Wireless Communications and Mobile Computing

Edge Caching and Computing for Wireless Networks

Lead Guest Editor: Lisheng Fan
Guest Editors: Zhao Junhui, George K. Karagiannidis, and Rose Qingyang Hu

Edge Caching and Computing for Wireless Networks

Wireless Communications and Mobile Computing

Edge Caching and Computing for Wireless Networks

Lead Guest Editor: Lisheng Fan
Guest Editors: Zhao Junhui, George K. Karagiannidis, and Rose Qingyang Hu

Chief Editor

Editorial Board

Mauro Tortonesi, Italy
Federico Tramarin, Italy
Tran Trung Duy, Vietnam
Reza Monir Vaghefi, USA
Juan F. Valenzuela-Valdés, Spain
Lorenzo Vangelista, Italy
S Velliangiri, India
Quoc-Tuan Vien, United Kingdom
Enrico M. Vitucci, Italy
Yingjie Wang, China
Pengfei Wang, China
Huaqun Wang, China
Honggang Wang, USA
Ding Wang, China
Lifei Wei, China
Miaowen Wen, China
Dapeng Wu, China
Huaming Wu, China
liang wu, China
Ding Xu, China
Jie Yang, USA
Long Yang, China
YAN YAO, China
Qiang Ye, Canada
Ya-Ju Yu, Taiwan
Marat V. Yuldashev, P.O. Box 35 (Agora),
FIN-40014, Finland, Finland
Sherali Zeadally, USA
Jie Zhang, United Kingdom
Yin Zhang, China
Hong-Hai Zhang, USA
Jiliang Zhang, United Kingdom
Yushu Zhang, China
Lei Zhang, Spain
Wence Zhang, China
Xu Zheng, USA
Fuhui Zhou, USA
Meiling Zhu, United Kingdom
Zhengyu Zhu, China

Contents

Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 9756304, 2 pages
https://doi.org/10.1155/2022/9756304

Editorial

Edge Caching and Computing for Wireless Networks

Lisheng Fan,[1] Junhui Zhao,[2,3] George K. Karagiannidis,[4] and Rose Qingyang Hu[5]

[1] *School of Computer Science, Guangzhou University, Guangzhou, China*
[2] *School of Electronic and Information Engineering, Beijing Jiaotong University, Beijing 100044, China*
[3] *School of Information Engineering, East China Jiaotong University, Nanchang 330013, China*
[4] *Aristotle University of Thessaloniki, Thessaloniki 54636, Greece*
[5] *Department of Electrical and Computer Engineering, Utah State University, Logan, UT 84321, USA*

Correspondence should be addressed to Lisheng Fan; lsfan@gzhu.edu.cn

Received 8 March 2022; Accepted 8 March 2022; Published 21 April 2022

In the next-generation wireless networks, there have been explosively increasing wireless data services, such as video streaming, push media, mobile application download/updates, and mobile TV. These new services involve both intensive communication and computation, and to meet these requirements, edge caching and computing have been recently proposed.

Caching brings content closer to users by prefetching the content during off-peak times and hence can greatly reduce network congestion and improve the user-perceived experience. Moreover, edge computing can help alleviate the computation load on the central node, by offloading the computation tasks into edge nodes through wireless transmission links. In this special issue, we have invited a few papers to give insights on wireless caching and computing for wireless networks.

One paper of this special issue in Ref. [1] investigated range-angle localization of targets via double-pulse FDA-MIMO radar, where an accurate sparse recovery algorithm was proposed to enhance the performance of localization. In particular, the localization error was reduced significantly, and the system accuracy was enhanced obviously. Moreover, another paper of this special issue in Ref. [2] studied multinode collaborative computing offloading algorithm based on minimization of energy consumption, where the system performance in terms of energy consumption was minimized, which can help prolong the service time of the nodes in the wireless networks. In further, another paper of this issue in Ref. [3] studied the impact of imperfect channel estimation for cache-enabled UAV relaying networks, where the system diversity order caused by caching and multiple UAVs vanished due to the presence of channel estimation error.

In addition to the above works, there are some rest papers in this special issue on the application of artificial intelligence on the wireless caching and computing networks, as shown in Refs. [4–6]. In particular, deep reinforcement learning was proposed in these works, in order to provide an intelligent solution to the system resource allocation, such as caching allocation and offloading allocation, bandwidth allocation, and power allocation. Some other recent works on the intelligent algorithms, such as deep reinforcement learning [11, 13], deep learning [9, 12], federated learning [7, 8], and cache-enabled learning [10], can be viewed as an important extension to these works, which could help enhance the system performance of caching and computing networks furthermore.

Conflicts of Interest

The editors declare that they have no conflicts of interest regarding the publication of this Special Issue.

Lisheng Fan
Junhui Zhao
George K. Karagiannidis
Rose Qingyang Hu

References

[1] Q. Liu, X. Wang, L. Wan, M. Huang, and L. Sun, “An accurate sparse recovery algorithm for range-angle localization of targets via double-pulse FDA-MIMO radar,” *Wireless Communications and Mobile Computing*, vol. 2020, Article ID 6698446, 12 pages, 2020.

[2] D. Han, Y. Liu, and J. Ni, “Research on multinode collaborative computing offloading algorithm based on minimization of energy consumption,” *Wireless Communications and Mobile Computing*, vol. 2020, Article ID 8858298, 11 pages, 2020.

[3] D. Deng, Y. Rao, and F. Zhu, “Impact of hardware impairments with imperfect channel estimation for cache- enabled UAV relaying networks,” *Wireless Communications and Mobile Computing*, vol. 2020, Article ID 8891793, 12 pages, 2020.

[4] Z. Wan and Y. Li, “Deep reinforcement learning-based collaborative video caching and transcoding in clustered and intelligent edge B5G networks,” *Wireless Communications and Mobile Computing*, vol. 2020, 16 pages, 2020.

[5] L. Li, C. F. Kwong, Q. Liu, and J. Wang, “A smart cache content update policy based on deep reinforcement learning,” *Wireless Communications and Mobile Computing*, vol. 2020, Article ID 8836592, 11 pages, 2020.

[6] Z. Wang, T. Li, J. Ye, X. Yang, and K. Xiong, “AN-aided secure beamforming in SWIPT-aware mobile edge computing systems with cognitive radio,” *Wireless Communications and Mobile Computing*, vol. 2020, Article ID 8899314, 10 pages, 2020.

[7] S. Tang and L. Chen, “Computational intelligence and deep learning for next generation edge-enabled industrial IoT,” *IEEE Transactions Network Science and Engineering*, vol. 8, no. 1, pp. 114–125, 2021.

[8] L. He, K. He, L. Fan, Y. Deng, G. K. Karagiannidis, and A. Nallanathan, “Learning-based signal detection for MIMO systems with unknown noise statistics,” *IEEE Transactions on Communications*, vol. 69, no. 5, pp. 3025–3038, 2021.

[9] L. Chen and F. Zhou, “Physical-layer security on mobile edge computing for emerging cyber physical systems,” *Computer Communications*, vol. 178, no. 1, pp. 297–306, 2021.

[10] W. Zhou, L. Chen, S. Tang et al., “PSO based offloading strategy with PSO for mobile edge computing based on cache mechanism,” *Cluster Computing*, 2021.

