Task Scheduling and Resource Management in MEC-Enabled Computing Networks

Jie Feng[1,2,3], Wenjing Zhang[4], Lei Liu[1,2](✉), Jianbo Du[3], Ming Xiao[5], and Qingqi Pei[1]

[1] State Key Laboratory of ISN, School of Telecommunication Engineering, Xidian University, Xi'an, Shaanxi, China
qqpei@mail.xidian.edu.cn
[2] Guangzhou Institute of Technology, Xidian University, Xi'an, China
tianjiaoliulei@163.com
[3] Shaanxi Key Laboratory of Information Communications, Xi'an University of Posts and Telecommunications, Xi'an 710121, China
[4] University of Guelph, Guelph, ON N1G2W1, Canada
wzhang25@uoguelph.ca
[5] Division of Information Science and Engineering, KTH Royal Institute of Technology, Stockholm, Sweden
ming.xiao@ee.kth.se

Abstract. The rapid development of the fifth generation (5G) promotes a variety of new applications, which will pose a huge challenge to the computing resources of networks. Computing networks is a promising technology, which can provide ubiquitous computing resources for applications in 5G. However, resource optimization in computing networks is still an open problem. In this paper, we propose a novel resource allocation framework for computing networks to investigate the energy consumption minimization problem in terms of delay constraint. To tackle the problem, we propose a dynamic task scheduling and resource allocation algorithm to utilizing the Lyapunov optimization method, which doesn't need to know any prior knowledge of networks. In order to reduce the complexity of solving the problem, we decompose the original problem into several sub-problem to solve. Particulary, the solutions of transmit power and subcarrier assignment are obtained by using the Lagrangian dual decomposition method. The solutions of computation time, postponing time, and CPU-cycle frequency are achieved in the closed form. Simulation results show that the performance of the proposed algorithms and can achieve the tradeoff between the average delay and the average energy consumption.

Keywords: Mobile edge computing (MEC) · Total cost saving · Network stability · Resource allocation

C. T. Calafate et al. (Eds.): MONAMI 2021, LNICST 418, pp. 127–137, 2022.
https://doi.org/10.1007/978-3-030-94763-7_10

1 Introduction

The fifth-generation (5G) network is paramount for supporting a multitude of emerging applications that require rich data communication in both directions, such as artificial intelligence (AI) [1,2] and virtual reality (VR) [3,4]. This is because it has the considerable capability to deliver high bandwidth and speed to enable high-reliability and rich two-way communication of high-volume data and high-definition video. Particularly, 5G technology creates enormous opportunities for the Internet of vehicles (IoV). With the rapid advancements of autonomous driving, the demand for computing power in the IoV services and applications has increased sharply.

Nevertheless, the proliferation of computing performance in 5G needs to be accelerated to keep up with these emerging applications that consume considerable amount of computing power [5]. Consequently, cloud computing [6] has become a popular paradigm aiming at providing on-demand availability of data storage and computing power [7]. Even though cloud computing has made data processing to be much more efficient, a major limitation is that it can cause large communication delay since the cloud center might be far away from users' devices [8].

As a more efficient alternative computing framework, Multi-access edge computing (MEC) is introduced to address the concern of communication latency [9–13], which has received widespread attention from both academia and industries. In terms of academic research, recent works studied multiple deployment approaches [14], joint radio-and-computational resource management [15], and applications which could take advantage of MEC [16–18]. In the industry side, several MEC based architectures have been developed, such as ThinkAir [19] and EdgeGallery [20].

In fact, how to achieve efficient computing power allocation and collaboration remains to be a big challenge in the deployment of MEC. To address this challenge, we aim of establishing deep integration between MEC and networking, i.e., building the Multi-access edge computing networks (MECN). The concept of a MECN is to use a ubiquitous network to connect distributed heterogeneous computational power, and provide personalized computational services for diverse upper layer services through unified management and on-demand scheduling [5,21]. However, the research on computing networks is still in early stage.

In this paper, we focus on jointly optimize transmit power, subcarrier assignment, computation time, postponing time, and CPU-cycle frequency to minimize the average energy consumption of the MEC-enabled computing networks. By using the Lyapunov optimization method, we develop a dynamic resource allocation algorithm to solve the minimum problem.

The main contributions of the paper are as follows:

– We propose a dynamic resource allocation scheme in MEC-enabled computing network to study the average energy consumption minimization problem, where transmit power, subcarrier assignment, computation time, postponing time, and CPU-cycle frequency are jointly optimized to stabilize the network.

