculate the value of $G(l)$.
8: **until** $G(l)$ no longer changes significantly.
9: $l = l + 1$.

D. Dynamic Solution to the Optimization Problem

As mentioned above, we decompose the original complex stochastic optimization problem (18) into two subproblems. As shown in Fig.2, we propose a short-time-slot resource allocation algorithm and a long-time-slot user admission algorithm to solve it. Due to the specificity of the two time-slot iteration algorithm, we need to solve the problem offline and adopt the strategies online. The detailed procedure to solve (18) is summarized in Algorithm 3.

Algorithm 3 Multi-time Scale User Admission and Resource Allocation algorithm.

1: In each LTS $l = iT, i = 0, 1, ...$ obtain the current queue state $\mathbf{Q_I}(t_k)$.
2: Determine the user admission $y_u(l)$ by calling Algorithm 2.
3: In each STS $t \in [l, l+T]$, obtain user association $\boldsymbol{x}(t)$, bandwidth allocation $\boldsymbol{w}(t)$ and computing capacity allocation $\boldsymbol{f}(t)$ according to (23) by calling Algorithm 1.
4: $t = t + 1$.
5: Update the current queue value $\mathbf{Q_I}(t)$, $\mathbf{Q_{II}}(t)$, and $\mathbf{\Phi}(t)$ according to (3), (11) and (12).

E. Analysis of the Proposed Algorithms

In this subsection, we analyse the temporal computational complexity and the convergence of the proposed algorithms.

In Algorithm 1, we adopt alternating iteration of three subproblems and obtain the solutions in closed form by convex optimization. According to the greedy algorithm and convex optimization theory [26], iteration of three convex subproblems can ensure $|N^q(t) - N^{q-1}(t)| \leqslant \varepsilon$ quickly. Therefore, the convergence of Algorithm 1 is obvious but only sub-optimality can be guaranteed [5]. From the perspective of complexity, the complexity of (24), (25) and (29) are $O(U^3K)$, $O(UK)$ and $O((UK)^{3.5}))$ [26]. We assume the number of iterations is L_1, then the complexity of Algorithm 1 is $O((U^3K + UK + (UK)^{3.5})L_1)$. This decoupled algorithm can make good use of convex optimization methods to solve complex coupling problem (23).

In Algorithm 2, since (31) a linear 0-1 integer programming problem and iterating with (23), we assume the number of iterations is L_2, then the complexity is $O((U^2 + \gamma_1)L_2)$ where γ_1 is the complexity of Algorithm 1. Obviously, the convergence of the Algorithm 2 depends on Algorithm 1 since the iteration is performed with it. Therefore, the proposed algorithm can reach convergence after several iterations as Algorithm 1 converges quickly. Algorithm 3 indicates that above algorithms run on time slots, therefore at LTS l the complexity of Algorithm 3 is $O((U^2+\gamma_1)L_2p)$. In the way, the complex stochastic optimization problem (18) is decomposed into low-complexity subproblems iteratively solved.

IV. Simulation Result

In this section, we first set the simulation parameters and then demonstrate simulation results to evaluate the performance of the proposed algorithms.

A. Simulation Parameters

As mentioned above, we consider system level simulation of uplink in a small cell F-RAN according to a 3GPP normative document of small cell network [27]. Four SBSs are deployed in a small cell area, with a total coverage area of 200m$\times$200m. The SBSs provide admission and resource allocation for users. We model the path loss of radio access link of the small cell network and use a hexagonal cellular deployment model. The distance between the user and the SBS is within the standard

of the 3GPP document and only the outdoor access link exists. At STS t, let $d_{uk}(t)$ be the distance between SBS k and user u, and users all moves randomly within the area at a speed of 3km/h. Note that the path loss between SBS k and user u is dependent on the link state of LoS and NLoS. When it is a LoS link, the path loss is given by

$$\begin{aligned}\mu_{uk}^{LoS/NLoS}(t) =& 22.0log_{10}(d_{uk}(t)) + 28.0 \\ &+ 20log_{10}(F^q),\end{aligned} \tag{32}$$