[11] L. Chen, R. Zhao, and Z. Zhao, “Intelligent ubiquitous computing for future UAV-enabled MEC network systems,” *Cluster Computing*, 2021.

[12] S. Tang and L. Lai, “Battery-constrained federated edge learning in UAV enabled IoT for B5G/6G networks,” *Physical Communication*, vol. 47, p. 101381, 2021.

[13] K. He and Y. Deng, “Efficient memory-bounded optimal detection for GSM MIMO systems,” *IEEE Transactions on Communications*, vol. 70, no. 2, pp. 105–121, 2022.

Hindawi
Wireless Communications and Mobile Computing
Volume 2020, Article ID 6698446, 12 pages
https://doi.org/10.1155/2020/6698446

Research Article

An Accurate Sparse Recovery Algorithm for Range-Angle Localization of Targets via Double-Pulse FDA-MIMO Radar

Qi Liu,[1] **Xianpeng Wang**,[1] **Liangtian Wan**,[2] **Mengxing Huang**,[1] **and Lu Sun**[3]

[1] *State Key Laboratory of Marine Resource Utilization in South China Sea and School of Information and Communication Engineering, Hainan University, Haikou 570228, China*
[2] *Key Laboratory for Ubiquitous Network and Service Software of Liaoning Province, School of Software, Dalian University of Technology, Dalian 116620, China*
[3] *Department of Communication Engineering, Institute of Information Science Technology, Dalian Maritime University, 116026, China*

Correspondence should be addressed to Xianpeng Wang; wxpeng2016@hainu.edu.cn

Received 17 October 2020; Revised 16 November 2020; Accepted 29 November 2020; Published 16 December 2020

Academic Editor: Lisheng Fan

In this paper, a sparse recovery algorithm based on a double-pulse FDA-MIMO radar is proposed to jointly extract the angle and range estimates of targets. Firstly, the angle estimates of targets are calculated by transmitting a pulse with a zero frequency increment and employing the improved l_1-SVD method. Subsequently, the range estimates of targets are achieved by utilizing a pulse with a nonzero frequency increment. Specifically, after obtaining the angle estimates of targets, we perform dimensionality reduction processing on the overcomplete dictionary to achieve the automatically paired range and angle in range estimation. Grid partition will bring a heavy computational burden. Therefore, we adopt an iterative grid refinement method to alleviate the above limitation on parameter estimation and propose a new iteration criterion to improve the error between real parameters and their estimates to get a trade-off between the high-precision grid and the atomic correlation. Finally, the proposed algorithm is evaluated by providing the results of the Cramér-Rao lower bound (CRLB) and numerical root mean square error (RMSE).

1. Introduction

Target localization has been acting as a pivotal part in the field of array signal processing, which expects various applications in radar, navigation, and communication [1–4]. In recent years, multiple-input multiple-output (MIMO) radar [5, 6] has attracted widespread consideration in target localization due to many potential merits [7], where multiple antennas are utilized to transmit different waveforms at the same time and simultaneously receive reflected signals. Compared with the phased array radar, MIMO radar can obtain enhanced spatial resolution, improved estimation performance, and increased degrees of freedom (DOF) [8–10] by effectively utilizing space diversity. However, MIMO radar cannot obtain the essential range estimates of targets. FDA-MIMO [11–14] radar, as a combination of frequency diverse array (FDA) radar [15–17] and MIMO radar, has a small frequency increment in adjacent transmitting array antennas to achieve the joint estimation of the angle and range [18].

Nowadays, the traditional DOA estimation algorithms have been applied for the joint angle-range estimation of FDA-MIMO radar, such as the estimation of signal parameters via rotational invariance techniques (ESPRIT) [19], unitary ESPRIT (U-ESPRIT) [20], and two-dimensional multiple signal classification (2D-MUSIC) [21]. However, the aforementioned algorithms based on subspace decomposition usually require a large number of snapshots and encounter performance degradation in the case of highly correlated targets.

The compressed sensing technique has attracted extensive attention in sparse signal reconstruction to deal with the above limitation of the subspace-based algorithms. The sparse signal recovery (SSR) algorithms [22–24] mainly estimate target parameters by constructing a sparse signal model and reconstructing the spatial spectrum. In an environment with a low signal-to-noise ratio (SNR) or a small number of snapshots, the SSR algorithm outperforms the subspace-based method in parameter estimation performance [25, 26]. In the past several years, some SSR algorithms have been presented, such as the sparse Bayesian learning (SBL) algorithm [27, 28], l_1-norm singular value decomposition (SVD) algorithm [29], and l_1-norm sparse representation of array covariance vector (SRACV) algorithm [30]. Nevertheless, they suffer from a complex two-dimensional overcomplete dictionary, which will bring a heavy computational burden.

In this paper, a sparse recovery algorithm is proposed based on a double-pulse FDA-MIMO radar. We extend the double-pulse concept of FDA radar in [31] to FDA-MIMO radar to solve the high-complexity problem of sparse recovery algorithms and simultaneously improve the parameter estimation performance of FDA-MIMO radar. Firstly, the angle estimates of targets are calculated by utilizing a pulse with a zero frequency increment and employing the improved l_1-SVD method. Subsequently, the range estimates of targets are achieved by transmitting a pulse with a nonzero frequency increment. Specifically, after obtaining the angle estimates of targets, we deleted the unnecessary elements in the overcomplete dictionary to reduce its dimensionality during the range estimation. Therefore, this algorithm not only decouples the angle and range of FDA-MIMO radar but also reduces the dimension of the overcomplete dictionary. Grid partition will bring the problem of the heavy computational burden. As a result, we utilize an iterative grid refinement method to overcome the adverse effects caused by the grid partition on parameter estimation. Furthermore, we propose a new iteration criterion to improve the error between real parameters and their estimates to get a trade-off between the high-precision grid and the atomic correlation, so the proposed algorithm can achieve better target localization performance with FDA-MIMO radar as compared with the subspace-based algorithm. Finally, we derive the CRLB for the target parameter of the double-pulse FDA-MIMO radar. Numerical simulation verifies the superior performance of the proposed algorithm.

Notation. Capital bold letters and lowercase bold letters represent matrices and vectors, respectively. $(\cdot)^*$, $(\cdot)^{-1}$, and $(\cdot)^T$ stand for conjugate, inverse, and transpose operations, respectively. $\odot$ represents the Hadamard product, and $\otimes$ denotes the Kronecker product. $\|\bullet\|_1$ and $\|\bullet\|_2$ denote l_1-norm and l_2-norm, respectively. $\mathbb{C}^{P\times Q}$ denotes a $P\times Q$ complex matrix set.

2. Signal Model

As shown in Figure 1, a monostatic double-pulse FDA-MIMO radar that consists of uniform linear arrays (ULAs) with interelement spacing $d=\lambda/2$ is considered, where the transmitter has M antennas and the receiver has N antennas. The first antenna in the transmitter is treated as the reference point. Considering the linearly increasing frequency increments, the carrier frequency at the m-th transmitter antenna is

$$f_m = f_1 + (m-1)\Delta f, \quad m = 1, 2, \cdots, M, \tag{1}$$

where Δf denotes the frequency increment and f_1 stands for the carrier frequency of the first antenna in the transmitter, where $\Delta f \ll f_1$.