- To resolve the problem, we design a dynamic radio and computation resource allocation algorithm, which does not need prior knowledge of the network. To improve the efficiency of the algorithm, we decompose the original problem into several sub-problems to obtain the optimal network control policy.
- In the simulation, we estimate the performance of the proposed algorithms. Observed from the simulation results, the proposed algorithm can achieve the trade off between the average delay and the average energy consumption.

The rest parts of this paper are organized as follows. In Sect. 2, The system model is described. We formulate the average energy consumption minimization problem in Sect. 3. In Sect. 4, an algorithm design framework is introduced. We design the network control policies in Sect. 5. In Sect. 5, we present the simulation results. Finally, the summary of the paper is given in Sect. 6.

2 System Model

We consider an MEC-enabled computing network that is composed of an MEC server and a set of applications. We assume that the MEC server is regarded as a computing provider medium installed at a base station (BS). Consequently, it can be accessed by applications through wireless communications to obtain the corresponding computing service. In the system, time is assumed to be slotted. We let N_t denote the number of tasks arriving in time slot t and $\mathcal{N}_t = \{1, 2, ..., N_t\}$ represent the index set of tasks. We consider two types of tasks, delay-intolerant and delay-tolerant tasks. Denoted by $\tau_n(t)$ the required service time for each task $n \in \mathcal{N}_t$. Let $k_n(t)$ represent the maximum delay allowed for each task from its arrival time t to completion, where $k_n(t) \geq \tau_n(t)$. Let $\boldsymbol{s(t)} = (s_n(t))$ denote the optimal "postponing" time for task n in time slot t. Assuredly, we have to $s_n(t) = 0$ for the delay-intolerant tasks.

2.1 Computation Offloading Model

We consider a non-interference communication environment in the system. There are M subcarriers, where $N_t \leq M$, and the bandwidth of each subcarrier is B. Let $\boldsymbol{P}(t) = (P_{n,m}(t))$ and $\boldsymbol{G}(t) = (G_{n,m}(t))$ represent the transmit power and the channel gain of task n on subcarrier m, respectively. Then, the transmit rate of task n on subcarrier m is given by

$$r_{n,m}(t) = B \log_2(1 + \frac{P_{n,m}(t) g_{n,m}(t)}{\sigma_n^2(t)}), \quad (1)$$

where $\sigma_n^2(t)$ is the noise power.

Consequently, the sum transmit rates and the total transmit power of task n are respectively expressed as $R_n(t)$ and $P_n(t)$.

We assume that the transmit time of task n is set to $T_n(t)$. Therefore, the amount of offloading task n to MEC server is denoted by $D_n^o(t) = T_n(t) R_n(t)$. The energy consumption of the task n is expressed as $E_n^o(t) = T_n(t) P_n(t)$.

We set that the power consumption of task n in idle state is $P^{id}(t)$, which is a variable that changes over time. Then the energy consumed while the task n is given by

$$E_n^s(t) = s_n(t)P^{id}(t). \tag{2}$$

Note that $E_n^s(t)$ relies on the decision made.

2.2 MEC Server Computation Model

After receiving the task n, the computation time of the MEC server is set to $\boldsymbol{e(t)} = (e_n(t))$. Let $\boldsymbol{f}(t) = (f_n(t))$ denote the CPU-cycle frequency of task n in time slot t. The data size of the task n executed by the MEC server is given by

$$D_n^m(t) = \frac{f_n(t)e_n(t)}{c_n}, \tag{3}$$

where c_n is the CPU cycles required by the MEC server to process 1-bit computing task n. Then the energy consumed by the MEC server to process the task n in time slot t is given by

$$E_n^m(t) = \kappa f_n(t)^3 e_n(t), \tag{4}$$

where κ is the effective switched capacitance of the MEC server.

