Energy-Efficient Partial Offloading with Transmission Power Control in Mobile-Edge Computing

Meng Wang[1], Shuo Shi[1,2,3](✉), Jian He[1], Cong Zhou[1], and Zhong Zheng[1,3]

[1] School of Electronic and Information Engineering, Harbin Institute of Technology, Harbin 150001, Heilongjiang, China
crcss@hit.edu.cn

[2] Network Communication Research Centre, Peng Cheng Laboratory, Shenzhen 518052, Guangdong, China

[3] International Innovation Institute of HIT in Huizhou, Huizhou 516000, Guangdong, China

Abstract. Mobile-Edge Computing (MEC) is a promising paradigm which enables mobile devices (MDs) to offload the computation-sensitive tasks (e.g., Augmented Reality) to the MEC server in close proximity to MDs to obtain low execution latency and energy consumption. Different from traditional partial offloading scheme, in this paper, we study the partial offloading in MEC. Consider single user scenario and Rayleigh fading channel, we first formulate the energy-efficient partial offloading problem as a non-convex problem. To solve the problem optimally, we reformulate the problem into two subproblems and use block coordinate descent method to solve the problem separately. For each subproblem, we give the proof of its feasibility. Simulation results show that the proposed method consumes less energy compared with two benchmark algorithms.

Keywords: Mobile-edge computing · Partial computation offloading · Convex optimization · Transmission power control

1 Introduction

Mobile devices (MDs) has been gaining enormous attention since 4G era. With the continuous development of information and communication technology (ICT), more and more new mobile applications with high requirement on quality of experience (QoE) are envisioned to be integrated into smart MDs [1]. However, those novel applications including augmented reality (AR), virtual reality (VR) are typically resource-hungry, requiring huge computation capacity [2]. Although the processing power of MDs has been greatly improved in recent years, due to the physical size limit, MDs are still lack of computation resource and battery life. The contradiction between resource-sensitive services and resource-constrained MDs poses a great challenge for future ICT system.

X. Wang et al. (Eds.): AICON 2021, LNICST 396, pp. 515–525, 2021.
https://doi.org/10.1007/978-3-030-90196-7_44

One promising approach to address such a challenge is mobile cloud computing, where MDs offload their resource-hungry tasks to the remote cloud center (e.g., Amazon EC2, Windows Azure) with rich computation resource through wireless access to reduce the energy consumption of MDs [3]. However, MDs may experience a long latency to finish the task since the cloud servers are typically far away from MDs and the backhaul network is congested. High task processing latency will hurt the QoE severely, since people are sensitive to delay and jitter. To address this issue, mobile-edge computing (MEC) as a new paradigm has been proposed in recent years [4]. The key idea for MEC is to set powerful servers in close proximity to MDs (e.g. wireless access point, base stations), so that MDs can offload the tasks with fast connection and low latency.

Many existing works have been focused on the computation offloading issue in MEC. [5] proposes a distributed energy-efficient computation offloading scheme based on game theory and shows the convergence and effectiveness. [6] formulates the multi-user computation offloading problem as a mixed integer nonlinear problem and prove its NP-hard property, and solves the problem suboptimally through reinforcement learning approach. [7] studies the energy-optimal mobile-cloud computing under stochastic wireless channel, which shows the impact of wireless channel in computation offloading. [8] studies the joint task offloading and resource allocation for multi-server MEC networks and decomposes the MINLP problem into two subproblems. [9] studies the energy-latency tradeoff for energy-aware offloading in MEC networks, where communication and computation resource allocation are jointly optimized under the limited energy and latency constraints. Although the existing works are insightful, they mainly focused on binary computation offloading, namely each task is either offloaded to the MEC server or executed locally. However, for some typical tasks, e.g., virus scan and image compression, a task can be partitioned into several subtasks so that a MD can offload part of the task to MEC server and process the rest of the task locally. Moreover, the performance of computation offloading is affected significantly by wireless channel state, so it is important to analyze the computation offloading issue under realistic channel state. Toward this end, [12] studies the energy-efficient partial computation offloading and resource allocation issue under IoT scenario. The problem is formulated as an non-convex optimization problem solved by block coordinate descent and successive convex optimization method. [13] studies the online partial offloading method in multi-user MEC networks by leveraging Deep Deterministic Policy Gradient (DDPG) framework. [14] studies the stochastic partial computation offloading problem where the stability issue is considered. The authors use lyapunov optimization method to convert the complex global optimization problem into a single time slot decision problem. However, the algorithm proposed in [12] is with high complexity so that it cannot adapt to the real-time MEC. Moreover, [13] faces the challenge of convergence, and DDPG needs a lot of pre-training. [14] gives a $[O(1/V), O(V)]$ tradeoff between stability and effectiveness, however it does not consider the realistic wireless channel model.