Suppose the narrowband signal emitted by the m-th antenna is

$$s_m(t) = \phi_m(t)e^{j2\pi f_m t}, \quad m = 1, 2, \cdots, M, 0 \le t \le T, \tag{2}$$

where T is the duration of the radar pulse and $\phi_m(t)$ is the m-th baseband waveform which follows that

$$\int_0^T \phi_m(t)\phi_n^*(t-\tau)e^{j2\pi(m-n)\Delta f t}dt = \begin{cases} 1, & m = n, \tau = 0, \\ 0, & m \ne n, \forall\tau, \end{cases} \tag{3}$$

where τ represents the time delay.

Assume that there are K far-field targets in the far-field whose ranges are much larger than the aperture of FDA-MIMO radar. Subsequently, the signal received by the n-th antenna in the receiver and transmitted by the m-th antenna in the transmitter can be represented by

$$x_{m,n}(t) = \sum_{k=1}^{K} \phi_m(t-\tau(m,n,\theta_k,r_k))e^{j2\pi f_m(t-\tau(m,n,\theta_k,r_k))}, \tag{4}$$

where $\tau(m,n,\theta_k,r_k)$ represents the delay between the m-th antenna in the transmitter and the n-th antenna in the receiver, which is expressed as

$$\tau(m,n,\theta_k,r_k) = \frac{2r_k}{c} - \frac{(m-1)d_t \sin(\theta_k)}{c} - \frac{(n-1)d_r \sin(\theta_k)}{c}, \tag{5}$$

where r_k and θ_k are the range and angle of the k-th target. d_t and d_r are the interval between transmitter antennas and receiver antennas, respectively. c is the speed of light.

The outputs of the received data after the matched filter (MF) can be expressed as [14]

$$\mathbf{x}(l) = \mathbf{A}\mathbf{s}(l) + \mathbf{n}(l), \tag{6}$$

where $\mathbf{s}(l) = [s_1(l), s_2(l), \cdots, s_K(l)]^T \in \mathbb{C}^{K\times 1}$ is a signal vector. $\mathbf{n}(l)$ represents the noise vector. $\mathbf{A} = [\bar{\mathbf{a}}(\theta_1, r_1), \bar{\mathbf{a}}(\theta_2, r_2), \cdots, \bar{\mathbf{a}}(\theta_K, r_K)] \in \mathbb{C}^{MN\times K}$ is a joint steering vector matrix, and $\bar{\mathbf{a}}(\theta_k, r_k) = \mathbf{a}_r(\theta_k) \otimes \mathbf{a}_t(\theta_k, r_k)$ with $k = 1, 2, \cdots, K$. The steering vectors of the receiver and transmitter can be defined by [13]

$$\begin{aligned} \mathbf{a}_r(\theta_k) &= \left[1, e^{j2\pi(d/\lambda)\sin\theta_k}, \cdots, e^{j2\pi(d/\lambda)(N-1)\sin\theta_k}\right]^T, \\ \mathbf{a}_t(\theta_k, r_k) &= \left[1, e^{-j4\pi(\Delta f/c)r_k}, \cdots, e^{-j4\pi(\Delta f/c)(M-1)r_k}\right]^T \odot \\ &\quad \cdot \left[1, e^{j2\pi(d/\lambda)\sin(\theta_k)}, \cdots, e^{j2\pi(d/\lambda)(M-1)\sin\theta_k}\right]^T, \end{aligned} \tag{7}$$

where $\mathbf{a}_r(\theta_k) \in \mathbb{C}^{N\times 1}$ and $\mathbf{a}_t(\theta_k, r_k) \in \mathbb{C}^{M\times 1}$.

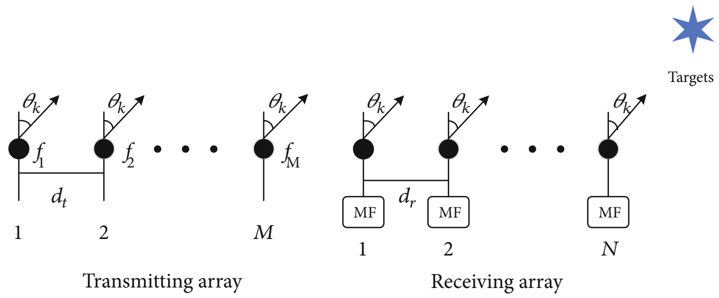

Figure 1: Simplified diagram of a monostatic double-pulse FDA-MIMO radar.

The output of the MF by collecting L snapshots can be described as

$$\mathbf{X} = \mathbf{A}\mathbf{S} + \mathbf{N}, \tag{8}$$

where $\mathbf{X} = [\mathbf{x}(1), \mathbf{x}(2), \cdots, \mathbf{x}(L)] \in \mathbb{C}^{MN\times L}$, $\mathbf{S} = [\mathbf{s}(1), \mathbf{s}(2), \cdots, \mathbf{s}(L)] \in \mathbb{C}^{K\times L}$, and $\mathbf{N} = [\mathbf{n}(1), \mathbf{n}(2), \cdots, \mathbf{n}(L)] \in \mathbb{C}^{MN\times L}$.

3. Range and Angle Estimation for Monostatic Double-Pulse FDA-MIMO Radar

In this section, we propose a target localization algorithm with a double-pulse FDA-MIMO radar based on iterative grid refinement to alleviate the problem that grid partition brings about, a heavy computational burden and correlation. Firstly, we decouple the range and angle parameters of the FDA-MIMO radar with two pulses. Then, the improved l_1-SVD method is utilized to estimate the angle and range of the target.

3.1. Angle Estimation for FDA-MIMO Radar. The angle estimates of targets are calculated by transmitting a pulse with a zero frequency increment and avoiding the range parameter. According to (8), the output of the FDA-MIMO radar after MF can be reconstructed by [32]

$$\mathbf{X}_a = \mathbf{A}_a\mathbf{S}_a + \mathbf{N}_a, \tag{9}$$

where $\mathbf{X}_a = [\mathbf{x}_a(1), \mathbf{x}_a(2), \cdots, \mathbf{x}_a(L)] \in \mathbb{C}^{MN\times L}$. $\mathbf{S}_a = [\mathbf{s}_a(1), \mathbf{s}_a(2), \cdots, s_a(L)] \in \mathbb{C}^{K\times L}$ is a transmit signal matrix. $\mathbf{N}_a = [\mathbf{n}_a(1), \mathbf{n}_a(2), \cdots, \mathbf{n}_a(L)] \in \mathbb{C}^{MN\times L}$ stands for the noise matrix. $\mathbf{A}_a = [\mathbf{a}_a(\theta_1), \mathbf{a}_a(\theta_2), \cdots, \mathbf{a}_a(\theta_K)]$ is a $MN \times K$ steering vector matrix, and $\mathbf{a}_a(\theta_k) = \mathbf{a}_{ar}(\theta_k) \otimes \mathbf{a}_{at}(\theta_k)$ with $k = 1, 2, \cdots, K$. Then, the steering vectors of the receiver and transmitter of the k-th target can be defined by

$$\begin{aligned}\mathbf{a}_{at}(\theta_k) &= \left[1, e^{j2\pi(d/\lambda)\sin\theta_k}, \cdots, e^{j2\pi(d/\lambda)(M-1)\sin\theta_k}\right]^T,\\ \mathbf{a}_{ar}(\theta_k) &= \left[1, e^{j2\pi(d/\lambda)\sin\theta_k}, \cdots, e^{j2\pi(d/\lambda)(N-1)\sin\theta_k}\right]^T.\end{aligned} \tag{10}$$

