Dynamic Admission Control and Resource Allocation for Mobile Edge Computing Enabled Small Cell Network

Jiwei Huang, *Member, IEEE*, Bofeng Lv, Yuan Wu, *Senior Member, IEEE*, Ying Chen, *Member, IEEE*, and Xuemin Shen, *Fellow, IEEE*

***Abstract*—Mobile edge computing (MEC) has recently risen as a promising paradigm to meet the increasing resource requirements of the terminal devices. Meanwhile, small cell network (SCN) with MEC has been emerging to handle the exponentially increasing data traffic and improve the network coverage, and is recognized as one key component of the next generation wireless networks. However, with the growing number of terminal devices requiring computation offloading to the edge servers, the network would be heavily congested and thus the performance would be degraded and unbalanced among multiple devices. In this paper, we propose the joint admission control and computation resource allocation in the MEC enabled SCN, and formulate it as a stochastic optimization problem. The goal is to maximize the system utility combining the throughput and fairness while bounding the queue. We decouple the original problem into three independent subproblems, which can be solved in a distributed manner without requiring the system statistical information. An admission control and computation resource allocation (ACCRA) algorithm is designed to obtain the optimal solutions of the subproblems. Theoretical analysis proves that the ACCRA algorithm can achieve the close-to-optimal system utility and reach the arbitrary tradeoff between the utility and the queue length. Experiments are conducted to validate the derived analytical results and evaluate the performance of the ACCRA algorithm.**

Manuscript received June 26, 2021; revised September 13, 2021; accepted November 28, 2021. Date of publication December 9, 2021; date of current version February 14, 2022. This work was supported in part by the National Natural Science Foundation of China under Grants 61972414, 61902029, 62072490, and 61973161, in part by the Beijing Nova Program under Grant Z201100006820082, in part by the Beijing Natural Science Foundation under Grant 4202066, in part by the Fundamental Research Funds for Central Universities under Grant 2462018YJRC040, in part by the Excellent Talents Projects of Beijing under Grant 9111923401, in part by the Scientific Research Project of Beijing Municipal Education Commission under Grant KM202011232015, in part by the FDCT-MOST Joint Fund Project under Grant 0066/2019/AMJ, in part by the Macao Science and Technology Development Fund under Grants 0060/2019/A1 and 0162/2019/A3, and in part by the Research Grant of University of Macau under Grant MYRG2018-00237-FST. The review of this article was coordinated by Dr. Yan Zhang. *(Corresponding author: Ying Chen.)*

Jiwei Huang and Bofeng Lv are with the Beijing Key Laboratory of Petroleum Data Mining, China University of Petroleum, Beijing 102249, China (e-mail: huangjw@cup.edu.cn; lvbofeng@foxmail.com).

Yuan Wu is with the State Key Laboratory of Internet of Things for Smart City and Department of Computer and Information Science, University of Macau, Macau 999078, China (e-mail: yuanwu@um.edu.mo).

Ying Chen is with the Computer School, Beijing Information Science and Technology University, Beijing 100101, China (e-mail: chenying@bistu.edu.cn).

Xuemin Shen is with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON N2L 3G1, Canada (e-mail: sshen@uwaterloo.ca).

Digital Object Identifier 10.1109/TVT.2021.3133696

***Index Terms*—MEC, small cell networks, admission control, resource allocation.**

I. Introduction

With the popularity of mobile devices, complex applications requiring heavy computation and high power consumption are emerging around us [1]. However, because of the hardware resource constraints, the computational capability as well as battery lifetime are usually limited, resulting in the impracticality of processing these computation-intensive applications on the terminal device [2]. Therefore, terminal devices tend to offload the computation tasks to cloud servers. Mobile cloud computing (MCC) is recognized as a feasible method to meet the increasing resource requirements of the terminal device. However, due to the explosively growing amount of application data, offloading all the data to the remote clouds would put a heavy burden on the core networks which are already congested nowadays. Besides, the uncertain network state and long transmission distance would lead to a large transmission delay.

To address the above issues, mobile edge computing (MEC) enables the computation processing in close proximity to terminal device. Different from MCC, MEC deploys the servers at the edge of radio access networks such as base station (BS) and wireless access point (AP) [3]–[5]. Terminal devices offload the computation tasks wirelessly, then the results would be returned directly. In this way, terminal device would get better quality of service such as lower transmission latency. Meanwhile, the traffic burden on the core networks would be greatly reduced.

In addition, to enhance the network coverage, small cell networks (SCNs) are proposed, where multiple low power small-cell base stations (SBSs, i.e., femtocell BSs, picocell BSs) are deployed within one macro cell [6], [7]. By reusing the spectrum among the small cells and the macro cell, the SCN is able to dramatically enhance the energy efficiency and spectrum efficiency. Moreover, compared with traditional networks, the configuration of SCN is more flexible. With the above advantages, MEC enabled SCN, which integrates MEC servers into the macro base station in SCN, is regarded as an important part of next generation wireless networks [8].

Several previous works have paid attention to the small cell network with MEC [9]–[11]. However, the throughput-fairness

problem in the network is largely unexplored. This paper studies the throughput and fairness problem for the MEC enabled SCN. Two important decisions (i.e., task admission control and computation resource allocation) need to be made. More specifically, 1) in each SBS, how many computation tasks should be admitted for MEC server execution; 2) how much computation resource (CPU cycles) in the MEC server should be allocated to compute the tasks from each SBS. More computation tasks admitted in the SBS would increase the system throughput, which, however, may degrade the fairness among the SBSs. Admitting too many computation tasks in the SBS with poor wireless channel quality would lead to the heavy congestion. Meanwhile, too many CPU cycles being allocated would waste lots of computing resources. Therefore, making such decisions has to carefully consider both the dynamic task arrivals and wireless network states. However, in the practical scenario, both of such two factors are highly dynamic and unpredictable in advance. As a result, it is challenging to make these decisions to adapt to these stochastic processes in the real situation.

To tackle these challenges, in this paper, we propose joint admission control and computation resource allocation in the MEC enabled SCN. A utility function is introduced to consider both the throughput and fairness. A stochastic optimization problem capturing the high dynamics of wireless channel states and the task arrivals is formulated, with the goal of maximizing the average utility under the constraint of queue length. By leveraging the stochastic optimization techniques, we separate the original problem into three independent ones. An admission control and computation resource allocation (ACCRA) algorithm is designed for finding the optimal solutions of the three subproblems in a parallel way. The asymptotic optimality of ACCRA is proven by rigorous mathematical analysis. And through adjusting the tradeoff parameter, the arbitrary *utility-queue* tradeoff can be achieved. We also carry out experiments to show ACCRA's performance.

For the remainder of this paper, Section II presents the system model. Section III proposes the ACCRA algorithm and analyzes its performance through mathematical analysis. Section IV presents the experiment results, and Section V discusses the related works. We finally conclude this paper in Section VI and discuss the future directions.

II. System Model

Consider a two-tier SCN with N SBSs denoted by $\mathcal{N} = \{1, 2, \ldots, N\}$ and a macro base station (MBS). These SBSs are overlaid by the MBS, and the MBS communicates with these SBSs through wireless links [2]. The MBS includes one MEC server providing the computing services. Through the connected SBS, the MEC server processes the computation tasks offloaded from the mobile devices. The Orthogonal Frequency-Division Multiple Access (OFDMA) is used, and the channel allocated to each SBS is orthogonal to the others [6]. Motivated by [12], [13], this paper studies a discrete time slotted system and each slot length is τ. In every slot t, the MBS can obtain the task arrival information of each SBS as well as the channel state information (CSI), and make the admission control and computation resource allocation decisions. Table I shows the notations and their definitions in our system model.