and when it is a NLoS link, the path loss is given by

$$\begin{aligned}\mu_{uk}^{LoS/NLoS}(t) =& 36.7log_{10}(d_{uk}(t)) + 22.7 \\ &+ 26log_{10}(F^q),\end{aligned} \tag{33}$$

where F^q indicates the carrier frequency. The LoS probability that determines the LoS/NLoS link state is denoted as

$$p_{uk}^{LoS}(t) = \min\left(\frac{18}{d_{uk}(t)}, 1\right)\left(1 - e^{-\frac{d_{uk}(t)}{36}}\right) + e^{-\frac{d_{uk}(t)}{36}}, \tag{34}$$

therefore the NLoS probability is $p_{uk}^{NLoS}(t) = 1 - p_{uk}^{LoS}(t)$. Then the channel gain can be expressed as

$$g_{uk}(t) = \left(p_{uk}^{LoS}10^{\mu_{uk}^{LoS}} + p_{uk}^{NLoS}10^{\mu_{uk}^{NLoS}}\right)^{-1}. \tag{35}$$

In our proposed algorithm, we notice that the parameter p of $T = p\tau$ in our algorithm represents the relationship between the length of LTSs and STSs. Hence, too small p will affect the effect of multi-time scales and optimization of the algorithm and too large p will increase the running time of our algorithm but the improved algorithm performance is not significant. Therefore, we set the appropriate values of T and τ. Part of simulation parameters are summarized in Table II [28].

TABLE II
PART OF SIMULATION PARAMETERS

Parameter	Value
Long Time Slot (LTS), T	1 s
Short Time Slot (STS), τ	0.1 s
Transmit power, p_u	37 dBm
Bandwidth resource, W_k	10 MHz
Computing capacity, F_k	200 Gigacycle/s
The noise power, σ^2	-100 dBm
The carrier frequency, F^q	3.5 GHz
Bus bandwidth, B_{bus}	10 Gbps
Arrival rate, λ	50
The total number of users, U	60
Adjusting parameter, η	10^{-6}
The number of iterations, L_1	50

According to the computing delay requirements of some services of ultra reliable low latency communications and combining task scenarios [29], [30], we set the basis delay requirement as 20 ms. Furthermore, we assume three task types whose delay requirements and model parameters progressive by task type and are equally distributed to the task set. The delay limits and model parameters of the tasks are shown in Table III.

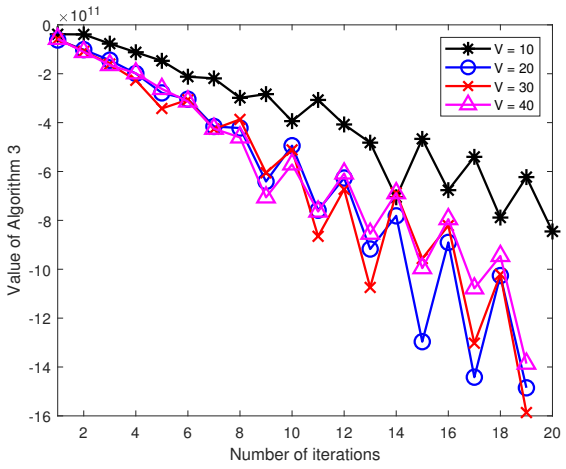

Fig. 3. Convergence of the proposed Algorithm

TABLE III
TASK PARAMETERS

Type	$\tilde{t}_m$	n_m
1	20 ms	1
2	40 ms	3
3	60 ms	5

B. Convergence of the Proposed Algorithm

For convenience, we integrate the value of STS iteration and the value LTS iteration, i.e., the system revenue and cost at each STS, to show our overall convergence. Fig. 3 shows the convergence of the proposed algorithm under different parameters V. From Fig. 3, we can see that the convergence of the proposed Algorithm is fast and the trend is basically fixed after convergence. Since Algorithm 1 is nested into Algorithm 2 for computing, Algorithm 1 will stop as system utility of Algorithm 2 converges and the trend will be basically fixed after convergence. Also, a larger V indicates a larger penalty weight in Lyapunov drift plus penalty which increases the weight of $G_S(l)$ relative to the overall queue stability and thus increases the impact of power consumption, therefore it leads to a change in the value of the utility which increases and a more volatile fluctuation after convergence as V becomes larger, which is a good proof of **Remark 2**.