We can utilize the sparse recovery method to achieve angle estimates. An overcomplete set of angles $\bar{\theta} = [\bar{\theta}_1, \bar{\theta}_2, \cdots, \bar{\theta}_P]$ is established by sampling the spatial domain range $[-\pi/2, \pi/2]$ uniformly, where $P \gg K$ is the number of grid points. Then, we need to reformulate the signal model (9) into the sparse signal model as

$$\mathbf{X}_a = \bar{\mathbf{A}}_a\bar{\mathbf{S}}_a + \mathbf{N}_a, \tag{11}$$

where $\bar{\mathbf{S}}_a = [\bar{\mathbf{s}}_a(1), \bar{\mathbf{s}}_a(2), \cdots, \bar{\mathbf{s}}_a(L)] \in \mathbb{C}^{P\times L}$ is a sparse matrix. $\bar{\mathbf{A}}_a = [\bar{\mathbf{a}}_a(\bar{\theta}_1), \bar{\mathbf{a}}_a(\bar{\theta}_2), \cdots, \bar{\mathbf{a}}_a(\bar{\theta}_P)] \in \mathbb{C}^{MN\times P}$ is an overcomplete dictionary, and $\bar{\mathbf{a}}_a(\bar{\theta}_p) = \bar{\mathbf{a}}_{ar}(\bar{\theta}_p) \otimes \bar{\mathbf{a}}_{at}(\bar{\theta}_p)$ with $p = 1, 2, \cdots, P$. $\bar{\mathbf{a}}_{at}(\bar{\theta}_p)$ and $\bar{\mathbf{a}}_{ar}(\bar{\theta}_p)$ can be denoted as

$$\begin{aligned}\bar{\mathbf{a}}_{at}\left(\bar{\theta}_p\right) &= \left[1, e^{j2\pi(d/\lambda)\sin\bar{\theta}_p}, \cdots, e^{j2\pi(d/\lambda)(M-1)\sin\bar{\theta}_p}\right]^T,\\ \bar{\mathbf{a}}_{ar}\left(\bar{\theta}_p\right) &= \left[1, e^{j2\pi(d/\lambda)\sin\bar{\theta}_p}, \cdots, e^{j2\pi(d/\lambda)(N-1)\sin\bar{\theta}_p}\right]^T.\end{aligned} \tag{12}$$

Then, we utilize the l_1-SVD method to estimate the angle. The SVD result of the matrix $\mathbf{X}_a$ can be represented by [29]

$$\mathbf{X}_a = \mathbf{U}_a\mathbf{Q}_a{\mathbf{V}_a}^T, \tag{13}$$

where $\mathbf{U}_a \in \mathbb{C}^{MN\times MN}$ and $\mathbf{V}_a \in \mathbb{C}^{L\times L}$ are orthogonal matrices and $\mathbf{Q}_a \in \mathbb{C}^{MN\times L}$ is a block matrix. We get a $MN \times K$ matrix $\mathbf{X}_{aSV}$, which contains nearly all the signal power, $\mathbf{X}_{aSV} = \mathbf{U}_a \mathbf{Q}_a\mathbf{D}_K = \mathbf{X}_a\mathbf{V}_a\mathbf{D}_K$. $\mathbf{D}_K = [\mathbf{I}_K, 0] \in \mathbb{C}^{L\times K}$ where $\mathbf{I}_K \in \mathbb{C}^{K\times K}$ is an identity matrix, and $0 \in \mathbb{C}^{(T-K)\times K}$ is a zero matrix. Moreover, suppose $\bar{\mathbf{S}}_{aSV} = \bar{\mathbf{S}}_a\mathbf{V}_a\mathbf{D}_K$ and $\mathbf{N}_{aSV} = \mathbf{N}\mathbf{V}_a\mathbf{D}_K$; we can derive $\mathbf{X}_{aSV}$ as

$$\mathbf{X}_{aSV} = \bar{\mathbf{A}}_a\bar{\mathbf{S}}_{aSV} + \mathbf{N}_{aSV}. \tag{14}$$

$\tilde{s}_{ai}^{(l_2)} = \sqrt{\sum_{k=1}^{K}(s_{ai}^{aSV}(k))^2}$, where $i = 1, 2, \cdots, P$. The sparsity vector $\tilde{\mathbf{s}}_a^{(l_2)}$ corresponds to the space spectrum, which can be calculated by the following constraint optimization problem:

$$\min \left\|\tilde{\mathbf{s}}_a^{(l_2)}\right\|_1, \text{subject to } \left\|\mathbf{X}_{aSV} - \bar{\mathbf{A}}_a\bar{\mathbf{S}}_{aSV}\right\|_2^2 \le \eta_a^2, \tag{15}$$

where η_a denotes the regularization parameter [29] to balance the mismatch degree of the model and the sparsity. According to the chi-squared distribution, the upper bound of $\mathbf{N}_{aSV}$ can be calculated by the regularization parameter η_a [33] with a high probability of 99.9%. Finally, we utilize the second-order cone (SOC) programming package, such as CVX, to solve the optimization problem in (15). Based on $\tilde{\mathbf{s}}_a^{(l_2)}$, the one-dimensional spectral peak search can be established where the angle estimates correspond to K maximum peaks.

3.2. Range Estimation for FDA-MIMO Radar. The range estimates of targets are calculated by transmitting a pulse with a nonzero frequency increment. The output of the MF by collecting L snapshots can be expressed as (8).

We assume that angle estimates obtained from subsection A are $\bar{\theta} = [\bar{\theta}_1, \bar{\theta}_2, \cdots, \bar{\theta}_K]^T$. To get the range estimates, we utilize the sparse recovery method. An overcomplete set of ranges $\bar{\mathbf{r}} = [\bar{r}_1, \bar{r}_2, \cdots, \bar{r}_W]$ is established by sampling the spatial domain range $[0, c/2\Delta f]$ uniformly, where W is the number of grid points and $c/2\Delta f$ denotes the maximum unambiguous range [34]. We stack the range complete set corresponding to K angles into a large row vector $\tilde{\mathbf{r}} = [\tilde{r}_{(\bar{\theta}_1,1)}, \tilde{r}_{(\bar{\theta}_1,2)}, \cdots, \tilde{r}_{(\bar{\theta}_1, W)}, \tilde{r}_{(\bar{\theta}_2,1)}, \cdots, \tilde{r}_{(\bar{\theta}_K,W-1)}, \tilde{r}_{(\bar{\theta}_K,W)}]$ to obtain automatically paired range and angle estimates. Then, we need to construct the signal model of (8) into a sparse signal model as