2.3 Task and Computation Queueing Models

We denote that $\boldsymbol{A}(t) = (A_n(t))$ is the amount of arriving task n at the beginning of the time slot t, which is assumed to an independent and identically distributed (i.i.d) over slots. The arrival rate is $\boldsymbol{\lambda}(t) = (\lambda_n(t))$, and $\mathbb{E}\{\boldsymbol{A}(t)\} = \boldsymbol{\lambda}(t)$. We assume that the dynamic update of all queues is based on the First-In-First-Out (FIFO) principle. Thus, each task in the system is equally important. Let $\boldsymbol{Q}^o(t) \triangleq [Q_1^0(t), Q_2^o(t), ...Q_n^o(t)]$ denote the backlog of the task buffers at time slot t. Then, the queue length of the task n in the network evolves according to

$$Q_n^o(t+1) = \max\{Q_n^o(t) - R_n(t)(\tau_n(t) - e_n(t)), 0\} + A_n(t), n \in \mathcal{N}_t. \tag{5}$$

Usually, the MEC server stores that the tasks offloaded but not yet processed by the server in a task buffer. The capacity of the MEC server is assumed to be sufficiently large. Denote by $Q(t)$ the queue length of the task buffer at the MEC server at slot t. Then, the evolution of the queue length $Q(t)$ on the MEC server is given by

$$Q(t+1) = \max\{Q(t) - \sum_{n=1}^{N_t} D_n^m(t), 0\} + D_{in}(t), \tag{6}$$

where $D_{in}(t) = \sum_{n=1}^{N_t} D_n^o(t)$. We note that $0 \leq D_n^m(t) \leq Q(t)/N_t$.

3 Problem Formulation

In the paper, we focus on the total energy consumption of the task processing in the system, including the energy consumed by transmission, the energy consumed while waiting, and the energy consumption of the MEC server. Then the time average energy consumption of task n is given by

$$\overline{E}_n = \limsup_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}\{E_n(t)\}, \forall n \in \mathcal{N}_t, \tag{7}$$

where $E_n(t) = E_n^o(t) + E_n^s(t) + E_n^m(t)$.

Therefore, the average energy consumption of the system is equal to

$$\overline{E} = \sum_{n\in\mathcal{N}_t} \overline{E}_n. \tag{8}$$

We denote the system operation at time slot t as $\boldsymbol{A(t)} \triangleq [\boldsymbol{P(t)}, \boldsymbol{\rho(t)}, \boldsymbol{s(t)}, \boldsymbol{e(t)}, \boldsymbol{f(t)}]$. Therefore, the average energy consumption minimization problem can be formulated as

$$\begin{aligned}
&\min_{\boldsymbol{A}(t)} \quad \overline{E} = \sum_{n\in\mathcal{N}_t} \overline{E}_n \\
&\text{s.t.} \quad (\text{C1}): \sum_{m=1}^{M} \rho_{n,m}(t) r_{n,m}(t) \geq R_n^{req} \forall n, \\
&(\text{C2}): \sum_{n=1}^{N_t} \rho_{n,m}(t) \leq 1, \forall m, \\
&(\text{C3}): \rho_{n,m}(t) \in \{0,1\}, \forall n, m, \\
&(\text{C4}): \sum_{m=1}^{M} \rho_{n,m}(t) P_{n,m}(t) \leq P_{n,max}, \forall n, \\
&(\text{C5}): s_n(t) + e_n(t) \leq k_n(t) - T_n(t), \forall n, \\
&(\text{C6}): \limsup_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \sum_{n=1}^{N_t} \overline{y_n^t(s_n(t))} \leq \alpha \\
&(\text{C7}): f_{n,min} \leq f_n(t) \leq f_{n,max}, \forall n, \\
&(\text{C8}): P_{n,m}(t) \geq 0, f_n(t) \geq 0, \forall n, \\
&(\text{C9}): \frac{f_n(t) e_n(t)}{c_n} \leq Q(t)/N_t, \forall n, \\
&(\text{C10}): \overline{Q}_n < \infty, \overline{Q} < \infty, \forall n,
\end{aligned} \tag{9}$$

where R_n^{req} and $P_{n,max}$ denote the required transmission rate and the maximum transmission power of task n, respectively. $f_{n,min}$ and $f_{n,max}$ are respectively the minimum and the maximum CPU-cycle frequency for executing task n. We note that the value of $y_n^t(s_n(t))$ is different for diverse tasks.

4 Algorithm Design Framework

In this section, we propose a dynamic energy consumption minimum algorithm to solve the stochastic optimization problem (9) by employing the Lyapunov optimization method.