C. Performance of the Proposed Algorithm

To verify the performance of the proposed algorithm, we will consider the following schemes:

- Fixed Allocation (FA): The scheme only optimizes user admission.
- Fixed Channel (FC): The scheme is that user association and bandwidth allocation are fixed and optimize access variables and computational resource allocation.

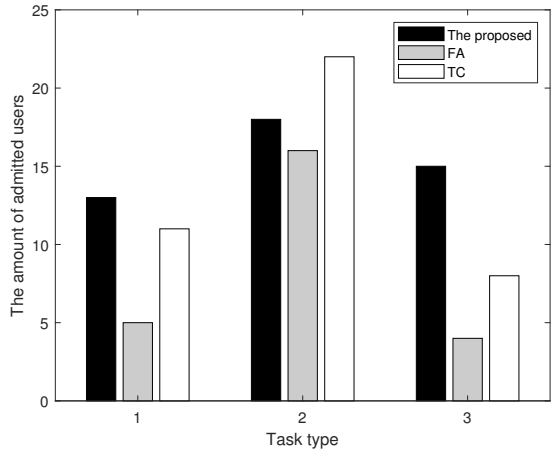

Fig. 4. The amount of admitted users varying with task types.

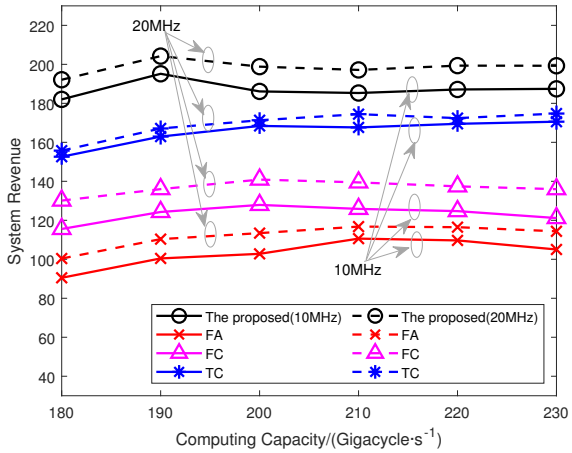

Fig. 6. System utility varying with computing capacity.

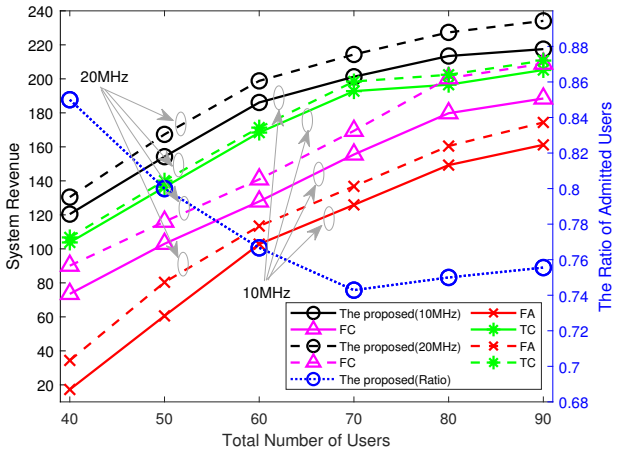

Fig. 5. System utility varying with total number of users.

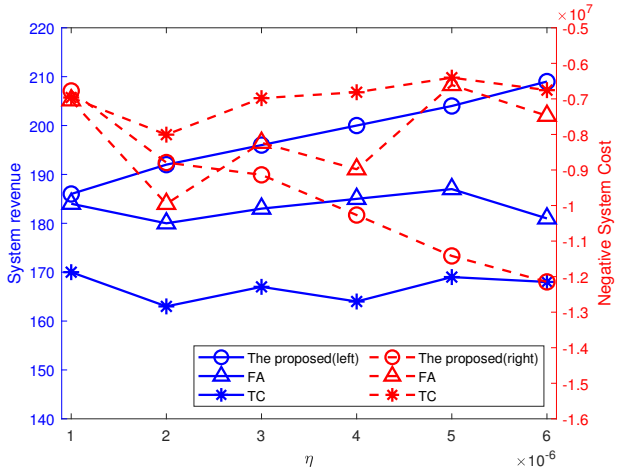

Fig. 7. Trade-off between system revenue and system cost vs. η.