TABLE I
Notations and Definitions

Notion	Definition
$\mathcal{N}$	SBSs set
p_i	i-th SBS's transmission power
$h_i(t)$	Wireless channel gain of SBS i
B	Bandwidth of the wireless channel
σ^2	Noise power spectral density
$r_i(t)$	The maximum task amount which could be offloaded by SBS i
$A_i(t)$	Arrived task amount at the i-th SBS
$d_i(t)$	Admitted task amount in SBS i
$b_i(t)$	Amount of the i-th SBS's computation tasks which could be computed in the MEC sever
$F(t)$	Computing capacity of the MEC server
$f_i(t)$	CPU cycles number allocated to SBS i
ϕ_i	CPU cycles number required for computing 1 bit task
$S_i(t)$	Queue length of SBS i's task buffer
Φ_i	Utility function of SBS i

A. Task Model

During slot t, the task arrival rate at the i-th SBS is expressed as $A_i(t)$. And there exists an upper bound A_i^{max} on $A_i(t)$. Since the radio and computation resources of the system are both limited, each SBS may only admit part of the arriving tasks to keep the system stable. For those denied tasks, the user can resend them to the cloud server. For SBS i, $d_i(t)$ denotes the admitted tasks amount. Thus,

$$0 \leq d_i(t) \leq A_i(t), \quad \forall i \in \mathcal{N}. \tag{1}$$

At slot t, each SBS would upload part of the tasks to the MBS for processing. Let p_i denote the corresponding transmission power, and $h_i(t)$ is the channel gain between SBS i and MBS. For SBS i, the maximum amount of offloadable computation tasks is (2),

$$r_i(t) = B\tau \log_2 \left(1 + \frac{p_i h_i(t)}{B\sigma^2}\right), \quad \forall i \in \mathcal{N}. \tag{2}$$

Here, B is the wireless channel's bandwidth. σ^2 is the noise power spectral density.

B. Computing Model

In every slot t, let $F(t)$ express the MEC server's computing capacity (CPU cycles), and $f_i(t)$ denote the allocated CPU cycles to SBS i. Then, the amount of SBS i's computation tasks which could be processed is

$$b_i(t) = \frac{f_i(t)}{\phi_i}, \tag{3}$$

where ϕ_i is the required CPU cycles number for computing 1 b data. Without loss of generality, ϕ_i may be different among the N SBSs.

Since the MEC server's computing ability is limited, we have

$$\sum_{i=1}^{N} f_i(t) \leq F(t), \tag{4}$$

which means that the allocated CPU cycles should be upper bounded by the server capacity. We focus on computationally intensive tasks. The processing complexity of computationally intensive tasks can be considered to be proportional to the size of the input task.

C. Task Queueing Model

For SBS i, there is a buffer storing the admitted but not yet transmitted tasks, and $S_i(t)$ is the length. According to the current buffer's queue length, the queue length in the next slot is

$$S_i(t+1) = \max\{S_i(t) - r_i(t), 0\} + d_i(t). \tag{5}$$

In addition, the MEC server also maintains a task buffer for each SBS i. Let $M_i(t)$ denote the queue length of task buffer on the MBS server for requests from SBS i at the beginning of slot t. Then,

$$M_i(t+1) = \max\{M_i(t) - b_i(t), 0\} + \min\{S_i(t), r_i(t)\}. \tag{6}$$

In (6), $\min\{S_i(t), r_i(t)\}$ is the amount of SBS i's input computation tasks at the MEC server. It is because that for each SBS, the amount of offloaded tasks can not be larger than what it has.

In this paper, inspired by [14], [15], we focus on the long term average system performance rather than the instantaneous performance in each slot. Then, the averages of $S_i(t)$ and $M_i(t)$ are given by

$$\bar{S}_i = \lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} \mathbf{E}\{S_i(t)\}, \tag{7}$$

$$\bar{M}_i = \lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} \mathbf{E}\{M_i(t)\}. \tag{8}$$

D. Problem Formulation

Here, we focus on the queue length as the performance metric, and the constraint is to bound the task buffer's queue shown as,

$$\bar{S}_i \le \zeta, \ \ \exists\, \zeta \in \mathbb{R}^+$$
$$\bar{M}_i \le \xi, \ \ \exists\, \xi \in \mathbb{R}^+. \tag{9}$$

We focus on the throughput and fairness in SCN with MEC, and targets at maximizing the utility. Inspired by the previous works [16], [17], each SBS i's utility function is

$$\Phi_i(\alpha_i, \bar{d}_i) = \begin{cases} (1-\alpha_i)^{-1}(\bar{d}_i)^{1-\alpha_i}, & \alpha_i \neq 1 \\ \log(\bar{d}_i), & \text{otherwise}, \end{cases} \tag{10}$$

where $\bar{d}_i = \lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} \mathbf{E}\{d_i(t)\}$. The utility function is concave and strictly nondecreasing. $\alpha_i \ge 0$ is the fairness parameter. When the value of α_i is larger, it means that fairness is put more weight than throughput.

Let $d_i(t)$ be the admission control variables of each SBS i, and $f_i(t)$ be the allocated computation resource variables for each SBS. Then, the optimization problem can be formulated as

$$\mathbf{P1}: \max_{\mathbf{d}(t),\mathbf{f}(t)} \sum_{i\in\mathcal{N}} \Phi_i(\alpha_i, \bar{d}_i), \tag{11}$$

subject to constraints (1), (4) and (9).

Remark: As the wireless channel state and the task arrival are dynamic as well as stochastic, **P1** is a stochastic optimization problem. It is considerably challenging to solve this problem online due to the unpredictability of these stochastic processes. In addition, the utility and the queue length are conflicting metrics. The utility function increases with the increase of the admission control variable. Thus, the system utility can be improved by admitting more tasks. However, as the radio and computation resources are limited, larger amount of admitted data would lead to the longer queue length. In some cases, admitting too many computation tasks may cause the system unstable. Therefore, designing an online strategy which can maximize the utility with bounded queue is critical and challenging.

III. Admission Control and Resource Allocation Algorithm

In this section, we design an admission control and computation resource allocation algorithm called ACCRA to solve the problem. It is obvious that the maximization optimization problem not only depends on the current task arrival and channel condition information, but also relies on their future statistical information over the time slots. However, the task arrival and channel state are usually unknown and unpredicted in prior. The lack of future statistical information would make the problem solving face great challenges. To address this issue, we decompose **P1** into three independent and deterministic optimization subproblems in every slot. With stochastic optimization techniques, ACCRA can be designed to concurrently solve these subproblems. In the end, we also analyze the asymptotic optimality of ACCRA.

A. Problem Transformation

Notably, the utility function defined in (10) is nonlinear and concave. Following the Lyapunov optimization framework, **P1** can be transformed to the following optimization problem with an *auxiliary variable* $e_i(t)$.

$$\mathbf{P2}: \max_{\mathbf{e}(t),\mathbf{d}(t),\mathbf{f}(t)} \sum_{i\in\mathcal{N}} \overline{\Phi}_i(\alpha_i, e_i(t)), \tag{12}$$

$$s.t.\ (1), (4), (9),$$

$$\bar{e}_i \le \bar{d}_i, \forall i \in \mathcal{N}, \tag{13}$$

$$0 \le e_i(t) \le A_i^{max}, \forall i \in \mathcal{N}. \tag{14}$$

where $\bar{e}_i = \lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} \mathbf{E}\{e_i(t)\}$. Because of the nondecreasing and concave properties of the utility function, it is easy to prove that **P1**'s optimal solution is equal to **P2**'s by the Jensen's inequality [18].