4.1 The Lyapunov Optimization-Based Algorithm

Firstly, we introduce the concept of virtual queue to deal with the average dissatisfaction constraint (C6) in (9). Particularly, the virtual queue is defined as $Q^v(t)$ with $Q^v(0) = 0$ and the update equations is defined as

$$Q^v(t+1) = \max\{Q^v(t) + \sum_{n=1}^{N_t} y_n^t(s_n(t)) - \alpha, 0\}. \tag{10}$$

Then solving problem (9) is transformed into minimizing the following problem.

$$\begin{aligned} &\min_{\boldsymbol{X}(t)} \Phi(\boldsymbol{X}(t)) \\ &\text{s.t.} \quad (\text{C1})-(\text{C5}), (\text{C7})-(\text{C9}), \end{aligned} \tag{11}$$

where

$$\Phi(\boldsymbol{X}(t)) = V\sum_{n=1}^{N_t}(\tau_n(t) - e_n(t))\sum_{m=1}^{M}\rho_{n,m}(t)P_{n,m}(t) - \sum_{n=1}^{N_t}(Q_n^o(t) - Q(t))(\tau_n(t) - e_n(t))$$
$$\sum_{m=1}^{M}\rho_{n,m}(t)B\log_2(1 + \frac{P_{n,m}(t)g_{n,m}(t)}{\sigma_n(t)^2}) + \sum_{n=1}^{N_t}\{V(\kappa f_n(t)^3 - P_n(t)) - [R_n(t)(Q_n^o(t)$$
$$-Q(t)) + Q(t)\frac{f_n(t)}{c_n}]\}e_n(t) + Q^v(t)\sum_{n=1}^{N_t}y_n^t(s_n(t)) + V\sum_{n=1}^{N_t}s_n(t)\sum_{j=0}^{\tau_n-1}P^{id}(j+t+s_n(t)),$$
$$V\sum_{n=1}^{N_t}\kappa f_n(t)^3 e_n(t) - Q(t)\sum_{n=1}^{N_t}\frac{f_n(t)e_n(t)}{c_n}$$

To solve (9), we develop a dynamic resource allocation algorithm, shown in Algorithm 1.

Algorithm 1. Dynamic Resource Allocation Algorithm

1: Monitor the current queue states $\boldsymbol{Q}^o(t)$, $\boldsymbol{Q}(t)$, $\boldsymbol{Q}^v(t)$ and the channel condition $\boldsymbol{G}(t)$ at each time slot t, respectively;
2: Obtain the network control policy, including power allocation, subcarrier assignment, computation time, postponing time, and CPU-cycle frequency accordingly to problem (11);
3: Update $\boldsymbol{Q}^o(t)$, $\boldsymbol{Q}(t)$, $\boldsymbol{Q}^v(t)$ according to (8), (9), and (14) based on the above obtained control policy.

---

5 Design of Control Strategies

In this section, we first develop an algorithm for power allocation and subcarrier assignment. We then obtain computation time allocation, the postponing time, and CPU-cycles frequency, respectively.

5.1 Power Allocation and Subcarrier Assignment

For a given $\boldsymbol{e}(t)$, the problem of the power allocation and subcarrier assignment is given by

$$\begin{aligned} &\min_{\boldsymbol{P}(t),\boldsymbol{\rho}(t)} \Phi_1(\boldsymbol{P}(t),\boldsymbol{\rho}(t)) \\ &\text{s.t.} \quad (\text{C1})-(\text{C4}), (\text{C8})' : P_{n,m}(t) \geq 0. \end{aligned} \tag{12}$$

To solve (12), we first relax $\rho_{n,m}(t)$ to interval $[0,1]$, then introduce a new variable $x_{n,m}(t) = \rho_{n,m}(t)P_{n,m}(t)$, where let $\boldsymbol{x}(t) = (x_{n,m}(t))$. We set $\rho_{n,m}(t)B\log_2(1+\frac{P_{n,m}(t)g_{n,m}(t)}{\sigma_n(t)^2} = 0$ when $\rho_{n,m}(t) = 0$. Therefore, (12) is rearranged as

$$\begin{aligned} &\min_{\boldsymbol{x}(t),\boldsymbol{\rho}(t)} \Phi_1(\boldsymbol{x}(t),\boldsymbol{\rho}(t)) \\ &\text{s.t.} \quad (\text{C1}), (\text{C2}), (\text{C3}), (\text{C4}), (\text{C8}). \end{aligned} \tag{13}$$

It is easy to prove that optimization problem (13) is a convex optimization problem. Therefore, we can easily obtain the solution of (13).