- Traditional Computing (TC) [5]: The scheme allocates computing resources based on input data size according to the traditional computing model.

We demonstrate the amount of admitted users varying with three different task types in Fig. 4. As we can see, different values of $\widetilde{t}_m$ affect the amount of admitted users. From (13), we notice that $\widetilde{t}_m$ affects the system revenue and can cause that our algorithm will choose more valuable users. This characteristic is also shown in figure and the amount of admitted users of task 2 is more than other tasks. In our system, integrated user value is affected by the needed computing resource F_{um} besides $\widetilde{t}_m$ and therefore users of task 2 become the most valuable in our parameters. However, our proposed algorithm can handle the difference between three types of tasks well and make user admission stable as we can see in Fig. 4 compared with other contrast algorithms.

The characteristics of system utility with total number of users under different bandwidth values as Fig. 5 shows, it can be found that it increases with the total number of users and slower after the total number of users is greater than 70. To better indicate the momentum of the system utility, we add the blue dotted line which represent the ratio of admitted users of our proposed algorithm as right vertical axis in Fig. 5 shows. When the total number of users is small, the resources are sufficient and resource allocation can be efficient, so the system utility will increase quickly as the total number of users increases. However, when the total number of users is large, the resources are limited and resouce alloction will not be efficient, therefore the rising tendency of utility will be reduced. The radio of admitted users will decrease first and then become stable after the total number of users is 70, which shows the increasing momentum of system utility will become slow when the total number of users become large. The comparison algorithms all have this property, but the trend is different for different algorithms. To make figure concise as far as possible, we do not show the ratio of admitted users of the comparison algorithms which has been compared in Fig. 4. There is a slight but insignificant increase in system utility for higher bandwidth values, therefore the bandwidth value change has a small impact on the system.

We compare system utility with computing capacity under different bandwidth values in Fig. 6, and it can be seen that

there is a maximum value in $F_k = 190$ Gigacycle/s. This is because system utility depends on the number of association users and power consumption and our algorithm needs to make a trade-off between them. When F_k reaches a certain amount, our proposed algorithm can take it to a better trade-off utility value. However when F_k continues to rise, the fairness design of this algorithm will allow more users to admit and thus generate more power consumption. Therefore it will lead to a decrease in the value of utility. While F_k increases to a certain magnitude, the number of associated users will not continue to increase due to bandwidth resource, so the change is no longer significant. This property is also present in comparison algorithms, mainly because the utility defined in this paper are indirectly influenced by the allocation of multiple resources. The increase in computing resource has little effect on total utility, so its change is insignificant and even has the characteristic of decreasing with its increase. We can see in this graph again that higher bandwidth values do not have a significant impact on total utility.

We plot the trade-off between system revenue and system cost vs. η in Fig. 7. The system utility (16) indicate that our algorithm can balance system revenue and cost, and in this figure we can see that our proposed algorithm can get better trade-off than comparison algorithm while η increases, which verifies **Remark 1**. There is obvious increase on system revenue and decrease on negative system cost when η increases in our proposed algorithm but this trend is not that significant in other comparison, which means our proposed algorithm can significantly attach importance to user admission and stabilize the system utility. Therefore, the system utility is increased by balancing revenue at LTSs and cost at STSs in the proposed algorithm and make systems increasingly stable compared with other algorithms.

V. Conclusion

In this paper, we studied the the dynamical resource allocation problem of specific characteristic of tasks in MEC systems. Specially, the stochastic optimization problem we proposed was decomposed into user admission at LTS and resource allocation at STS by the Lyapunov optimization technique and we decoupled the optimization variables for efficient algorithm design and solve each subproblem at low complexity. Simulation results has demonstrated that, compared with the benchmarks, the proposed algorithm improves the performance of user admission and resource allocation efficiently and achieves a flexible trade-off between system revenue and cost at multi-time scales and considering semantic extraction tasks.