Different from existing works, in this paper, we focus on the jointly optimizing the partial offloading ratio and the transmission power of the MD. Specifically, the optimization objective is the total energy consumption in a time period under the task process latency constraint. In the single-user scenario with Rayleigh fading channel, we first formulate the energy-efficient partial offloading problem as a non-convex problem. By using block coordinate descent method, we reformulate the problem into two subproblems and solve the problem separately with low complexity. Our contributions can be summarized as follows:

1) Different from previous work, we study the partial offloading method under rayleigh fading channel through monte carlo method, and the transmission power control issue is jointly considered.

2) We formulate the energy-efficient partial offloading problem as a non-convex problem and decompose the problem into two subproblems. Instead of using traditional block coordinate descent method, we analyze the monotonicity property of each subproblem and solve them with low-complexity.

3) Simulation results show that our proposed partial offloading method consumes less energy compared with three benchmark algorithms.

The rest of this paper is organized as follows: Sect. 2 gives the system model and problem formulation. In Sect. 3, we propose our energy-efficient partial offloading method to solve the non-convex optimization problem. Finally, simulation and results are given in Sect. 4 and 5, respectively.

2 System Model and Problem Formulation

2.1 Network Model

Figure 1 gives the network model. Specifically, we consider a single MD either offloading the tasks to a single MEC server located in an e-NodeB or execute the task locally. Different from previous works in binary offloading, partial offloading in this paper allows a single task being partitioned (e.g., virus scan, video compression) so that it can be executed simultaneously both in MEC server and local processor (i.e., CPU). We consider a discrete time period $\mathcal{T} = \{1, 2, 3, ..., T\}$, in each time slot $t \in \mathcal{T}$ a task $A(X, L)$ needs to be executed, where X denotes the size, L denotes the maximum tolerant delay of the task, respectively. According to [10], each bit of the task need to consume α CPU circles. $\lambda(t) \in [0, 1]$ represents the offloading ratio to the MEC server in time slot t.

2.2 Communication Model

In each time slot, the MD offloads the task through wireless channel. For convenience, we consider the channel coherence time is larger than a single time slot, namely the channel fading is constant in each time slot but varies among different time slots. Considering the white noise power density is N_0, the uplink transmission rate in time slot t is given by:

$$r(t) = B\log_2(1 + \frac{P_t|h(t)|^2}{N_0 B}), \tag{1}$$

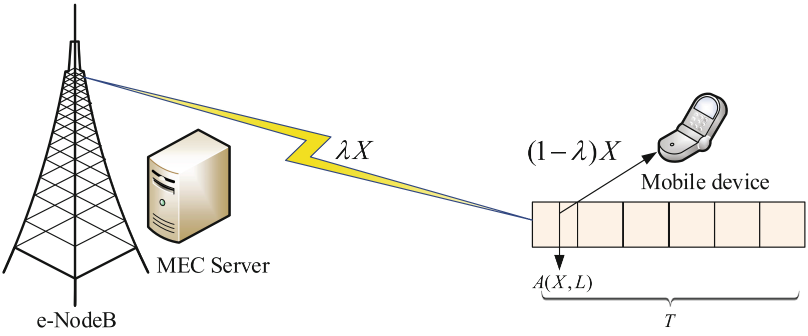

Fig. 1. Single-user partial offloading in mobile-edge computing

where B denotes the bandwidth allocated to the MD, P_t is the transmission power, $h(t)$ is a random variable which represents the channel fading coefficient. In this paper, we assume the probabilistic density function (PDF) of $h(t)$ is given as:

$$p(h(t) = r) = \frac{r}{\sigma^2} \exp(-\frac{r^2}{2\sigma^2}), 0 < r < \infty, \tag{2}$$

where σ^2 is the average receive power. According to the probabilistic theory, the PDF of $|h(t)|^2$ is given as:

$$p(|h(t)|^2 = r) = \frac{1}{2\sigma^2} \exp(-\frac{r}{2\sigma^2}), 0 < r < \infty. \tag{3}$$