5.2 Computation Time Allocation

For a given $\boldsymbol{P}(t), \boldsymbol{\rho}(t), \boldsymbol{f}(t)$, and $\boldsymbol{s}(t)$, the optimization problem of computation time allocation is given by

$$\begin{aligned} &\min_{\boldsymbol{e}(t)} \Phi_2(\boldsymbol{e}(t)) \\ &\text{s.t.} \quad (\text{C5}) : e_n(t) \leq k_n(t) - T_n(t) - s_n(t), \forall n, \\ &\quad (\text{C9}) : e_n(t) \leq \frac{Q(t)c_n}{N_t f_n(t)}, \forall n. \end{aligned} \tag{14}$$

We can observe that (14) is a standard linear programming, so the optimal computation time is denoted as $e_n^*(t)$.

5.3 Postponing Time Allocation

For a given computation time $\boldsymbol{e}(t)$, the optimization problem of postponing time allocation is given by

$$\begin{aligned} &\min_{\boldsymbol{s}(t)} \Phi_3(\boldsymbol{s}(t)) \\ &\text{s.t.} \quad (\text{C5}) : s_n(t) \leq k_n(t) - T_n(t) - e_n(t), \forall n. \end{aligned} \tag{15}$$

Since the objective function of (15) is convex and its constraint is linear, (15) is a convex optimization problem. so the optimal postponing time is given by

$$s_n^*(t) = \arg \min_{0 \le s_n(t) \le k_n(t) - T_n(t) - e_n(t)} Q^v(t) y_n^t(s_n(t)) + V \sum_{j=0}^{\tau_n - 1} P^{id}(j + t + s_n(t)). \quad (16)$$

5.4 Optimal CPU-Cycle Frequency

For a given computation time $\boldsymbol{e}(t)$, the optimization problem of CPU-cycle frequency is given by

$$\begin{aligned} &\min_{\boldsymbol{f}(t)} \Phi_4(\boldsymbol{f}(t)) \\ &\text{s.t.} \quad (\text{C7})' : f_{n,min} \le f_n(t) \le \min\{f_{n,max}, \frac{c_n Q(t))}{N_t e_n(t)}\}, \forall n. \end{aligned} \quad (17)$$

In (17), we can observe that both its objective function and constraint are convex, so (17) is convex optimization problem. Then, the optimal solution of CPU-cycle frequency for any user n is denoted as $f_n^*(t)$.

6 Simulation Results

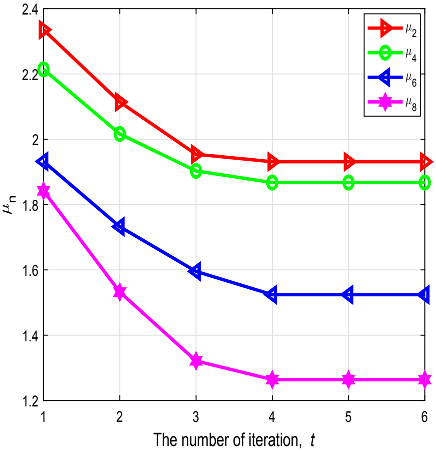

Fig. 1. Convergence of Algorithm 2.

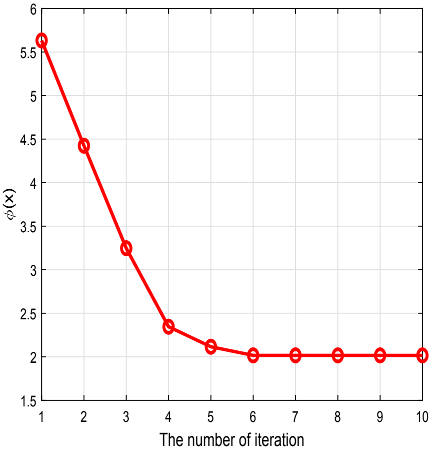

Fig. 2. Convergence of Algorithm 3.

In this section, we show simulation results to analyze the proposed algorithms. There are $N = 8$ users located at the central of BS. We assume that the arrival process of all tasks obeys follows Poission process with different intensity λ_n. In the simulation process, we consider two types of applications, delay-tolerant tasks and delay-tolerant tasks. The arrival intensities for the tasks are $2.5, 1.5, 0.2,$ and 40, respectively. The functions of dissatisfaction are assumed to be $y(x) = x^2 + x$. The maximum CPU computation capacity f_{max} is set to be 2 GHz. The parameter α of the average dissatisfaction constraint is set to be 5000, and we set the control parameter $V = 200$. To evaluate the performance of the proposed algorithms, we give two schemes as comparison scheme. The first scheme is that the variables of power allocation, computation time, and CPU-cycle frequency allocation are separately optimized (Policy A). The second scheme only optimize power allocation and CPU-cycle frequency allocation (Policy B).