$$\mathbf{X} = \bar{\mathbf{A}}_r \bar{\mathbf{S}}_r + \mathbf{N}, \tag{16}$$

where $\bar{\mathbf{S}}_r = [\bar{\mathbf{s}}_r(1), \bar{\mathbf{s}}_r(2), \cdots, \bar{\mathbf{s}}_r(L)] \in \mathbb{C}^{KW \times L}$ denotes a sparse matrix. $\bar{\mathbf{A}}_r = [\bar{\mathbf{a}}_r(\bar{\theta}_1, \bar{r}_1), \bar{\mathbf{a}}_r(\bar{\theta}_1, \bar{r}_2), \cdots, \bar{\mathbf{a}}_r(\bar{\theta}_K, \bar{r}_{W-1}), \bar{\mathbf{a}}_r(\bar{\theta}_K, \bar{r}_W)] \in \mathbb{C}^{MN \times KW}$ is a known overcomplete dictionary, and $\bar{\mathbf{a}}_r(\bar{\theta}_k, \bar{r}_w) = \bar{\mathbf{a}}_{rr}(\bar{\theta}_k) \otimes \bar{\mathbf{a}}_{rt}(\bar{\theta}_k, \bar{r}_w)$ with $k = 1, 2, \cdots, K$ and $w = 1, 2, \cdots, W$. $\bar{a}_{rt}(\bar{\theta}_k, \bar{r}_w)$ and $\bar{a}_{rr}(\bar{\theta}_k)$ can be defined as

$$\begin{aligned}\bar{\mathbf{a}}_{rt}\left(\bar{\theta}_k, \bar{r}_w\right) &= \left[1, e^{-j4\pi(\Delta f/c)\bar{r}_w}, \cdots, e^{-j4\pi(M-1)(\Delta f/c)\bar{r}_w}\right]^T \mathrm{e} \\ &\quad \cdot \left[1, e^{j2\pi(d/\lambda)\sin\left(\bar{\theta}_k\right)}, \cdots, e^{j2\pi(d/\lambda)\sin\left(\bar{\theta}_k\right)}\right]^T, \\ \bar{\mathbf{a}}_{rr}\left(\bar{\theta}_k\right) &= \left[1, e^{j2\pi(d/\lambda)\sin\left(\bar{\theta}_k\right)}, \cdots, e^{j2\pi(d/\lambda)(N-1)\sin\left(\bar{\theta}_k\right)}\right]^T.\end{aligned} \tag{17}$$

Then, we utilize the l_1-SVD method to estimate the range. The SVD result of the matrix $\mathbf{X}$ can be expressed as [29]

$$\mathbf{X} = \mathbf{U}_r \mathbf{Q}_r \mathbf{V}_r^T, \tag{18}$$

where $\mathbf{U}_r \in \mathbb{C}^{MN \times MN}$ and $\mathbf{V}_r \in \mathbb{C}^{L \times L}$ are orthogonal matrices and $\mathbf{Q}_r \in \mathbb{C}^{MN \times L}$ is a block matrix. We can get a $MN \times K$ matrix $\mathbf{X}_{rSV}$, which contains nearly all the signal power, $\mathbf{X}_{rSV} = \mathbf{U}_r \mathbf{Q}_r \mathbf{D}_K = \mathbf{X}_r \mathbf{V}_r \mathbf{D}_K$. Besides, suppose $\bar{\mathbf{S}}_{rSV} = \bar{\mathbf{S}}_r \mathbf{V}_r \mathbf{D}_K$ and $\mathbf{N}_{rSV} = \mathbf{N}\mathbf{V}_r \mathbf{D}_K$; we can derive the expression for $\mathbf{X}_{rSV}$ as

$$\mathbf{X}_{rSV} = \bar{\mathbf{A}}_r \bar{\mathbf{S}}_{rSV} + \mathbf{N}_{rSV}. \tag{19}$$

$\tilde{s}_{ri}^{(l_2)} = \sqrt{\sum_{k=1}^{K} (s_{ri}^{rSV}(k))^2}$, where $i = 1, 2, \cdots, KW$. The sparsity of $\tilde{\mathbf{s}}_r^{(l_2)}$ corresponds to the sparsity of the space spectrum, which can be calculated by the following constraint optimization problem:

$$\min \left\| \tilde{\mathbf{s}}_r^{(l_2)} \right\|_1, \text{subject to } \left\| \mathbf{X}_{rSV} - \bar{\mathbf{A}}_r \bar{\mathbf{S}}_{rSV} \right\|_2^2 \le \eta_r^2. \tag{20}$$

According to the chi-squared distribution, the upper bound of the $\mathbf{N}_{rSV}$ power can be calculated as the regularization parameter η_r [33] with a high probability of 99.9%. Finally, we utilize the SOC programming package, such as CVX, to solve the optimization problem in (20). Based on $\tilde{\mathbf{s}}_r^{(l_2)}$, the one-dimensional spectral peak search can be established where the range estimates correspond to K maximum peaks.

3.3. Grid Refinement. Since it is impossible that all parameter estimates fall on the grid points, the refining operation for the grid is required, which will bring high computational complexity and produce highly correlated atoms. To tackle the problems, we propose an improved iterative grid refinement algorithm. For example, the algorithm steps of range estimation are given as follows:

(1) Set refinement times $o = 1$. A simple grid $\mathbf{r}^{(o)}$ is constructed by discretizing the interval between 0 and $c/2\Delta f$ to estimate the target parameters. The grid spacing is B_o

(2) Use the proposed method in subsection B to get $\tilde{\mathbf{r}}_o = [r_1, r_2, \cdots, r_K]$ and $F_o^r = |\|\mathbf{R}\|_2 - \|\mathbf{R}_o^r\|_2|$. Then, set $o = o + 1$

(3) According to the range estimates in step 2, a new grid $\mathbf{r}^{(o)}$ composed of K subgrids is constructed, where K subgrids are $\tilde{r}_1, \tilde{r}_2, \cdots, \tilde{r}_K$ and $\tilde{r}_i$ is a grid established by sampling the spatial domain range with $[r_i - B_o, r_i + B_o]$ uniformly, and $i = 1, 2, \cdots, K$. Set grid spacing $B_o = B_{o-1}/100$

(4) Return to step 2 until $F_{o-1}^r < F_o^r$, the final range estimates can be obtained as $\tilde{\mathbf{r}}_{o-1}$

In the proposed algorithm, we use F to improve the error between real parameters and their estimates to get a trade-off between the high-precision grid and the atomic correlation. The detailed steps of the proposed sparse recovery algorithm based on a double-pulse FDA-MIMO radar are summarized in Algorithm 1.