In this paper, we assume the base station have perfect channel state information (CSI) of the uplink channel. The uplink transmission time can thus be modelled as:

$$T_U(t) = \frac{\lambda(t)X}{r(t)}. \tag{4}$$

The transmission energy in each time slot is given as:

$$E_U(t) = P_t T_U(t). \tag{5}$$

According [5], we do not take the downlink transmission into consideration, since after processing in MEC server the size of the task and the downlink transmission time is negligible.

2.3 Computation Model

According to [7], we model the computation energy consumption of CPU as

$$P = \theta f^3, \tag{6}$$

where θ and k are the coefficient and computation speed of the CPU, respectively. Since f denotes CPU cycles per second, the energy consumption per cycle can be expressed as θf^2. Thus, the computation energy consumption in MD can be expressed as

$$E_l(t) = \alpha(1-\lambda(t))X\theta f^2. \tag{7}$$

In this paper, we focus on the energy consumption in MD, thus the energy consumption in MEC server is not considered. Furthermore, the execution delay is given as

$$T_l(t) = \frac{(1-\lambda(t))X\alpha}{f} \tag{8}$$

2.4 Problem Formulation

Based on the model, the total energy consumption in a single time slot is given as

$$E_{total}(t) = E_U(t) + E_l(t) \tag{9}$$

The total execution latency in time slot t is given as

$$T_{total}(t) = \max(T_U(t), T_l(t)) \tag{10}$$

Therefore, the energy-efficient partial computation offloading problem can be formulated as

$$\begin{aligned}
(\mathbf{P1}) \quad & \min_{\boldsymbol{\lambda},\boldsymbol{P}} \sum_{t=1}^{T} \alpha(1-\lambda(t))X\theta f^2) + \frac{P(t)\lambda(t)X}{B\log_2(1+\frac{P(t)|h(t)|^2}{N_0 B})} \\
s.t. \quad & C1: 0 \leq \lambda(t) \leq 1 \\
& C2: 0 \leq P(t) \leq P_{\max} \\
& C3: T_{total}(t) < L
\end{aligned}$$

The objective is to minimize the total energy consumption of the MD in a time period T. C1 gives the range of λ, C2 constrains the transmission energy, C3 guarantees the task should be finished under the maximum tolerant delay constraint.

3 Proposed Energy-Efficient Partial Offloading Method

Note that in P1, $\boldsymbol{\lambda}$ and $\boldsymbol{P}$ are coupled in both objective function and C3, therefore P1 is not a standard convex optimization problem. To obtain the optimal solution, we first reformulate the problem as:

$$\begin{aligned}
(\mathbf{P2}) \quad & \min_{\lambda,P} \alpha(1-\lambda)X\theta f^2 + \frac{P\lambda X}{B\log_2(1+\frac{P|h|^2}{N_0 B})} \\
s.t. \quad & C1: 0 \leq \lambda \leq 1 \\
& C2: 0 \leq P \leq P_{\max} \\
& C3: T_{total} < L
\end{aligned}$$

Since in each time slot t MU needs to execute one independent task, in order to minimize the total energy consumption in time period T, we just need to minimize the energy consumption in every single time slot t, which reduces the number of optimization variables significantly. However, P2 is still non-trivial to solve due to the couple of λ and P and the non-convex optimization objective and constraint.

Lemma 1: It is always true that

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

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

[11] gives the complete proof, we do not expand in this paper due to the page limit. Lemma 1 allows us to solve P2 through minimizing λ and P sequentially.