Appendix
A Proof of Theorem 1

First of all, we have that $\{\max[A-B,0]+C\}^2 \leq A^2 + B^2 + C^2 - 2A(B-C)$ always holds if $A \geq 0$, $B \geq 0$ and $C \geq 0$. Suppose $V > 0$, then squaring both sides of (3) yields

$$\mathbf{Q_I}(l+T-1)^2 \leqslant \mathbf{Q_I}(t)^2 + \left[\sum_{u,k} y_u(l) \sum_{t=pl}^{p(l+1)-1} x_{uk}(t) r_{uk}(t)\tau\right]^2 + \max_{u\in U^S(l),k\in K^S}\left[\sum_{t=pl}^{p(l+1)-1} r_{uk}(t)\right]^2 + \left[\sum_{u=1}^{U} y_u(l) \sum_{t=pl}^{p(l+1)-1} A_u(t)\right]^2 - 2\sum_{t=pl}^{p(l+1)-1} \mathbf{Q_I}(t)\left[\sum_{u,k} y_u(l) x_{uk}(t) r_{uk}(t)\tau - \sum_{u=1}^{U} y_u(l) A_u(t)\right]. \tag{36}$$

For (11), similarly, we have

$$\mathbf{Q_{II}}(l+T-1)^2 \leqslant \mathbf{Q_{II}}(t)^2 + \left[\sum_{u,k} y_u(l) \sum_{t=pl}^{p(l+1)-1} B_{bus}\tau\right]^2 + \max_{u\in U^S(l),k\in K^S}\left[y_u(l)\sum_{t=pl}^{p(l+1)-1} r_{uk}(t)\right]^2 - 2\sum_{t=pl}^{p(l+1)-1} \mathbf{Q_{II}}(t)\left[\sum_{u,k} y_u(l) B_{bus}\tau - \sum_{u=1}^{U} y_u(l) x_{uk}(t) r_{uk}(t)\right]. \tag{37}$$

For (12), we also have

$$\mathbf{\Phi}(l+T-1)^2 \leqslant \mathbf{\Phi}(t)^2 + \left[\sum_{u,k} y_u(l) \sum_{t=pl}^{p(l+1)-1} x_{uk}(t) f_{uk}(t)\right]^2 + \max_{u\in U^S(l)}\left[\sum_{t=pl}^{p(l+1)-1}\sum_{m=1}^{M} z_{um}(t) F_{um}(y_u(l)B_{bus})\right]^2 - 2\sum_{t=pl}^{p(l+1)-1} \mathbf{\Phi}(t)\left[\sum_{u,k} y_u(l) x_{u,k}(t) f_{u,k}(t) - \sum_{u} F_u(y_u(l)B_{bus})(t)\right]. \tag{38}$$

By organizing the above inequalities, we can obtain

$$\frac{\mathbf{Q_I}(l+T-1)^2 - \mathbf{Q_I}(t)^2}{2} \leqslant \frac{1}{2}\left\{\left[\sum_{u,k} y_u(l) \sum_{t=pl}^{p(l+1)-1} x_{uk}(t) r_{uk}(t)\tau\right]^2 + \left[\sum_{u=1}^{U} y_u(l) \sum_{t=pl}^{p(l+1)-1} A_u(t)\right]^2\right\} - \sum_{t=pl}^{p(l+1)-1} \mathbf{Q_I}(t)\left[\sum_{u,k} y_u(l) x_{uk}(t) r_{uk}(t)\tau - \sum_{u=1}^{U} y_u(l) A_u(t)\right], \tag{39}$$