In Fig. 1, we give the performance evaluation of Algorithm 2. Peculiarly, the trend of the dual variables $\boldsymbol{\mu}(t) = (\mu_n(t)), n \in N_t$ is regards as the judgment target. Observed from Fig. 1, we can find that the convergence rate of Algorithm 2 is fast. Figure 2 shows the convergence rate of Algorithm 3. It is observed that it has also a fast convergence rate. Therefore, we can conclude that the proposed algorithm is highly efficient.

Figure 3 shows the comparison of the average energy consumption and control parameter V. The average energy consumption deceases as V increase for a given arrival rate $\lambda_n(t)$. Besides, observed from Fig. 3, we can see that the average energy consumption increases as arrival rate $\lambda_n(t)$ for a given control parameter V.

In Fig. 4, we display the average energy consumption of with the variation of maximum transmit power $P_{n,max}$. The average energy consumption deceases as maximum transmit power $P_{n,max}$ increase for a given control parameter $V = 100$. By comparing the two schemes (Policy A and Policy B), we can conclude

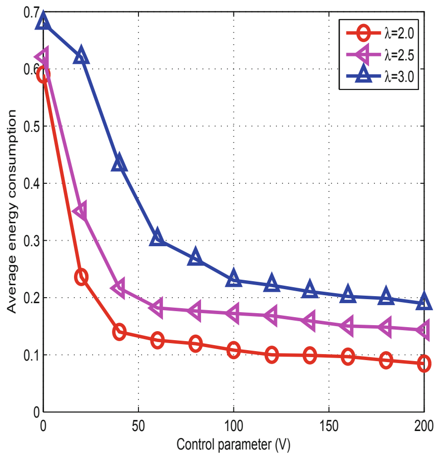

Fig. 3. Average energy consumption vs. control parameter V.

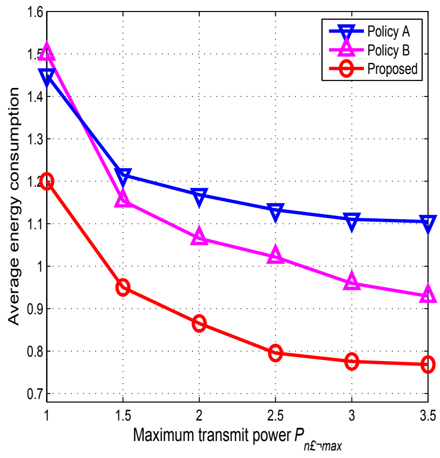

Fig. 4. Average energy consumption vs. maximum transmit power $P_{n,max}$.

that the proposed scheme consumes the least energy, followed by Policy B, and finally Policy A.

7 Conclusion

In this paper, we have researched the average energy consumption minimization problem, where jointly optimize transmit power allocation, subcarrier assignment, computation time, postponing time, and CPU-cycle frequency for MEC-enabled computing networks. We propose an algorithm that does not require prior knowledge of the network in advance. Particulary, the solutions of transmit power and subcarrier assignment were obtained by using the Lagrangian dual decomposition method. The solutions of computation time, postponing time, and CPU-cycle frequency were achieved in the closed form. Simulation results shown that the performance of the proposed algorithms and can achieve the tradeoff between the average delay and the average energy consumption.

Acknowledgment. This work is supported by the National Key Research and Development Program of China (No. 2020YFB1807500), the National Natural Science Foundation of China (NO. 6210070336, No. 62001357), the Key Research and Development Programs of Shaanxi (No. 2021ZDLGY06-03, No. 2019ZDLGY13-07), the Guangdong Basic and Applied Basic Research Foundation (2020A1515110496, 2020A1515110079), and the Fundamental Research Funds for the Central Universities (No. XJS210105, XJS210107). Finally, thank Xinyi Tang for all her hard work in organizing the related materials, participating in the academic discussion.