Remark 2. The main computational complexity of the proposed algorithm is the singular value decomposition and the SOC programming problem. The singular value decomposition of $\mathbf{X}$ and $\mathbf{X}_a$ requires altogether $O\{2U(MN)^2 + 2MNUL^2\}$ flops, and it takes $O\{(K^2W)^3 + (PK)^3 + 2(U-1)(K^2G)^3\}$ flops to solve the above SOC programming problem, where U denotes the number of iterations and G represents the number of refinement grid points for each target. Compared with the U-ESPRIT algorithm [20], the proposed algorithm requires more computation. However, this algorithm has outstanding advantages, which can not only adapt

```
(1) The FDA-MIMO radar transmits a pulse with a zero frequency increment to obtain the received signal $\mathbf{X}_a$
(2) The sparse vector $\tilde{\mathbf{s}}_a^{(l_2)}$ is obtained by CVX optimization of (15). Angle estimates are realized by searching $K$ maximum values through a one-dimensional spectrum of $\tilde{\mathbf{s}}_a^{(l_2)}$
(3) Use the method in subsection $C$ to optimize the angle estimates in step 2. Then, the refined angle estimates can be received as $[\theta_1, \theta_2, \cdots, \theta_K]$
(4) The FDA-MIMO radar transmits a pulse with a nonzero frequency increment to obtain the received signal $\boldsymbol{X}$
(5) The sparse vector $\tilde{\mathbf{s}}_r^{(l_2)}$ is obtained by CVX optimization of (20). Range estimates are realized by searching $K$ maximum values through a one-dimensional spectrum of $\tilde{\mathbf{s}}_r^{(l_2)}$
(6) Use the method in subsection $C$ to optimize the range estimates in step 5. Then, the range estimates can be obtained as $[r_1, r_2, \cdots, r_K]$, and get automatically paired range and angle estimates $(\theta_k, r_k)$, for $k = 1, 2, \cdots, K$
```

Algorithm 1: An accurate sparse recovery algorithm for double-pulse FDA-MIMO radar.

to the scene of insufficient snapshots and high target correlation but also provide higher precision and resolution.

Remark 3. We assume that angle estimates obtained from subsection A are $\overline{\theta}_1, \overline{\theta}_2, \cdots, \overline{\theta}_K$, respectively. Therefore, we can construct a simplified overcomplete dictionary containing range and angle information via adding a sparse grid of the range dimension. By solving the SOC programming problem in (20), a $KW \times 1$ sparse vector $\tilde{\mathbf{s}}_r^{(l_2)}$ can be obtained, where the first W elements represent the range estimates of the target with angle $\overline{\theta}_1$, the $W+1$ to $2W$ represent the range estimates of the target with angle $\overline{\theta}_2$, and the last W elements represent the range estimates of the target with angle $\overline{\theta}_K$. Hence, the corresponding angle can be found through the position of the element in the sparse space spectrum $\tilde{\mathbf{s}}_r^{(l_2)}$ to obtain automatically paired range and angle estimates.

Remark 4. In this paper, we utilize F to improve the error between real parameters and their estimates. F can be written as F^a in the angle estimates, which is defined by $|\|\mathbf{R}_a\|_2 - \|\mathbf{R}^a\|_2|$, where $\mathbf{R}_a = \mathbf{X}_a\mathbf{X}_a^H$ and $\mathbf{R}^a = (\mathbf{A}^a\mathbf{S}_a)(\mathbf{A}^a\mathbf{S}_a)^H$. $\mathbf{A}^a = [\mathbf{a}_a(\overline{\theta}_1), \mathbf{a}_a(\overline{\theta}_2), \cdots, \mathbf{a}_a(\overline{\theta}_K)]$ is a new steering vector matrix constructed by θ. $\theta = [\overline{\theta}_1, \overline{\theta}_2, \cdots, \overline{\theta}_K]^T$ is the angle estimate for each iteration. F can be written as F^r in the range estimates, which is defined by $|\|\mathbf{R}\|_2 - \|\mathbf{R}^r\|_2|$, where $\mathbf{R} = \mathbf{X}\mathbf{X}^H$ and $\mathbf{R}^r = (\mathbf{A}^r\mathbf{S})(\mathbf{A}^r\mathbf{S})^H$. $\mathbf{A}^r = [\bar{\mathbf{a}}(\widehat{\theta}_1, \widehat{r}_1), \bar{\mathbf{a}}(\widehat{\theta}_2, \widehat{r}_2), \cdots, \bar{\mathbf{a}}(\widehat{\theta}_K, \widehat{r}_K)]$ is a new steering vector matrix constructed by $\widehat{\theta}$ and $\widehat{\mathbf{r}}$. $\widehat{\theta} = [\theta\wedge_1, \theta\wedge_2, \cdots, \theta\wedge_K]^T$ is the angle estimate in advance, and $\widehat{\mathbf{r}} = [r\wedge_1, r\wedge_2, \cdots, r\wedge_K]^T$ denotes the range estimate for each iteration.

4. CRLB Analysis

In this section, we derive the CRLB results with regard to the angle and range. According to (8) and (9), we can rewrite the signal model as vector

$$\mathbf{x}_{\text{new}} = \mathbf{a}_{\text{new}}(\theta_k, r_k) \cdot s + \mathbf{v}, \tag{21}$$

where the $\mathbf{v}$ is the normalized Gaussian noise with zero mean and unit variance $\mathbf{I}$. $\mathbf{a}_{\text{new}}(\theta_{k,} r_k) \in \mathbb{C}^{2MN\times 1}$ is the equivalent steering vector and can be presented by

$$\mathbf{a}_{\text{new}}(\theta_k, r_k) = \left[\mathbf{a}(\theta_k, r_k)|_{\Delta f=0}, \mathbf{a}(\theta_k, r_k)\right]^T, \tag{22}$$

where

$$\begin{aligned} \mathbf{a}(\theta_k, r_k)\big|_{\Delta f=0} &= \mathbf{a}_{ar}(\theta_k) \otimes \mathbf{a}_{at}(\theta_k), \\ \mathbf{a}(\theta_k, r_k) &= \mathbf{a}_r(\theta_k) \otimes \mathbf{a}_t(\theta_k, r_k). \end{aligned} \tag{23}$$

Assuming that there are K targets, the Fisher information matrix (FIM) is [35]

$$\mathbf{J} = \begin{bmatrix} J_{\theta_k\theta_k} & J_{\theta_k r_k} \\ J_{r_k\theta_k} & J_{r_k r_k} \\ & \end{bmatrix}, \tag{24}$$

where $k = 1, 2, \cdots, K$.

$$\begin{aligned} J_{\theta_k\theta_k} &= \frac{2L}{\sigma^2}\,\mathrm{Re}\left[\left(\frac{d\varepsilon}{d\theta_k}\right)^H \Gamma^{-1}\left(\frac{d\varepsilon}{d\theta_k}\right)\right], \\ J_{\theta_k r_k} &= \frac{2L}{\sigma^2}\,\mathrm{Re}\left[\left(\frac{d\varepsilon}{d\theta_k}\right)^H \Gamma^{-1}\left(\frac{d\varepsilon}{dr_k}\right)\right], \\ J_{r_k\theta_k} &= \frac{2L}{\sigma^2}\,\mathrm{Re}\left[\left(\frac{d\varepsilon}{dr_k}\right)^H \Gamma^{-1}\left(\frac{d\varepsilon}{d\theta_k}\right)\right], \\ J_{r_k r_k} &= \frac{2L}{\sigma^2}\,\mathrm{Re}\left[\left(\frac{d\varepsilon}{dr_k}\right)^H \Gamma^{-1}\left(\frac{d\varepsilon}{dr_k}\right)\right], \end{aligned} \tag{25}$$

where σ^2 represents the noise power, $\varepsilon = a_{\text{new}}(\theta_k, r_k)$, and