By defining a *virtual queue* $G_i(t)$, we reform the constraint (13) as a queue stability problem to improve the tractability of **P2**. Specifically,

$$G_i(t+1) = \max\{G_i(t) + e_i(t) - d_i(t), 0\}. \tag{15}$$

Lemma 1: For any $i \in \mathcal{N}$, if $\lim_{t\to\infty} \mathbf{E}\{G_i(t)\}/t = 0$, i.e., $G_i(t)$ is stable, the inequality constraint $\bar{e}_i \le \bar{d}_i$ would be established.

Proof: Recall that

$$G_i(t+1) = \max\{G_i(t) + e_i(t) - d_i(t), 0\} \geq G_i(t) + e_i(t) - d_i(t). \quad (16)$$

Summing (16) over the slots and dividing by t, (17) can be obtained.

$$\frac{G_i(t) - G_i(0)}{t} \geq \frac{1}{t}\sum_{\iota=0}^{t-1} e_i(\iota) - \frac{1}{t}\sum_{\iota=0}^{t-1} d_i(\iota). \quad (17)$$

For generality, we assume that $G_i(0) = 0$. Taking expectations on (17) and letting $t \to \infty$,

$$\lim_{t\to\infty} \frac{\mathbf{E}\{G_i(t)\}}{t} + \bar{d}_i \geq \bar{e}_i. \quad (18)$$

It is clear that when $G_i(t)$ is stable, i.e., $\lim_{t\to\infty} \mathbf{E}\{G_i(t)\}/t = 0$, the inequality constraint (13) would be satisfied. ■

$\mathbf{\Theta}(t) = [\mathbf{S}(t), \mathbf{M}(t), \mathbf{G}(t)]$ stands for the queue lengths' vector in the system. Then, define

$$L(\mathbf{\Theta}(t)) = \frac{1}{2}\sum_{i\in\mathcal{N}}[S_i^2(t) + M_i^2(t) + G_i^2(t)]. \quad (19)$$

It denotes the queue congestion state in the system. To keep each queue length at a small value and ensure the queue stability, we further present the *drift* function as

$$\Delta(\mathbf{\Theta}(t)) = \mathbf{E}\{L(\mathbf{\Theta}(t+1)) - L(\mathbf{\Theta}(t))|\mathbf{\Theta}(t)\}. \quad (20)$$

Then, we jointly consider the queue stability and system utility, and then the *drift-minus-utility* is given by

$$\Delta(\mathbf{\Theta}(t)) - V\mathbf{E}\{\sum_{i\in\mathcal{N}} \Phi_i(\alpha_i, e_i(t))|\mathbf{\Theta}(t)\}, \quad (21)$$

where $V > 0$ is a coefficient to achieve the *stability-utility* tradeoff. The *drift-minus-utility*'s supremum bound is presented in Theorem 1.

Theorem 1: If $r_i(t)$'s and $F(t)$'s upper bounds r_i^{max} and F^{max} exist, no matter what the value of $\mathbf{\Theta}(t)$ is, for any feasible policies, the following inequality would be satisfied,

$$\begin{aligned}\Delta(\mathbf{\Theta}(t)) - V\mathbf{E}\left\{\sum_{i\in\mathcal{N}} \Phi_i(t)|\mathbf{\Theta}(t)\right\} &\leq C - V\mathbf{E}\left\{\sum_{i\in\mathcal{N}} \Phi_i(t)|\mathbf{\Theta}(t)\right\} \\ &+ \sum_{i\in\mathcal{N}} S_i(t)\mathbf{E}\{d_i(t) - r_i(t)|\mathbf{\Theta}(t)\} \\ &+ \sum_{i\in\mathcal{N}} M_i(t)\mathbf{E}\{r_i(t) - b_i(t)|\mathbf{\Theta}(t)\} \\ &+ \sum_{i\in\mathcal{N}} G_i(t)\mathbf{E}\{e_i(t) - d_i(t)|\mathbf{\Theta}(t)\}. \end{aligned} \quad (22)$$

Here, $C = \frac{1}{2}\sum_{i\in\mathcal{N}}[2(r_i^{max})^2 + 3(A_i^{max})^2 + (\frac{F^{max}}{\phi_i})^2]$ is a constant.

Proof: According to the facts that $(max\{a-b,0\}+c)^2 - a^2 \leq b^2 + c^2 + 2a(c-b)$ and $min\{a,b\} \leq a$, and squaring (5), (6) and (15), it can yield that

$$S_i^2(t+1) - S_i^2(t) \leq r_i^2(t) + d_i^2(t) + 2S_i(t)[d_i(t) - r_i(t)], \quad (23)$$

$$M_i^2(t+1) - M_i^2(t) \leq b_i^2(t) + r_i^2(t) + 2M_i(t)[r_i(t) - b_i(t)], \quad (24)$$

$$H_i^2(t+1) - H_i^2(t) \leq e_i^2(t) + d_i^2(t) + 2H_i(t)[e_i(t) - d_i(t)]. \quad (25)$$

Based on (23), (24) and (25), we have

$$\begin{aligned}\Delta(\mathbf{\Theta}(t)) \leq \frac{1}{2}\mathbf{E}\left\{\sum_{i\in\mathcal{N}}(2r_i^2(t) + 2d_i^2(t) + b_i^2(t) + e_i^2(t))|\mathbf{\Theta}(t)\right\} \\ + \sum_{i\in\mathcal{N}} S_i(t)\mathbf{E}\{d_i(t) - r_i(t)|\mathbf{\Theta}(t)\} \\ + \sum_{i\in\mathcal{N}} M_i(t)\mathbf{E}\{r_i(t) - b_i(t)|\mathbf{\Theta}(t)\} \\ + \sum_{i\in\mathcal{N}} G_i(t)\mathbf{E}\{e_i(t) - d_i(t)|\mathbf{\Theta}(t)\}. \end{aligned} \quad (26)$$

Note that it holds $r_i(t) \leq r_i^{max}$, $d_i(t) \leq A_i^{max}$, $b_i(t) \leq F^{max}/\phi_i$ and $e_i(t) \leq A_i^{max}$. We can obtain (27).

$$\begin{aligned}&\mathbf{E}\left\{\sum_{i\in\mathcal{N}}(2r_i^2(t) + 2d_i^2(t) + b_i^2(t) + e_i^2(t))|\mathbf{\Theta}(t)\right\} \\ &\leq \sum_{i\in\mathcal{N}}\left[2(r_i^{max})^2 + 3(A_i^{max})^2 + \left(\frac{F^{max}}{\phi_i}\right)^2\right]. \end{aligned} \quad (27)$$

Then, putting (27) into (26) and adding $-V\mathbf{E}\{\sum_{i\in\mathcal{N}} \Phi_i(t)|\mathbf{\Theta}(t)\}$ obtains (22). ■

B. Online Optimal Algorithm Design

In this subsection, we design an online optimal algorithm to get the minimization of *drift-minus-utility*'s supremum bound given in Theorem 1. The minimization of supremum bound can be decomposed into three independent subproblems. As a result, the optimal solutions to these subproblems can be obtained in a decentralized way.