References

1. Yang, T., Chen, J., Zhang, N.: Ai-empowered maritime Internet of Things: a parallel-network-driven approach. IEEE Network **34**(5), 54–59 (2020)
2. Liu, L., et al.: Blockchain-enabled secure data sharing scheme in mobile-edge computing: an asynchronous advantage actor-critic learning approach. IEEE Internet Things J. **8**(4), 2342–2353 (2020)
3. Sukhmani, S., Sadeghi, M., Erol-Kantarci, M., El Saddik, A.: Edge caching and computing in 5G for mobile AR/VR and tactile internet. IEEE MultiMedia **26**(1), 21–30 (2018)
4. Yang, T., Kong, L., Zhao, N., Sun, R.: Efficient energy and delay tradeoff for vessel communications in SDN based maritime wireless networks. IEEE Trans. Intell. Transp. Syst. **22**(6), 3800–3812 (2021)
5. Wang, X., Ren, X., Qiu, C., Cao, Y., Taleb, T., Leung, V.C.: Net-in-AI: a computing-power networking framework with adaptability, flexibility and profitability for ubiquitous AI. IEEE Network **35**(1), 280–288 (2020)
6. Armbrust, M., et al.: A view of cloud computing. Commun. ACM **53**(4), 50–58 (2010)
7. Montazerolghaem, A., Yaghmaee, M.H., Leon-Garcia, A.: Green cloud multimedia networking: NFV/SDN based energy-efficient resource allocation. IEEE Trans. Green Commun. Networking **4**(3), 873–889 (2020)

8. Shi, W., Dustdar, S.: The promise of edge computing. Computer **49**(5), 78–81 (2016)
9. Taleb, T., Samdanis, K., Mada, B., Flinck, H., Dutta, S., Sabella, D.: On multi-access edge computing: a survey of the emerging 5G network edge cloud architecture and orchestration. IEEE Commun. Surv. Tutorials **19**(3), 1657–1681 (2017)
10. Feng, J., Yu, F.R., Pei, Q., Chu, X., Du, J., Zhu, L.: Cooperative computation offloading and resource allocation for blockchain-enabled mobile-edge computing: a deep reinforcement learning approach. IEEE Internet Things J. **7**(7), 6214–6228 (2019)
11. Mao, S., Wu, J., Liu, L., Lan, D., Taherkordi, A.: Energy-efficient cooperative communication and computation for wireless powered mobile-edge computing. IEEE Syst. J. 1–12 (2020). https://doi.org/10.1109/JSYST.2020.3020474
12. Feng, J., Yu, F.R., Pei, Q., Du, J., Zhu, L.: Joint optimization of radio and computational resources allocation in blockchain-enabled mobile edge computing systems. IEEE Trans. Wirel. Commun. **19**(6), 4321–4334 (2020)
13. Liu, L., Chen, C., Pei, Q., Maharjan, S., Zhang, Y.: Vehicular edge computing and networking: a survey. Mobile Networks Appl. **26**(3), 1145–1168 (2021)
14. Beck, M.T., Werner, M., Feld, S., Schimper, S.: Mobile edge computing: a taxonomy. In: Proceedings of the Sixth International Conference on Advances in Future Internet, pp. 48–55. Citeseer (2014)
15. Mao, Y., You, C., Zhang, J., Huang, K., Letaief, K.B.: A survey on mobile edge computing: the communication perspective. IEEE Commun. Surv. Tutorials **19**(4), 2322–2358 (2017)
16. Yi, S., Li, C., Li, Q.: A survey of fog computing: concepts, applications and issues. In: Proceedings of the 2015 Workshop on Mobile Big Data, pp. 37–42 (2015)
17. Yang, T., Qin, M., Cheng, N., Xu, W., Zhao, L.: Liquid software-based edge intelligence for future 6G networks. IEEE Network (2021, to appear)
18. Du, J., Yu, F.R., Lu, G., Wang, J., Jiang, J., Chu, X.: MEC-assisted immersive VR video streaming over terahertz wireless networks: a deep reinforcement learning approach. IEEE Internet Things J. **7**(10), 9517–29 (2020)
19. Kosta, S., Aucinas, A., Hui, P., Mortier, R., Zhang, X.: Thinkair: dynamic resource allocation and parallel execution in the cloud for mobile code offloading. In: Proceedings IEEE Infocom, pp. 945–953. IEEE (2012)
20. Edgegallery. https://www.edgegallery.org/en/
21. Mao, S., et al.: Computation rate maximization for intelligent reflecting surface enhanced wireless powered mobile edge computing networks. IEEE Trans. Veh. Technol. **7**, 1 (2021). https://doi.org/10.1109/TVT.2021.3105270