$\Gamma = \mathbf{I}$. $d\varepsilon/d\theta_k$ and $d\varepsilon/dr_k$ can be expressed as

$$\frac{d\varepsilon}{d\theta_k} = \left[\frac{\partial a_{ar}(\theta_k)}{\partial \theta_k} \otimes a_{ar}(\theta_k) + a_{ar}(\theta_k) \otimes \frac{\partial a_{at}(\theta_k)}{\partial \theta_k}, \frac{\partial a_r(\theta_k)}{\partial \theta_k} \otimes a_t(\theta_k, r_k) + a_r(\theta_k) \otimes \frac{\partial a_t(\theta_k, r_k)}{\partial \theta_k}\right],$$

$$\frac{d\varepsilon}{dr_k} = \left[0, a_r(\theta_k) \otimes \frac{\partial a_t(\theta_k, r_k)}{\partial r_k}\right],$$

$$\frac{\partial a_{ar}(\theta_k)}{a\theta_k} = j2\pi \frac{d}{\lambda} \cos(\theta_k) \begin{bmatrix} 0 & & \\ & \ddots & \\ & & N-1 \end{bmatrix} a_{ar}(\theta_k),$$

$$\frac{\partial a_{at}(\theta_k)}{a\theta_k} = j2\pi \frac{d}{\lambda} \cos(\theta_k) \begin{bmatrix} 0 & & \\ & \ddots & \\ & & M-1 \end{bmatrix} a_{at}(\theta_k),$$

$$\frac{\partial a_r(\theta_k)}{a\theta_k} = j2\pi \frac{d}{\lambda} \cos(\theta_k) \begin{bmatrix} 0 & & \\ & \ddots & \\ & & N-1 \end{bmatrix} a_r(\theta_k),$$

$$\frac{\partial a_t(\theta_k, r_k)}{a\theta_k} = j2\pi \frac{d \cos(\theta_k)}{\lambda} \begin{bmatrix} 0 & & \\ & \ddots & \\ & & M-1 \end{bmatrix} a_t(\theta_k, r_k),$$

$$\frac{\partial a_t(\theta_k, r_k)}{ar_k} = -j4\pi \frac{\Delta f}{c} \begin{bmatrix} 0 & & \\ & \ddots & \\ & & M-1 \end{bmatrix} a_t(\theta_k, r_k). \tag{26}$$

The CRLB for the range and angle can be expressed as

$$\mathrm{CRLB}_{\theta_k} = \left[\mathbf{J}^{-1}\right]_{1,1}, \quad \mathrm{CRLB}_{r_k} = \left[\mathbf{J}^{-1}\right]_{2,2}. \tag{27}$$

5. Numerical Simulation Results

In this section, we demonstrate the superiority of the proposed algorithm via simulation, where $M = N = 8$ and $d = \lambda/2$. The carrier frequency f_0 is 10 GHz, the frequency increment Δf is 0 kHz in angle estimation, and the frequency increment Δf is 1 kHz in range estimation.

5.1. Simple Process of Target Estimation. Suppose $K = 3$ narrowband targets with angles $\theta_1 = -20°$ and $\theta_2 = \theta_3 = 40°$ and ranges $r_1 = 21000$m, $r_2 = 34000$m, and $r_3 = 54000$m, respectively. We set SNR = 10 dB, and the number of snapshots is

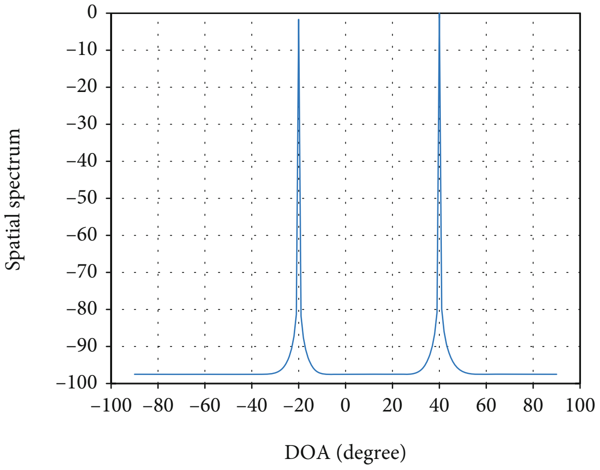

Figure 2: The spatial spectrum of angle estimation.

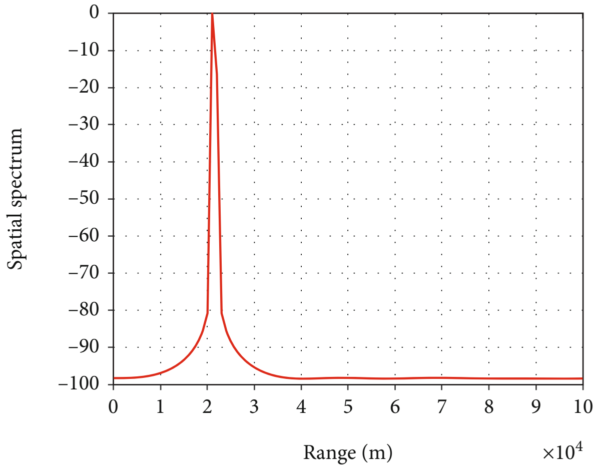

Figure 3: Range estimation of the target with angle θ_1.

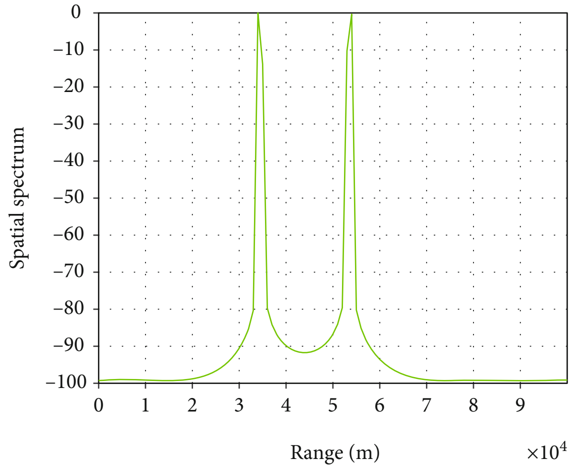

Figure 4: Range estimation of the target with angle θ_2.

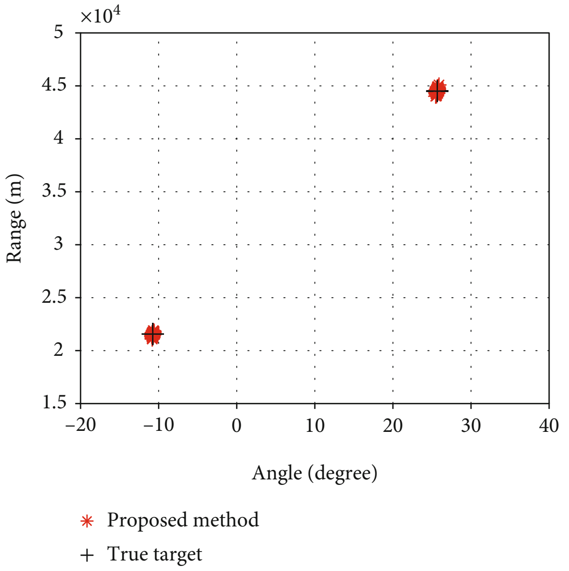

Figure 5: Range and angle estimation results using the proposed algorithm.