Notably, in every slot, C, $\mathbf{\Theta}(t)$ and $r_i(t)$ are constant. To be more specific, parameter C is the constant defined in Theorem 1. Besides, for each given time slot t, $\mathbf{\Theta}(t)$ and $r_i(t)$ can be obtained and known at the beginning of each slot t; thus, they are also constants of the decision variables. Then, the supremum bound minimization problem can be equivalently reformulated by eliminating the constant term,

$$\begin{aligned}\mathbf{P3}: \min_{\mathbf{e}(t),\mathbf{d}(t),\mathbf{f}(t)} &\sum_{i\in\mathcal{N}}[G_i(t)e_i(t) - V\Phi_i(\alpha_i, e_i(t))] \\ &+ \sum_{i\in\mathcal{N}} d_i(t)[S_i(t) - G_i(t)] - \sum_{i\in\mathcal{N}} M_i(t)b_i(t), \\ &s.t.\ (1),\ (4),\ (14). \end{aligned} \quad (28)$$

Note that in the objective and constraints of **P3**, the decision variables $\mathbf{e}(t), \mathbf{d}(t), \mathbf{f}(t)$ are all decoupled. Then, **P3** can be decoupled to three independent subproblems. Specifically, the three subproblems are: auxiliary variable selection, admission

control decision and computation resource allocation. Next, the optimal solutions to these subproblems are given separately.

1) Auxiliary Variable Optimization: The auxiliary variables among the N SBSs are independent, where the optimal auxiliary variable of each SBS can be computed concurrently. For each SBS, the following optimization problem can be formulated with the decision variable $e_i(t)$.

$$\min_{e_i(t)} \; G_i(t)e_i(t) - V\Phi_i(\alpha_i, e_i(t)),$$
$$s.t. \;\; 0 \le e_i(t) \le A_i^{max}, \forall i \in \mathcal{N}. \tag{29}$$

Since $G_i(t) - V\Phi_i(\alpha_i, e_i(t))$ is differential, we can obtain its first-order derivation $\psi(\alpha_i, e_i(t))$ as the following

- If $\alpha_i = 0$, $\psi(\alpha_i, e_i(t)) = G_i(t) - V$;
- If $\alpha_i = 1$, $\psi(\alpha_i, e_i(t)) = G_i(t) - \frac{V}{e_i(t)}$;
- If $\alpha_i \neq 0, 1$, $\psi(\alpha_i, e_i(t)) = G_i(t) - Ve_i(t)^{-\alpha_i}$.

Then, according to the above results, the optimal auxiliary variable $e_i^*(t)$ can be obtained as follows

- If $\alpha_i = 0$,

$$e_i^*(t) = \begin{cases} A_i^{max}, & G_i(t) < V \\ 0, & \text{otherwise}; \end{cases} \tag{30}$$

- If $\alpha_i = 1$,

$$e_i^*(t) = \begin{cases} A_i^{max}, & G_i(t) < \frac{V}{A_i^{max}} \\ \frac{V}{G_i(t)}, & otherwise; \end{cases} \tag{31}$$

- If $\alpha_i \neq 0, 1$,

$$e_i^*(t) = \begin{cases} A_i^{max}, & G_i(t) < \frac{V}{(A_i^{max})^{\alpha_i}} \\ (\frac{V}{G_i(t)})^{\frac{1}{\alpha_i}}, & otherwise. \end{cases} \tag{32}$$

2) Admission Control Decision: Similarly, the admission control decisions among the N SBSs are also independent. The optimal admission decision of each SBS can be computed in a distributed way. For each SBS, considering the second term in the objective of **P3** and constraint (1), then, (33) can be formulated with the decision variable $d_i(t)$.

$$\min_{d_i(t)} \; d_i(t)[S_i(t) - G_i(t)],$$
$$s.t. \;\; 0 \le d_i(t) \le A_i(t), \;\; \forall i \in \mathcal{N}. \tag{33}$$

(33) is a simple linear programming problem. For each SBS, the optimal value of $d_i(t)$ is

$$d_i^*(t) = \begin{cases} A_i(t), & S_i(t) - G_i(t) < 0 \\ 0, & \text{otherwise}. \end{cases} \tag{34}$$

3) Computation Resource Allocation: For the second last term in **P3**'s objective and constraint (4), the following optimization problem can be formulated with the decision variable $\mathbf{f}(t)$.

$$\min_{\mathbf{f}(t)} \; \sum_{i \in \mathcal{N}} -\frac{M_i(t)}{\phi_i} f_i(t),$$
$$s.t. \; \sum_{i=1}^{N} f_i(t) \le F(t). \tag{35}$$

(35) is a min-weight problem, and the computation resource allocation $f_i(t)$ is weighted by the value of $-\frac{M_i(t)}{\phi_i}$. Thus, the optimal decision $f_i^*(t)$ is

$$f_i^*(t) = \begin{cases} F_i(t), & i = i^* \\ 0, & otherwise, \end{cases} \tag{36}$$

where $i^* = \text{argmin}_{i \in \{1,2,\ldots,\mathcal{N}\}} -\frac{M_i(t)}{\phi_i}$, which represents the minimum value of $-\frac{M_i(t)}{\phi_i}$ among all the SBSs.

After the optimal auxiliary variables $\mathbf{e}^*(t)$ are determined, the virtual queue $\mathbf{G}(t)$ can be updated by (15). Likewise, according to the optimal admission control $\mathbf{d}^*(t)$ and computation resource allocation $\mathbf{f}^*(t)$, we can update the actual queues $\mathbf{S}(t)$ and $\mathbf{M}(t)$ according to (5) and (6), respectively. The details of ACCRA are presented in Algorithm 1.

Algorithm 1: Admission Control and Computation Resource Allocation (ACCRA).

```
Get the current lengths of actual queues S(t), M(t) and virtual queue G(t).
for all i ∈ N do
  if α_i = 0 then
    Determine the optimal auxiliary variables e_i(t) according to (30).
  else if α_i = 1 then
    Determine the optimal auxiliary variables e_i(t) according to (31).
  else
    Determine the optimal auxiliary variables e_i(t) according to (32).
  end if
  Make the optimal admission control decision d_i(t) according to (34).
end for
for all i ∈ N do
  Find i* = arg min_{i∈{1,2,...,N}} −M_i(t)/φ_i.
end for
Allocate the computation resource according to (36).
```

C. Algorithm Analysis

This subsection theoretically analyzes the performance of ACCRA and proves that ACCRA can approach the maximal system utility by setting a sufficiently large value of tradeoff parameter V.

Let $\mathbf{\Phi}(t) = \sum_{i \in \mathcal{N}} \Phi_i(t)$ represent the total system utility. We first present Lemma 2 to help prove the asymptotic optimality of ACCRA.

Lemma 2: If **P2** is feasible, there exist a randomized strategy π and $\varepsilon > 0$ satisfying the following,

$$\begin{aligned} \mathbf{E}\{\mathbf{\Phi}^{\pi}(t)\} &= \mathbf{\Phi}^{OPT}(\varepsilon), \\ \mathbf{E}\{d_i^{\pi}(t)\} &\le \mathbf{E}\{r_i(t)\} - \varepsilon, \\ \mathbf{E}\{r_i(t)\} &\le \mathbf{E}\{b_i^{\pi}(t)\} - \varepsilon, \\ \mathbf{E}\{e_i^{\pi}(t)\} &\le \mathbf{E}\{d_i^{\pi}(t)\} - \varepsilon. \end{aligned} \tag{37}$$

where $\mathbf{\Phi}^{OPT}(\varepsilon)$ represents the optimal average system utility with ε.

Proof: By applying Caratheodory's theorem, one can prove the Lemma 2 [18]. The details of the proof are omitted for simplicity. ■

Let $\mathbf{\Phi}^{ACCRA}$ denote the average system utility obtained by ACCRA, and $\mathbf{\Phi}^{GAP}$ denote the gap between $\mathbf{\Phi}^{OPT}$ and $\mathbf{\Phi}^{ACCRA}$. Next, we present Theorem 2 which shows the upper bound of $\mathbf{\Phi}^{GAP}$.