We first consider P as a constant and try to find λ^*. The problem can then be formulated as:

$$\begin{aligned}
(\mathbf{P3}) \quad & \min_{\lambda} \alpha(1-\lambda)X\theta f^2 + \frac{P\lambda X}{B\log_2(1+\frac{P|h|^2}{N_0 B})}, \\
s.t. \quad & C1: 0 \le \lambda \le 1 \\
& C2: \max\{\frac{\lambda X}{r}, \frac{(1-\lambda)X\alpha}{f}\} < L
\end{aligned}$$

where $r = B\log_2(1+\frac{P|h|^2}{N_0 B})$ is the uplink transmission rate in time slot t. It can be observed that P3 is a linear programming problem with respect to λ, thus it can be solved through convex optimization.

Theorem 1. *The optimal value of λ in each time slot is:*

$$\lambda^* = \begin{cases} \min\left\{\frac{1-Lf}{X\alpha}, 0\right\}, r < \frac{P}{\alpha\theta f^2} \\ \forall\lambda \in [0,1], r = \frac{P}{\alpha\theta f^2} \\ \max\left\{\frac{Lr}{X}, 1\right\}, r > \frac{P}{\alpha\theta f^2} \end{cases}$$

Proof. We rewrite the objective function as $f(\lambda) = \lambda(\frac{PX}{r} - \alpha X\theta f^2) + \alpha X\theta f^2$. Since $f(\lambda)$ is a linear function of λ. The minimum value of $f(\lambda)$ is determined by the sign of $f(r) = \frac{PX}{r} - \alpha X\theta f^2$. If $f(r) > 0$, to obtain the minimum $f(\lambda)$, $\lambda^* = \min\{\lambda|\lambda \in dom\}$. Relatively, If $f(r) < 0$, to obtain the minimum $f(\lambda)$, $\lambda^* = \max\{\lambda|\lambda \in dom\}$. For the special case when $f(r) = 0$, $\lambda = \forall\lambda \in [0,1]$ since the objective value is not related with λ. Furthermore, C2 should be satisfied. $\lambda_0 = \frac{1}{1+\frac{\alpha r}{f}}$ denotes the case when $\frac{\lambda X}{r} = \frac{(1-\lambda)X\alpha}{f}$. If $\lambda > \lambda_0$, we have $\lambda_0 < \lambda < \frac{Lr}{X}$. If $\lambda < \lambda_0$, we have $\frac{1-Lf}{X\alpha} < \lambda < \lambda_0$. Taking intersection with C2, we finally finish the proof.

By using λ^* obtained from P3, we now consider the P as optimization variable and try to find P^*. The problem can then be formulated as

$$
\begin{aligned}
(\mathbf{P4}) \quad & \min_{P} \alpha(1-\lambda)X\theta f^2 + \frac{P\lambda X}{B\log_2(1+\frac{P|h|^2}{N_0 B})} \\
& s.t. \qquad C1: 0 \le P \le P_{\max} \\
& \qquad\qquad C2: \max\left\{\frac{\lambda X}{B\log_2(1+\frac{P|h|^2}{N_0 B})}, \frac{(1-\lambda)X\alpha}{f}\right\} < L
\end{aligned}
$$

Theorem 2. $f(P) = \frac{\lambda X P}{B\log_2(1+\frac{P|h|^2}{N_0 B})}$ *is monotonically increasing with respect to variable* P.

Proof. Let $\Delta P \to 0$, we define function $g(P) = f(P+\Delta P) - f(P)$. By substituting $f(P)$, $g(P)$ can be expressed as: $g(P) = \frac{abP[\log_2(1+aP)-\log_2(1+aP+a\Delta P)]+ab\Delta P\log_2(1+aP)}{a^2\log_2(1+aP+\Delta P)\log_2(1+aP)}$. Since the denominator is always positive, we observe the numerator $h(P) = P\left[\log_2(1+aP) - \log_2(1+aP+a\Delta P)\right] + \Delta P\log_2(1+aP)$. Divide both sides of the equation by the term $P\log_2(1+aP)$: $\frac{h(P)}{P\log_2(1+ap)} = 1 - \log_2(a\Delta P) + \frac{\Delta P}{P} > 1 - \log_2(a\Delta P)$. Since, $\Delta P \to 0$, thus $1 - \log_2(a\Delta P) > 0$, which indicates $h(P) > 0$, i.e., $g(P) > 0$. Since $f(P+\Delta P) - f(P) > 0$, we finish the proof.