$$\frac{\mathbf{Q_{II}}(l+T-1)^2 - \mathbf{Q_{II}}(t)^2}{2} \leqslant \frac{1}{2}\left\{\left[\sum_{u,k} y_u(l) \sum_{t=pl}^{p(l+1)-1} B_{bus}\tau\right]^2 + \max_{u\in U^S(l),k\in K^S}\left[y_u(l)\sum_{t=pl}^{p(l+1)-1} r_{uk}(t)\right]^2\right\} - \sum_{t=pl}^{p(l+1)-1} \mathbf{Q_{II}}(t)\left[\sum_{u,k} y_u(l) B_{bus}\tau - \sum_{u=1}^{U} y_u(l) x_{uk}(t) r_{uk}(t)\right], \tag{40}$$

and

$$\frac{\mathbf{\Phi}(l+T-1)^2 - \mathbf{\Phi}(t)^2}{2} \leqslant \frac{1}{2}\left\{\left[\sum_{u,k} y_u(l) \sum_{t=pl}^{p(l+1)-1} x_{uk}(t) f_{uk}(t)\right]^2 + \max_{u\in U^S(l)}\left[\sum_{t=pl}^{p(l+1)-1} F_u(y_u(l)B_{bus})(t)\right]^2\right\} - \sum_{t=pl}^{p(l+1)-1} \mathbf{\Phi}(t)$$

$$\left[\sum_{u,k} y_u(l)x_{u,k}(t)f_{u,k}(t) - \sum_u F_u(y_u(l)B_{bus})(t)\right]. \quad (41)$$

Summing the above three equations yields

$$\begin{aligned}
&L(\Theta(l+T)-\Theta(l))\\
&\leqslant \frac{1}{2}\Bigg\{\left[\sum_{u=1}^{U} y_u(l)\sum_{t=pl}^{p(l+1)-1} r_u(t)\tau\right]^2 + \left[\sum_{u=1}^{U} y_u(l)\sum_{t=pl}^{p(l+1)-1} A_u(t)\right]^2 + \left[\sum_{u=1}^{U} y_u(l)\sum_{t=pl}^{p(l+1)-1} B_{bus}\tau\right]^2 + \max_{u\in U^S(l)}\left[y_u(l)\sum_{t=pl}^{p(l+1)-1} r_u(t)\right]^2 + \left[\sum_{u=1}^{U} y_u(l)\sum_{t=pl}^{p(l+1)-1} f_u(t)\right]^2 + \max_{u\in U^S(l)}\left[\sum_{t=pl}^{p(l+1)-1} F_u(y_u(l)B_{bus})(t)\right]^2\Bigg\} - \sum_{t=pl}^{p(l+1)-1}\mathbf{Q_I}(t)\left[\sum_{u,k} y_u(l)x_{uk}(t)r_{uk}(t)\tau - \sum_{u=1}^{U} y_u(l)A_u(t)\right] - \sum_{t=pl}^{p(l+1)-1}\mathbf{Q_{II}}(t)\left[\sum_{u,k} y_u(l)B_{bus}\tau - \sum_{u=1}^{U} y_u(l)x_{uk}(t)r_{uk}(t)\right] - \sum_{t=pl}^{p(l+1)-1}\mathbf{\Phi}(t)\left[\sum_{u,k} y_u(l)x_{u,k}(t)f_{u,k}(t) - \sum_u F_u(y_u(l)B_{bus})(t)\right]. \quad (42)
\end{aligned}$$

We take conditional expectation to the above inequality and can obtain

$$\begin{aligned}
&\Delta_T(\Theta(l)) - V\mathbb{E}\{G(l)|\Theta(l)\}\\
&\leqslant C - \sum_{t=pl}^{p(l+1)-1}\mathbf{Q_I}(t)\mathbb{E}\left\{\left[\sum_{u=1}^{U} y_u(l)r_u(t)\tau - \sum_{u=1}^{U} y_u(l)A_u(t)\right]\middle|\Theta(l)\right\} - \sum_{t=1}^{l+T-1}\mathbf{Q_{II}}(t)\mathbb{E}\left\{\left[\sum_{u=1}^{U} y_u(l)B_{bus}\tau - \sum_{u=1}^{U} y_u(l)r_u(t)\right]\middle|\Theta(l)\right\} - \sum_{t=1}^{l+T-1}\mathbf{\Phi}(t)\mathbb{E}\left\{\left[\sum_{u=1}^{U} y_u(l)f_u(t) - \sum_u F_u(y_u(l)B_{bus})(t)\right]\middle|\Theta(l)\right\} - V\mathbb{E}\left\{\left[G_L(l) - \eta\sum_{t=pl}^{p(l+1)-1} P(t)\right]\middle|\Theta(l)\right\}, \quad (43)
\end{aligned}$$