Theorem 2: For any V, $\mathbf{\Phi}^{GAP}$ would satisfy the following,

$$\mathbf{\Phi}^{GAP} \leq \frac{C}{V}, \tag{38}$$

where C is defined in Theorem 1.

Proof: Note that ACCRA minimizes *drift-minus-utility*'s upper bound. Then,

$$\begin{aligned}\Delta(\mathbf{\Theta}(t)) - V\mathbf{E}\{\mathbf{\Phi}^{ACCRA}(t)|\mathbf{\Theta}(t)\} \leq C\\ -V\mathbf{E}\{\mathbf{\Phi}^{\pi}(t)|\mathbf{\Theta}(t)\} + \sum_{i\in\mathcal{N}} S_i(t)\mathbf{E}\{d_i^{\pi}(t) - r_i(t)|\mathbf{\Theta}(t)\}\\ + \sum_{i\in\mathcal{N}} M_i(t)\mathbf{E}\{r_i(t) - b_i^{\pi}(t)|\mathbf{\Theta}(t)\}\\ + \sum_{i\in\mathcal{N}} G_i(t)\mathbf{E}\{e_i^{\pi}(t) - d_i^{\pi}(t)|\mathbf{\Theta}(t)\}.\end{aligned} \tag{39}$$

Substituting (37) into (39), taking expectations and using iterated expectations, (40) can be obtained.

$$\begin{aligned}&\mathbf{E}\{L(\Theta(t+1))\} - \mathbf{E}\{L(\Theta(t))\} - V\mathbf{E}\{\mathbf{\Phi}^{ACCRA}(t)\}\\ &\leq C - V\mathbf{\Phi}^{OPT}(\varepsilon) - \varepsilon\mathbf{E}\left\{\sum_{i\in\mathcal{N}}[S_i(t)+M_i(t)+G_i(t)]\right\}\end{aligned} \tag{40}$$

Summing (40) over the slots, dividing by T and let $T \to \infty$, we have

$$-V\mathbf{\Phi}^{ACCRA} \leq C - V\Phi^{OPT}(\varepsilon) - \varepsilon\bar{Q}. \tag{41}$$

Then, dividing (41) by V and let $\varepsilon \to 0$, (42) can be obtained.

$$\mathbf{\Phi}^{OPT} - \mathbf{\Phi}^{ACCRA} \leq \frac{C}{V}. \tag{42}$$

Define $\bar{Q}$ to be the average length, which is expressed by,

$$\bar{Q} = \lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} \sum_{i\in\mathcal{N}} \mathbf{E}\{S_i(t) + M_i(t) + G_i(t)\}. \tag{43}$$

Then, the upper bound of $\bar{Q}$ is given by Theorem 3.

Theorem 3: For any value of V, $\bar{Q}$ would be upper bounded by

$$\bar{Q} \leq \frac{C + V\mathbf{\Phi}^{OPT} - V\mathbf{\Phi}^{OPT}(\varepsilon)}{V}. \tag{44}$$

Proof: Dividing (41) by εV, we have

$$\begin{aligned}\bar{Q} &\leq \frac{C + V\mathbf{\Phi}^{ACCRA} - V\mathbf{\Phi}^{OPT}(\varepsilon)}{\varepsilon},\\ &\leq \frac{C + V\mathbf{\Phi}^{OPT} - V\mathbf{\Phi}^{OPT}(\varepsilon)}{\varepsilon}.\end{aligned} \tag{45}$$

Remark: Theorem 2 shows that the system utility of ACCRA rises as V increases. When V becomes sufficiently large, the gap would be sufficiently small and the optimal system utility can be approximated by ACCRA. Theorem 3 demonstrates that with the rise of V, the queue length would increase. Nevertheless, the queue length always has a supremum bound. Combining Theorem 2 and Theorem 3, ACCRA can make an $[O(1/V), O(V)]$ tradeoff between the system utility and queue length.

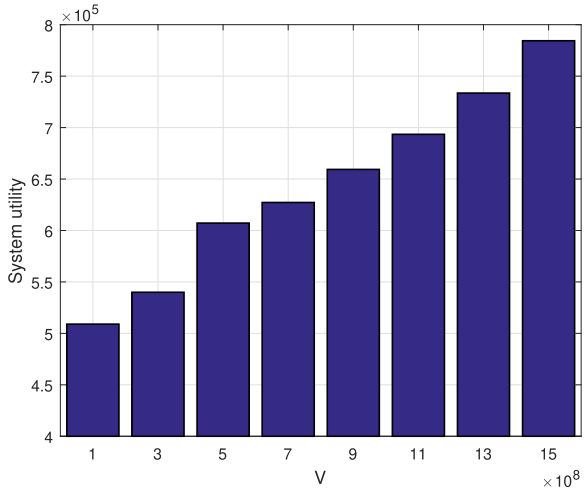

Fig. 1. System utility with different values of V.

For the first loop in the ACCRA algorithm (line 2 to line 11), the execution time of (30)-(32) and (34) is constant. The time complexity of the first loop is $O(n)$. The second loop (line 12 to 14) is to find the element with the lowest weight. The time complexity of the second loop is also $O(n)$. The 15th line of the ACCRA algorithm allocates computing resources according to equation (36). This requires traversing the queue once, and the time complexity is $O(n)$. Thus, the total time complexity of the ACCRA algorithm is $O(n)$.

IV. Evaluation

This section presents the experiments to validate ACCRA's performance. Consider an SCN with 5 SBSs and 1 MBS. For each SBS i, $p_i \sim U[50, 150]$ mW, $\alpha_i \sim U[0, 1]$ [2], $\phi_i \sim U[1000, 2000]$ cycles/bit [19] and $F(t) \sim U[10, 12]$ GHz. The task arrival rate $A_i(t)$ is uniform with the maximization value A_i^{max}. Similar to [14], we adopt the Rayleigh fading channel and $h_i(t)$ is exponentially distributed with unit mean. In addition, for each SBS, $B = 5$ MHz and $\sigma^2 = 10^{-7}$. The slot length is $\tau = 1$ s.

A. Parameter Analysis

1) Tradeoff Parameter: Fig. 1 shows the impact of tradeoff parameter V on the average utility. It can be found that the utility rises as V increases, which agrees with Theorem 2 that a larger V would improve the utility. Actually, setting a larger V would emphasize more on the system utility, and ACCRA would dynamically adjust the admission control variables to increase the utility. Fig. 2 shows the average queue length when V is changed. It can be found that the queue length would be larger when V is increased, agreeing with (44) in Theorem 3. Combining the two figures, we can find that by setting different V, the arbitrary utility-queue tradeoff can be obtained.

2) Task Arrival Rate: Fig. 3 shows the impact of the arrival rate on the utility, and the rate is set as $\gamma \cdot A_i(t)$, where $\gamma =$

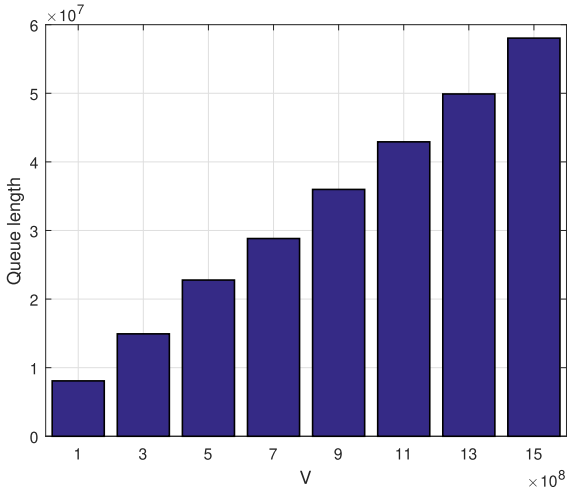

Fig. 2. Queue length with different values of V.