Theorem 2 gives the fact that the optimization objective value is monotonically increasing with respect to P, thus the minimum value can be obtained according to the lower bound of the domain. Since λ is constant in C2 and $\frac{\lambda X}{B\log_2(1+\frac{P|h|^2}{N_0 B})}$ is monotonically decreasing with respect to P. Define $P_0 = \frac{N_0 B}{|h|^2}\left[2^{\frac{\lambda f}{(1-\lambda)B\alpha}} - 1\right]$ which satisfies $\frac{\lambda X}{B\log_2(1+\frac{P_0|h|^2}{N_0 B})} = \frac{(1-\lambda)X\alpha}{f}$. Therefore, when $P_{max} > P > P_0$, $\frac{(1-\lambda)X\alpha}{f} < L$ must hold, this gives the feasible set of L, namely $L > \frac{X\alpha}{f}$, and $P^* = P_0$. When $P < P_0$, $\frac{\lambda X}{B\log_2(1+\frac{P|h|^2}{N_0 B})} < L$, which gives $P > \frac{N_0 B}{|h|^2}\left(2^{\frac{\lambda X}{LB}} - 1\right)$. Thus, when $0 < P < P_0$, $P^* = \frac{N_0 B}{|h|^2}\left(2^{\frac{\lambda X}{LB}} - 1\right)$. Finally, we solve P4 with $P^* = \min\left\{\frac{N_0 B}{|h|^2}\left(2^{\frac{\lambda X}{LB}} - 1\right), P_0\right\}$.

Algorithm 1 gives the detailed procedure of the proposed algorithm. For each task i, we do a total of $\mathcal{M}$ monte carlo simulations to illustrate the statistic property of rayleigh fading. We first give the input parameters according to [5–10]. For each task i, we first take λ as variable and give an initial value P_0 to find the optimum λ^* according to Theorem 1. Then we take P as variable and let $\lambda = \lambda^*$, and find the optimum P^* through Theorem 2. The average energy consumption is the mean value of $f_j(\lambda^*, P^*)$.

Algorithm 1. Proposed energy-efficient partial offloading method

Initialize input parameters B, N_0, θ, f, α, X, L, M
for task i in $\mathcal{N}$ **do**
  for j in $\mathcal{M}$ **do**
    Obtain the channel coefficient $h(t)$ through channel state information
    Take λ as variable, let $P = P_0$
    Find λ^* through linear programming method according to Theorem 1
    Take P as variable, let $\lambda = \lambda^*$
    Calculate $P^* = \min\left\{\frac{N_0 B}{|h|^2}\left(2^{\frac{\lambda X}{LB}} - 1\right), P_0\right\}$.
    Find the optimum value of the objective function $f_j(\lambda^*, P^*)$ in j-th simulation
  end for
  Find the average energy consumption of task i under rayleigh fading channel.
end for
Calculate total energy consumption $E_{total} = \sum\limits_{i=1}^{N} f_i(\lambda^*, P^*)$

4 Simulation Results

4.1 Parameters Settings

The main parameters are listed in Table 1. Specifically, we consider a time period T consists of 100 time slots. The bandwidth and the white noise density is set as 10 MHz and 10^{-9} W/Hz, respectively. The size and maximum tolerant delay of each task are assumed to be equal in this paper, with specific value 2000 Kb and 0.1 s, respectively. The computation capacity of MD is set as 1 GHz and the maximum transmission power of MD is 0.2 W. According to [10], α is set as 40, θ is set as 10^{-26}. The simulation environment is Windows 10 under Python 3.7.