where

$$\begin{aligned}
C \geqslant \frac{1}{2}\Bigg\{&\left[\sum_{u=1}^{U} y_u(l)\sum_{t=pl}^{p(l+1)-1} r_u(t)\tau\right]^2 + \left[\sum_{u=1}^{U} y_u(l)\sum_{t=pl}^{p(l+1)-1} A_u(t)\right]^2 + \left[\sum_{u=1}^{U} y_u(l)\sum_{t=pl}^{p(l+1)-1} B_{bus}\tau\right]^2 + \max_{u\in U^S(l)}\left[y_u(l)\sum_{t=pl}^{p(l+1)-1} r_u(t)\right]^2\\
&+ \left[\sum_{u=1}^{U} y_u(l)\sum_{t=pl}^{p(l+1)-1} f_u(t)\right]^2 + \max_{u\in U^S(l)}\left[\sum_{t=pl}^{p(l+1)-1} F_u(y_u(l)B_{bus})(t)\right]^2\Bigg\}. \quad (44)
\end{aligned}$$

Then we complete the proof of Theorem 1.

REFERENCES

[1] Y. Guo, F. R. Yu, J. An, K. Yang, C. Yu and V. C. M. Leung, "Adaptive Bitrate Streaming in Wireless Networks With Transcoding at Network Edge Using Deep Reinforcement Learning," *IEEE Trans. Veh. Technol.*, vol. 69, no. 4, pp. 3879-3892, Apr. 2020.

[2] S. Jošilo and G. Dán, "Computation Offloading Scheduling for Periodic Tasks in Mobile Edge Computing," *IEEE Trans. Netw.*, vol. 28, no. 2, pp. 667-680, Apr. 2020.

[3] L. Lei, C. Chen, Q. Pei, S. Maharjan and Y. Zhang, "Vehicular edge computing and networking: A survey." *Mobile Netw. Appl.*, vol.26, no. 3, pp. 1145-1168, Jun. 2021.

[4] J. Du, L. Zhao, J. Feng, and X. Chu, "Computation offloading and resource allocation in mixed fog/cloud computing systems with min-max fairness guarantee," *IEEE Trans. Commun.*, vol. 66, no. 4, pp. 1594-1608, Apr. 2018.

[5] 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.

[6] S. Zarandi and H. Tabassum, "Delay Minimization in Sliced Multi-Cell Mobile Edge Computing (MEC) Systems," *IEEE Commun. Lett.*, vol. 25, no. 6, pp. 1964-1968, Jun. 2021.

[7] G. Faraci, C. Grasso, and G. Schembra, "Design of a 5G Network Slice Extension With MEC UAVs Managed With Reinforcement Learning," *IEEE J. Sel. Areas Commun.*, vol. 16, no. 7, pp. 2356-2371, Oct. 2020.

[8] J. Y. Hwang, L. Nkenyereye and N. M. Sung, "IoT service slicing and task offloading for edge computing," *IEEE Internet Things J.*, vol. 44, no. 4, pp. 1-14, Apr. 2020.

[9] X. Cao, J. Xu and R. Zhang, "Mobile edge computing for cellular-connected UAV: Computation offloading and trajectory optimization," *IEEE 19th International Workshop Signal Process. Adv. Wireless Commun. (SPAWC)*, pp. 1-5, 2018.

[10] M. A. Hossain, and N. Ansari, "Energy Aware Latency Minimization for Network Slicing Enabled Edge Computing," *IEEE Trans. Green Commun. Netw.*, pp. 1-10, May. 2021.

[11] T. Zhang, Y. Xu, J. Loo, D. Yang and L. Xiao, "Joint Computation and Communication Design for UAV-Assisted Mobile Edge Computing in IoT," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 8, pp. 5505-5516, Aug. 2020.