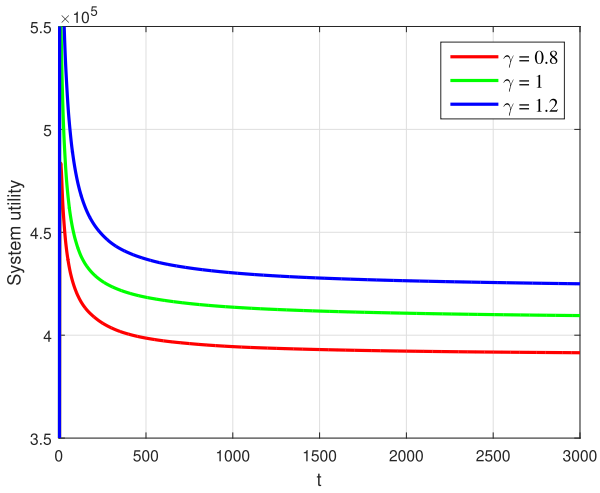

Fig. 3. System utility with different task arrival rates.

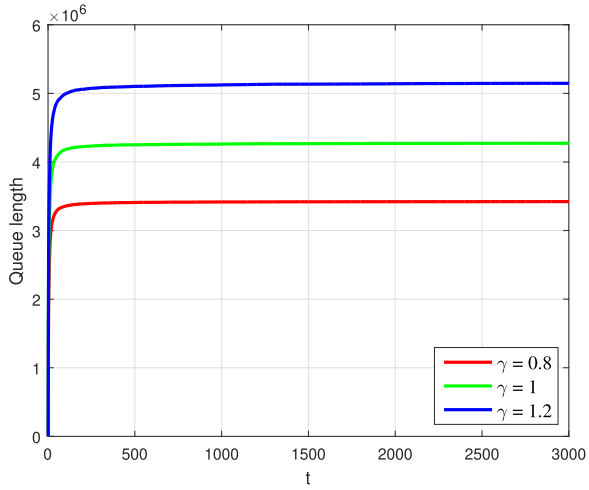

Fig. 4. Queue length with different task arrival rates.

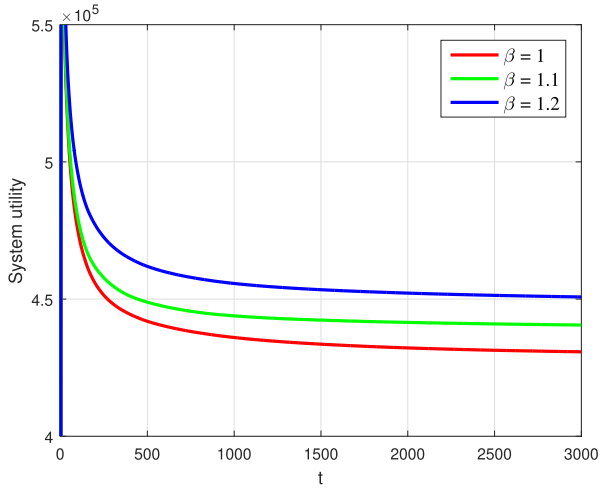

Fig. 5. System utility with different MEC computing capacities.

0.8, 1 and 1.2, respectively. It is observed that the utility is higher with larger arrival rate. This is because more computation tasks could be admitted by ACCRA when the arrival rate is increased. Fig. 4 shows ACCRA's queue length when the arrival rate is changed. It is observed that the length is larger when the arrival rate rises. Nevertheless, it is seen that the queue is always stabilized by the ACCRA algorithm. From the two figures, it can be concluded that with different arrival rates, ACCRA can adjust the decision variables, and maintain the queue stability while achieving higher utility.

3) MEC Computing Capacity: Fig. 5 and Fig. 6 show the impact of the MEC computing capacity on the utility and the queue. The capacity is set $\beta \cdot F(t)$, where $\beta = 1$, 1.1 and 1.2, respectively. Fig. 5 illustrates that the utility is higher with larger computing capacity. Actually, the processing ability would be enhanced and more computation tasks could be admitted, improving the utility. From Fig. 6, it is observed that the queue length is reduced with the increased computing capacity. This is because the processing ability would be enhanced and reducing the queue length. According to the two figures, it can be illustrated that the ACCRA's utility would be higher with bounded queue length when the computing capacity is increased.

B. Comparison Experiment

This subsection further evaluates the performance of ACCRA by comparing it with two benchmark algorithms [20], [21].

- Round Robin Greedy (RRG): Each SBS's tasks are admitted in turn in each slot, and the algorithm admits the computation tasks as many as possible;
- Fair Greedy (FG): In each slot t, each SBS is prioritized by $1/\sum_{\iota=0}^{t-1} d_i(t)$, and only the computation tasks from the SBS with the highest priority are admitted. In addition, the tasks are admitted as many as possible.

Fig. 7 is the average utility with the three different algorithms. We can observe that among the 3 algorithms, ACCRA's utility is the highest. It is because that ACCRA can dynamically adjust the admission control and computing resource allocation decisions based on the current queue lengths, task arrivals as well as the wireless channel states. However, both RRG and FG make the admission control decisions according to only

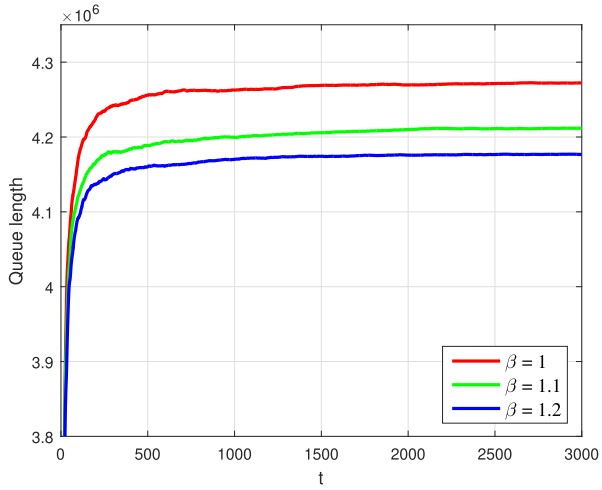

Fig. 6. Queue length with different MEC computing capacities.

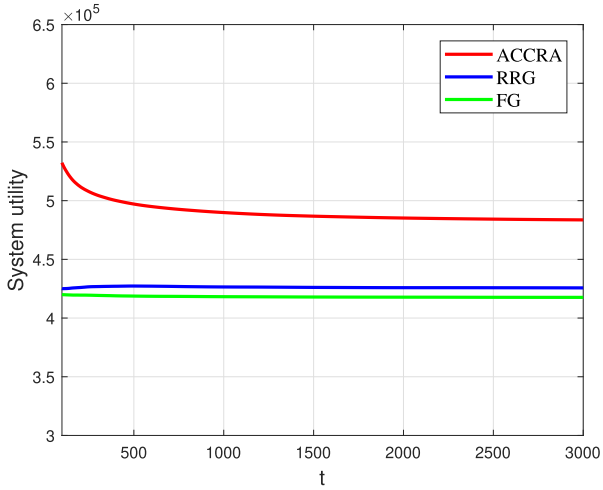

Fig. 7. System utility with the three different algorithms.