Table 1. Parameters settings

Parameters	Values
Bandwidth B	$2 * 10^6$
Noise power density N_0	10^{-9}
Data input size X	500–2000 KB
Maximum tolerant delay L_{max}	0.1 s
Maximum CPU frequency of mobile devices f_{max}	1 GHz
Maximum transmission power P_t	0.2 W
Computation factor α	40
Average receive power σ^2	0.1 W
Computation energy coefficient θ	10^{-26}

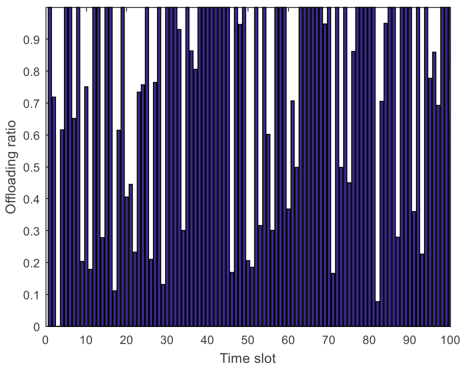

Fig. 2. Offloading ratio with respect to each executed task

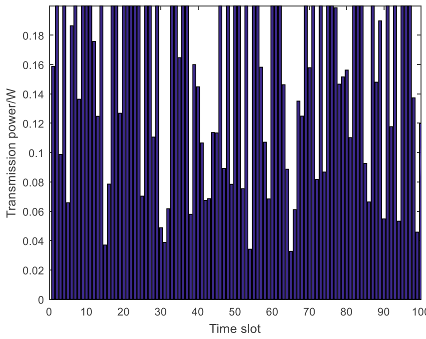

Fig. 3. Transmission power with respect to each executed task

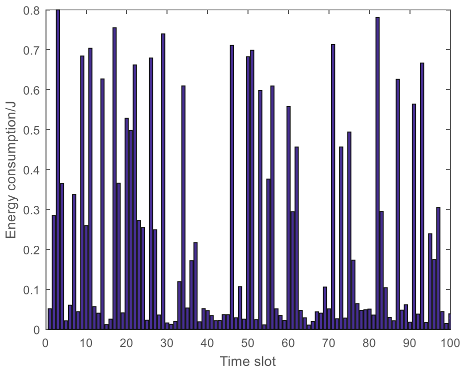

Fig. 4. Total energy consumption versus each task

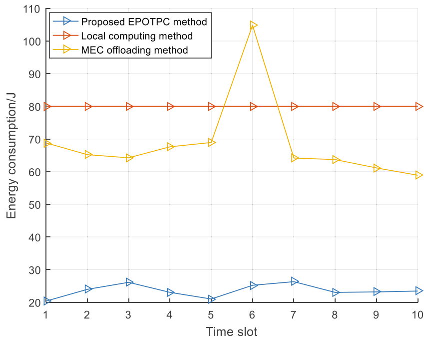

Fig. 5. Performance comparison with benchmark algorithms

4.2 Performance Analysis

Figure 2 gives the offloading ratio versus each executed task. Note that, the offloading ratio varies with each time slot. However, the tasks in this paper all have the same property, which indicates that the channel state has significant influence on the offloading ratio. When the channel fading is severe, MDs tend to offload a small part of taks to MEC server to obtain a better performance. When the channel state is ideal, MDs tend to offload the whole task to the MEC server, since the transmission energy and latency are relatively low.

Figure 3 gives the transmission power with respect to each executed task, the maximum transmission power of each MD is 0.2 W. Figure 3 shows that instead of transmitting message with maximum power, the transmission power can change adaptively under the latency constraints, which can reduce the energy consumption of MDs significantly. Furthermore, this result also shows the impact of wireless channel in MEC, namely when the channel fading is severe, the MD needs to use higher transmission energy to guarantee the transmission rate so that the transmission delay can be reduced.

Figure 4 gives the total energy consumption with respect to each task. Note that, the energy consumption of each task varies significantly. The reason is that local execution consumes more energy than offloading to MEC server, however, when the channel state is poor, offloading the task to MEC server can not satisfy the latency constraint. Therefore, Fig. 4 gives the insight that, when the channel state is ideal, offloading the task to MEC server can guarantee the latency constraint, resulting in a low energy consumption. When the channle fading is severe, offloading to MEC still consume less energy than local offloading, however, due to the latency constraint, part of the task should be executed locally, which results in a higher energy consumption.