[12] H. Xie, Z. Qin, G. Y. Li, and B. H. Juang. "Deep learning enabled semantic communication systems," *IEEE Trans. Signal Process.*, vol. 69, pp. 2663-2675, Apr. 2021.

[13] H. Xie and Z. Qin, "A lite distributed semantic communication system for internet of things," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 1, pp. 142-153, Nov. 2020.

[14] H. Qi, E. R. Sparks and A. Talwalkar, "PALEO: A Performance Model for Deep Neural Networks," *International Conf. Learn. Representations (ICLR)*, 2016.

[15] D. Justus, J. Brennan, S. Bonner and A. S. McGough, "Predicting the computational cost of deep learning models." *IEEE International Conf. Big Data*, pp. 3873-3882, Dec. 2018.

[16] D. Bienstock, G. Muñoz and S. Pokutta, "Principled deep neural network training through linear programming," *arXiv preprint arXiv:1810.03218*, Oct. 2020.

[17] M. Bianchini and F. Scarselli, "On the complexity of neural network classifiers: A comparison between shallow and deep architectures," *IEEE Tran. Neural Netw. Learn. Syst.*, vol. 25, no. 8, pp. 1553-1565, Jan. 2014.

[18] F. Guo, F. R. Yu, H. Zhang, H. Ji, M. Liu, and V. C. M. Leung, "Adaptive resource allocation in future wireless networks with blockchain and mobile edge computing," *IEEE Trans. Wireless Commun.*, vol. 19, no. 3, pp. 1689-1703, Mar. 2020.

[19] Y. Xiao and M. Krunz, "Dynamic network slicing for scalable fog computing systems with energy harvesting," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 12, pp. 2640-2654, Dec. 2018.

[20] N. Van Huynh, D. Thai Hoang, D. N. Nguyen, and E. Dutkiewicz, "Optimal and fast real-time resource slicing with deep dueling neural networks," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 6, pp. 1455-1470, Jun. 2019.

[21] G. Sun, H. Al-Ward, G. O. Boateng and G. Liu, "Autonomous cache resource slicing and content placement at virtualized mobile edge network," *IEEE Access*, vol. 7, pp. 84727-84743, Jun. 2019.

[22] M. J. Neely, "Stochastic network optimization with application to communication and queueing systems," *Synth. Lect. Commun.*. San Rafael, CA, USA: Morgan & Claypool Publishers, 2010.
[23] S. Boyd, "Convex optimization problems," Lecture slides and notes. 2008. [Online]. Available: http://web.stanford.edu/class/ee364a/lectures.html.
[24] M. Grant, S. Boyd, and Y. Ye, "CVX: MATLAB software for disciplined convex programming," 2014. [Online]. Available: http://cvxr.com/cvx/.
[25] S. Boyd, "Interior-point methods," Lecture slides and notes. 2008. [Online]. Available: http://web.stanford.edu/class/ee364a/lectures.html.
[26] S. Boyd and L. Vandenberghe, Convex Optimization, Cambridge University Press, 2004.
[27] 3GPP, "Technical Specification Group Radio Access Network; Small cell enhancements for E-UTRA and E-UTRAN Physical layer aspects," *TR 36.872*, *Release 15*, pp. 9, 76-77, Dec. 2013.
[28] 3GPP, "Technical Specification Group Radio Access Network; Evolved Universal Terrestrial Radio Access (E-UTRA); Further advancements for E-UTRA physical layer aspects," *TR 36.814*, *Release 9*, pp. 94-96, Mar. 2017.
[29] J. Li, Q. L. Dong and M. Liao, "Study on the Scenarios and Future Development of URLLC," *Mob. Commun.*, vol. 44, no. 2, pp. 20-24, Dec. 2020.
[30] S. Zarandi and H. Tabassum, "Delay minimization in sliced multi-cell mobile edge computing (MEC) systems," *IEEE Commun. Lett.*, vol. 25, no. 6, pp. 1964-1968, Jan. 2021.