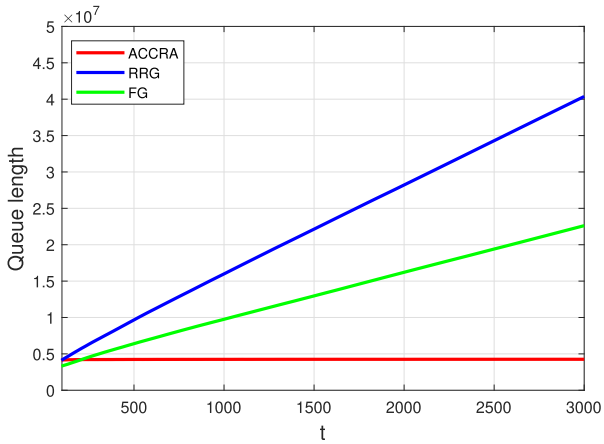

Fig. 8. Queue length with the three different algorithms.

the current task arrivals. Therefore, ACCRA can achieve the better system utility compared with RRG and FG. In Fig. 8, to further show the performance of ACCRA, we also illustrate the 3 different algorithms' queue lengths. Our ACCRA algorithm has the lowest queue length, and it is able to stabilize the queueing state in a fast speed. And the queue lengths of RRG and FG both continually increase as time goes by. The reason is that for the SBS with long queue length, ACCRA would reduce its amount of admitted tasks to keep the queue stable. Fig. 7 and Fig. 8 show ACCRA's superiorities on stabilizing the queue length at a low level while increasing the utility.

V. Related Work

Mao *et al.* [14] studied a multi-user MEC system with the objective of energy consumption minimization by optimization of communication and computing resources. They integrated energy harvesting technologies into the offloading optimization problem in MEC in [15]. Ren *et al.* [22] jointly considered the radio and computing resources allocation to minimize the latency. A resource allocation algorithm was designed to obtain the optimal solutions of this latency minimization problem. The above works all focused on the single tier homogeneous cellular network with MEC, and mainly paid attention to the energy consumption and delay.

Recently, there have been some works integrating MEC into small cell network. Yang *et al.* [2] investigated the task offloading problem in small cell network with MEC, and an offloading optimization model was built aiming at energy consumption minimization with the constraints of computing capacity and delay. An offloading scheme based on the artificial fish swarm algorithm was designed. Zhang *et al.* [6] studied the fronthaul and backhaul links selection problem, and minimized the offloading energy while guaranteeing that the delay did not exceed a threshold. The computation offloading problem was studied targeting at energy consumption minimization in [8]. An offloading and communication resource allocation method was put forward to achieve the minimal energy consumption.

The above works all focused on the energy consumption minimization problem. Different from them, Zhang *et al.* [9] studied the balance between delay and energy in the MEC powered small cell network, and considered both the single cell and multiple cells scenarios. They put forward an iterative search method to find the optimum strategy. Besides, the cost minimization problem which combined the energy consumption and delay was also investigated in some works. The task offloading and interference management was studied in [10], and the joint transmission power, communication and computing resources allocation was studied in [23]. A task offloading approach for edge computing in the scenario of ultradense 5G cells was proposed in [24]. A game-theoretic greedy scheme was designed for solving the overhead minimization problem while satisfying the given wireless channel constraints.

In summary, most existing works about MEC powered small cell network mainly focused on the energy efficient offloading or energy-delay tradeoff. However, the throughput-fairness problem was given less insights. Therefore, the throughput and fairness are jointly considered in this paper with maximization of utility for the small cell network with MEC.

VI. Conclusion

In this paper, we jointly study the task admission control and computing resource allocation for the MEC-enabled SCN.

A utility function which jointly considers the throughput and fairness is adopted, and a dynamic optimization scheme called ACCRA is designed for maximizing the utility function while keeping the buffer from congested. Theoretical analysis prove that ACCRA is able to balance arbitrary trade-off between the system utility and the queue stability. Simulation experiments are also conducted and the efficacy of ACCRA is validated by empirical results. In the future, the user mobility and privacy issues will be further considered in the design and optimization of the admission control and resource allocation schemes for MEC.

REFERENCES

[1] X. Wang, Y. Zhang, V. C. M. Leung, N. Guizani, and T. Jiang, "D2D Big Data: Content deliveries over wireless device-to-device sharing in large-scale mobile networks," *IEEE Wireless Commun.*, vol. 25, no. 1, pp. 32–38, Feb. 2018.

[2] L. Yang, H. Zhang, M. Li, J. Guo, and H. Ji, "Mobile edge computing empowered energy efficient task offloading in 5G," *IEEE Trans. Veh. Technol.*, vol. 67, no. 7, pp. 6398–6409, Jul. 2018.

[3] J. Huang, S. Li, and Y. Chen, "Revenue-optimal task scheduling and resource management for IoT batch jobs in mobile edge computing," *Peer-to-Peer Netw. Appl.*, vol. 13, no. 5, pp. 1776–1787, 2020.

[4] K. Zhang, J. Cao, and Y. Zhang, "Adaptive digital twin and multi-agent deep reinforcement learning for vehicular edge computing and networks," *IEEE Trans. Ind. Inform.*, vol. 18, no. 2, pp. 1405–1413, Feb. 2022, doi: 10.1109/TII.2021.3088407.

[5] X. Zhang, H. Huang, H. Yin, D. O. Wu, G. Min, and Z. Ma, "Resource provisioning in the edge for IoT applications with multilevel services," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4262–4271, Jun. 2019.

[6] H. Zhang, J. Guo, L. Yang, X. Li, and H. Ji, "Computation offloading considering fronthaul and backhaul in small-cell networks integrated with MEC," in *Proc. IEEE Conf. Comput. Commun. Workshops*, 2017, pp. 115–120.

[7] X. Huang, S. Leng, S. Maharjan, and Y. Zhang, "Multi-agent deep reinforcement learning for computation offloading and interference coordination in small cell networks," *IEEE Trans. Veh. Technol.*, vol. 70, no. 9, pp. 9282–9293, Sep. 2021.

[8] K. Zhang *et al.*, "Energy-efficient offloading for mobile edge computing in 5G heterogeneous networks," *IEEE Access*, vol. 4, pp. 5896–5907, 2016.

[9] J. Zhang *et al.*, "Energy-latency tradeoff for energy-aware offloading in mobile edge computing networks," *IEEE Internet Things J.*, vol. 5, no. 4, pp. 2633–2645, Aug. 2018.

[10] C. Wang, F. R. Yu, C. Liang, Q. Chen, and L. Tang, "Joint computation offloading and interference management in wireless cellular networks with mobile edge computing," *IEEE Trans. Veh. Technol.*, vol. 66, no. 8, pp. 7432–7445, Aug. 2017.

[11] Y. Chen, N. Zhang, Y. Zhang, X. Chen, W. Wu, and X. S. Shen, "TOFFEE: Task offloading and frequency scaling for energy efficiency of mobile devices in mobile edge computing," *IEEE Trans. Cloud Comput.*, vol. 9, no. 4, pp. 1634–1644, Oct.–Dec. 2021.

[12] Y. Chen, F. Zhao, Y. Lu, and X. Chen, "Dynamic task offloading for mobile edge computing with hybrid energy supply," *Tsinghua Sci. Technol.*, to be published, doi: 10.26599/TST.2021.9010050.

[13] J. Huang, C. Zhang, and J. Zhang, "A multi-queue approach of energy efficient task scheduling for sensor hubs," *Chin. J. Electron.*, vol. 29, no. 2, pp. 242–247, 2020.

[14] Y. Mao, J. Zhang, S. H. Song, and K. B. Letaief, "Stochastic joint radio and computational resource management for multi-user mobile-edge computing systems," *IEEE Trans. Wireless Commun.*, vol. 16, no. 9, pp. 5994–6009, Sep. 2017.

[15] 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.