Figure 5 gives the performance comparison of the proposed EPOTPC method with two benchmark algorithms in terms of energy consumption through monte-carlo simulation. Local computing represents that all tasks are executed locally, and MEC offloading represents that all tasks are offloaded to the MEC server if the transmission time is above the maximum tolerant delay, a energy punishment factor will be added. It can be shown that, the total energy consumption of the proposed EPOTPC method is significantly lower than the two benchmark algorithms, which shows the effectiveness of our proposed method.

5 Conclusion

In this paper, we study the energy-efficient partial computation offloading and transmission power control of single mobile user under the Rayleigh fading channel. We first formulate the problem as an optimization problem with non-convex optimization objective and constraints. To solve the problem, we first reformulate the problem and analyze the property of each subproblem using block coordinate descent method to optimize each variable separately. Simulation results show that the proposed method has a better performance compared with two benchmark algorithms in terms of total energy consumption. As for future work, studying the joint optimization of computation and communication resource allocation and partial offloading in multi-user scenario will be challenging and interesting which is our future work direction.

References

1. Shi, W., Cao, J., Zhang, Q., Li, Y., Xu, L.: Edge computing: vision and challenges. IEEE Internet Things J. **3**(5), 637–646 (2016)
2. Mach, P., Becvar, Z.: Mobile edge computing: a survey on architecture and computation offloading. IEEE Commun. Surv. Tutor. **19**(3), 1628–1656 (2017, thirdquarter)
3. Mao, Y., You, C., Zhang, J., Huang, K., Letaief, K.B.: A survey on mobile edge computing: the communication perspective. IEEE Commun. Surv. Tutor. **19**(4), 2322–2358 (2017, Fourthquarter)

4. Dinh, T.Q., Tang, J., La, Q.D., Quek, T.Q.S.: Offloading in mobile edge computing: task allocation and computational frequency scaling. IEEE Trans. Commun. **65**(8), 3571–3584 (2017)
5. Chen, X., Jiao, L., Li, W., Fu, X.: Efficient multi-user computation offloading for mobile-edge cloud computing. IEEE/ACM Trans. Netw. **24**(5), 2795–2808 (2016)
6. Wang, M., Shi, S., Gu, S., et al.: Q-learning based computation offloading for multi-UAV-enabled cloud-edge computing networks. IET Commun. **14**(15), 2481–2490 (2020)
7. Zhang, W., Wen, Y., Guan, K., Kilper, D., Luo, H., Wu, D.O.: Energy-optimal mobile cloud computing under stochastic wireless channel. IEEE Trans. Wireless Commun. **12**(9), 4569–4581 (2013)
8. Tran, T.X., Pompili, D.: Joint task offloading and resource allocation for multi-server mobile-edge computing networks. IEEE Trans. Veh. Technol. **68**(1), 856–868 (2019)
9. Zhang, J., et al.: Energy-latency tradeoff for energy-aware offloading in mobile edge computing networks. IEEE Internet Things J. **5**(4), 2633–2645 (2018)
10. Wang, Y., Sheng, M., Wang, X., Wang, L., Li, J.: Mobile-edge computing: partial computation offloading using dynamic voltage scaling. IEEE Trans. Commun. **64**(10), 4268–4282 (2016)
11. Boyd, S., Vandenberghe, L.: Convex Optimization. Cambridge Univ. Press, Cambridge (2004)
12. Mao, Y., Zhang, J., Song, S.H., Letaief, K.B.: Stochastic joint radio and computational resource management for multi-user mobile-edge computing systems. IEEE Trans. Wireless Commun. **16**(9), 5994–6009 (2017)
13. Guo, K., Gao, R., Xia, W., Quek, T.Q.S.: Online learning based computation offloading in MEC systems with communication and computation dynamics. IEEE Trans. Commun. **69**(2), 1147–1162 (2021)
14. Xie, R., Tang, Q., Liang, C., Yu, F.R., Huang, T.: Dynamic computation offloading in IoT fog systems with imperfect channel-state information: a POMDP approach. IEEE Internet Things J. **8**(1), 345–356 (2021)