[16] L. P. Qian, Y. Wu, H. Zhou, and X. Shen, "Joint uplink base station association and power control for small-cell networks with non-orthogonal multiple access," *IEEE Trans. Wireless Commun.*, vol. 16, no. 9, pp. 5567–5582, Sep. 2017.

[17] S. Tao, L. Gu, D. Zeng, H. Jin, and K. Hu, "Fairness-aware dynamic rate control and flow scheduling for network function virtualization," in *Proc. IEEE/ACM 25th Int. Symp. Qual. Service*, 2017, pp. 1–6.

[18] M. J. Neely, *Stochastic Network Optimization With Application to Communication and Queueing Systems*. San Rafael, CA, USA: Morgan & Claypool, 2010, pp. 1–198.

[19] X. Lyu, C. Ren, W. Ni, H. Tian, and R. P. Liu, "Distributed optimization of collaborative regions in large-scale inhomogeneous fog computing," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 3, pp. 574–586, Mar. 2018.

[20] X. Lyu *et al.*, "Optimal schedule of mobile edge computing for Internet of Things using partial information," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 11, pp. 2606–2615, Nov. 2017.

[21] D. Zhang *et al.*, "Resource allocation for green cloud radio access networks with hybrid energy supplies," *IEEE Trans. Veh. Technol.*, vol. 67, no. 2, pp. 1684–1697, Feb. 2018.

[22] J. Ren, G. Yu, Y. Cai, and Y. He, "Latency optimization for resource allocation in mobile-edge computation offloading," *IEEE Trans. Wireless Commun.*, vol. 17, no. 8, pp. 5506–5519, Aug. 2018.

[23] J. Zhang, W. Xia, F. Yan, and L. Shen, "Joint computation offloading and resource allocation optimization in heterogeneous networks with mobile edge computing," *IEEE Access*, vol. 6, pp. 19324–19337, 2018.

[24] H. Guo, J. Liu, J. Zhang, W. Sun, and N. Kato, "Mobile-edge computation offloading for ultradense IoT networks," *IEEE Internet Things J.*, vol. 5, no. 6, pp. 4977–4988, Dec. 2018.

Jiwei Huang (Member, IEEE) received the B.Eng. and Ph.D. degrees in computer science and technology from Tsinghua University, Beijing, China, in 2009 and 2014, respectively. He was a Visiting Scholar with the Georgia Institute of Technology, Atlanta, GA, USA. He is currently a Professor and the Dean with the Department of Computer Science and Technology, China University of Petroleum, Beijing, and the Director with the Beijing Key Laboratory of Petroleum Data Mining, Beijing. He has authored or coauthored one book and more than 50 articles in international journals and conference proceedings, including the IEEE TRANSACTIONS ON SERVICES COMPUTING, IEEE TRANSACTIONS ON CLOUD COMPUTING, ACM SIGMETRICS, IEEE ICWS, and IEEE SCC. His research interests include services computing, Internet of Things, and edge computing. He is currently on the Editorial Board of the *Chinese Journal of Electronics and Scientific Programming*.

Bofeng Lv received the B.Eng. degree in computer science and technology in 2020 from the China University of Petroleum, Beijing, China, where he is currently working toward the M.Eng. degree in computer science and technology. His current research interests include edge computing, Internet of Things, and reinforcement learning.

Yuan Wu (Senior Member, IEEE) received the Ph.D. degree in electronic and computer engineering from The Hong Kong University of Science and Technology, Hong Kong, in 2010. From 2016 to 2017, he was a Visiting Scholar with the Department of Electrical and Computer Engineering, University of Waterloo, ON, Canada. He is currently an Associate Professor with the State Key Laboratory of Internet of Things for Smart City, University of Macau, Zhuhai, China, and also with the Department of Computer and Information Science, University of Macau. His research interests include resource management for wireless networks, green communications and computing, mobile edge computing, and smart grids. He was the recipient of the Best Paper Award from the IEEE International Conference on Communications in 2016, the Best Paper Award from IEEE Technical Committee on Green Communications and Computing in 2017, and the Best Paper Award from the International Wireless Communications and Mobile Computing Conference in 2021. He is currently on the Editorial Boards of IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, IEEE TRANSACTIONS ON NETWORK SCIENCE AND ENGINEERING, IEEE INTERNET OF THINGS JOURNAL, and IEEE Open Journal of the Communication Society. He was the Guest Editor of the *IEEE Communications Magazine*, IEEE NETWORK, *IEEE Wireless Communications Magazine*, IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, and *IET Communications*.

Ying Chen (Member, IEEE) received the Ph.D. degree from Tsinghua University, Beijing, China, in 2017. She is currently an Associate Professor with Computer School, Beijing Information Science and Technology University, Beijing, China. From 2016 to 2017, she was a joint Ph.D. Student with the University of Waterloo, ON, Canada . Her current research interests include Internet of Things, mobile edge computing, wireless networks and communications, and machine learning. She was the recipient of the Best Paper Award at IEEE SmartIoT 2019, 2016 Google Ph.D. Fellowship Award, and 2014 Google Anita Borg Award, respectively. She is/was the TPC Member of the IEEE HPCC, and PC Member of IEEE Cloud, CollaborateCom, IEEE CPSCom, and CSS. She is also a Reviewer of several journals, such as *IEEE Wireless Communications Magazine*, IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING, IEEE INTERNET OF THINGS JOURNAL, IEEE TRANSACTIONS ON COMPUTERS, IEEE TRANSACTIONS ON CLOUD COMPUTING, and IEEE TRANSACTIONS ON SERVICES COMPUTING.

Xuemin (Sherman) Shen (Fellow, IEEE) received the Ph.D. degree in electrical engineering from Rutgers University, New Brunswick, NJ, USA, in 1990. He is currently a University Professor with the Department of Electrical and Computer Engineering, University of Waterloo, ON, Canada. His research interests include network resource management, wireless network security, Internet of Things, 5G and beyond, and vehicular networks.

Dr. Shen was the recipient of the Canadian Award for Telecommunications Research from the Canadian Society of Information Theory (CSIT) in 2021, R.A. Fessenden Award in 2019 from IEEE, Canada, Award of Merit from the Federation of Chinese Canadian Professionals (Ontario) in 2019, James Evans Avant Garde Award in 2018 from the IEEE Vehicular Technology Society, Joseph LoCicero Award in 2015 and Education Award in 2017 from the IEEE Communications Society (ComSoc), Technical Recognition Award from Wireless Communications Technical Committee (2019) and AHSN Technical Committee (2013), Excellent Graduate Supervision Award in 2006 from the University of Waterloo, and Premier's Research Excellence Award (PREA) in 2003 from the Province of Ontario, Canada. He was the Technical Program Committee Chair/Co-Chair for IEEE GLOBECOM'16, IEEE INFOCOM'14, IEEE VTC'10 Fall, IEEE GLOBECOM'07, and the Chair for the IEEE ComSoc Technical Committee on Wireless Communications. He is the President Elect of the IEEE ComSoc. He was the Vice President for Technical and Educational Activities, Vice President for Publications, Member-at-Large on the Board of Governors, Chair of the Distinguished Lecturer Selection Committee, and a Member of IEEE Fellow Selection Committee of the ComSoc. He was the Editor-in-Chief of IEEE INTERNET OF THINGS JOURNAL, IEEE NETWORK, and *IET Communications*. He is a registered Professional Engineer of Ontario, Canada, an Engineering Institute of Canada Fellow, a Canadian Academy of Engineering Fellow, a Royal Society of Canada Fellow, a Chinese Academy of Engineering Foreign Fellow, and a Distinguished Lecturer of the IEEE Vehicular Technology Society and Communications Society.