50. Figure 2 depicts the spatial spectrum of the sparse vector $\tilde{\mathbf{s}}_a^{(l_2)}$ for angle estimates in the proposed algorithm. Since there are two targets from the same direction, it is necessary to extract the range estimates of targets. Figures 3 and 4 give the spatial spectrums of the first W elements and the last W elements of $\tilde{\mathbf{s}}_r^{(l_2)}$, respectively, where the first W elements of $\tilde{\mathbf{s}}_r^{(l_2)}$ represent range estimates of the target with angle θ_1, and the last W elements of $\tilde{\mathbf{s}}_r^{(l_2)}$ represent range estimates of the target with angles θ_2 and θ_3. Figure 3 shows the range estimates of the target with angle θ_1, and Figure 4 shows the range estimates of the target with angles θ_2 and θ_3.

5.2. Detection and Estimation Performance. In this subsection, we carry out a series of simulations under different conditions to verify the superiority of the proposed algorithm. The ESPRIT method [19] and the U-ESPRIT method [20] are compared with the proposed algorithm. We assume that the frequency increment and the carrier frequency of the above algorithms are 1 kHz and 10 GHz, respectively. Suppose that two targets are located: $(-10.75°, 21565\,\mathrm{m})$ and $(25.68°, 44505\,\mathrm{m})$. The grids of the angle and range for the proposed algorithm are $[-90° : 1° : 90°]$ and $[0 : 1 : 150]$ km, respectively. The root mean square errors (RMSEs) for the range and angle are defined as

$$\mathrm{RMSE}_\theta = \sqrt{\frac{1}{\rho K}\sum_{i=1}^{\rho}\sum_{k=1}^{K}(\theta_{ik}-\theta_k)^2},$$
$$\mathrm{RMSE}_r = \sqrt{\frac{1}{\rho K}\sum_{i=1}^{p}\sum_{k=1}^{K}(r_{ik}-r_k)^2}, \quad (28)$$

where $\rho = 100$ is the number of Monte Carlo experiments. θ_{ik} and r_{ik} represent the estimates of θ_k and r_k in the i-th Monte Carlo experiment.

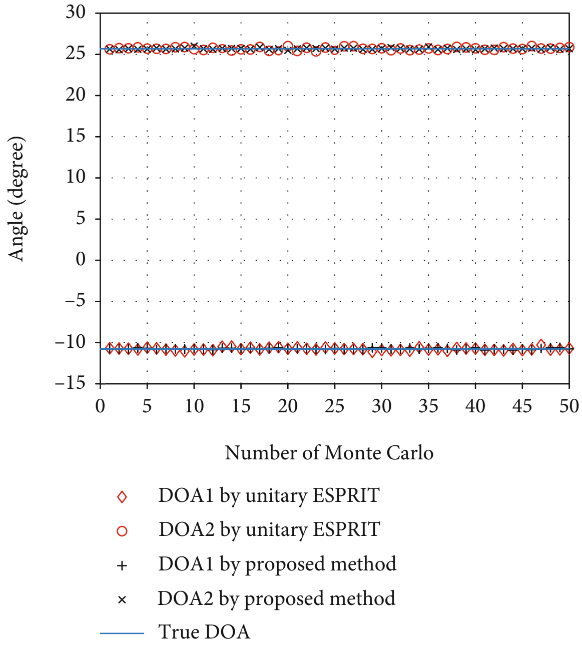

Figure 6: Comparison of DOA estimated results of 50 independent simulations.

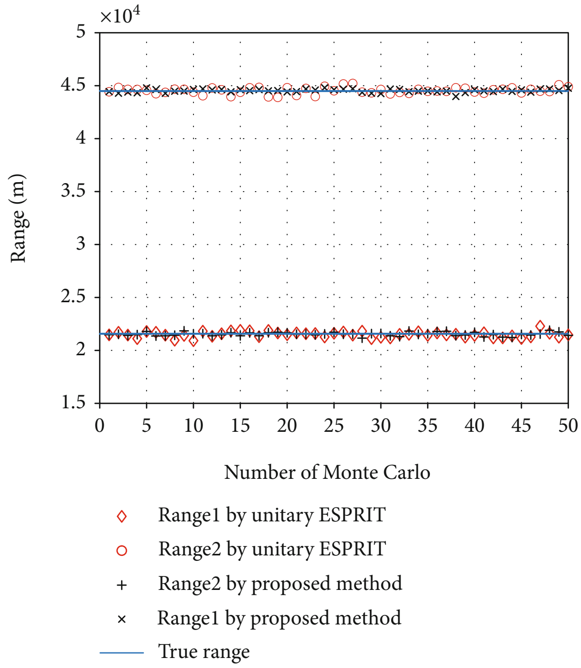

Figure 7: Comparison of range estimated results of 50 independent simulations.

The angle-range estimation results are exhibited in Figure 5 for FDA-MIMO radar using the proposed algorithm in which Monte Carlo trials are implemented 100 times. Moreover, the results of U-ESPRIT and the proposed algorithm are provided for comparison with the proposed

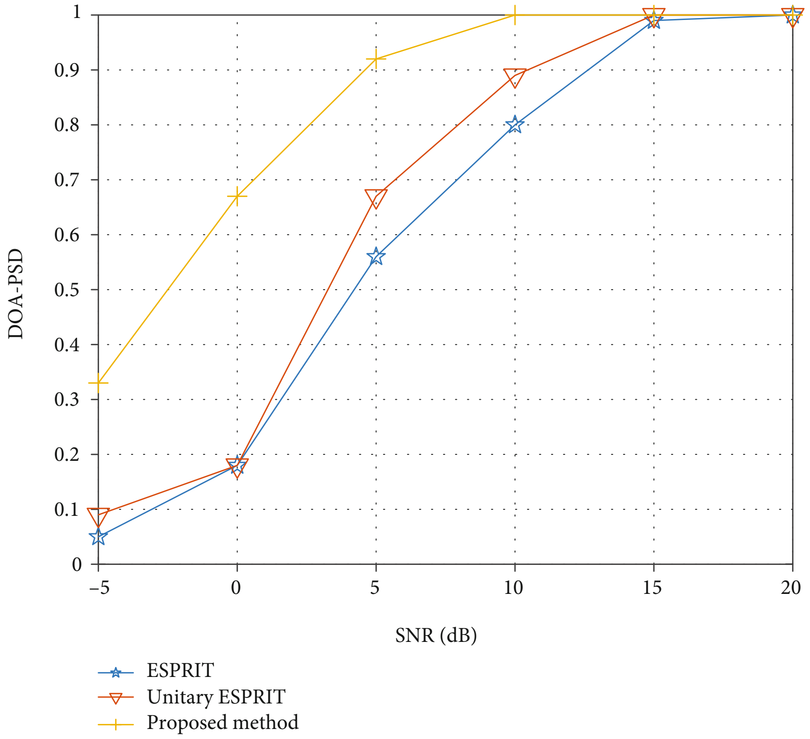

Figure 8: PSD versus SNR for angle estimation.

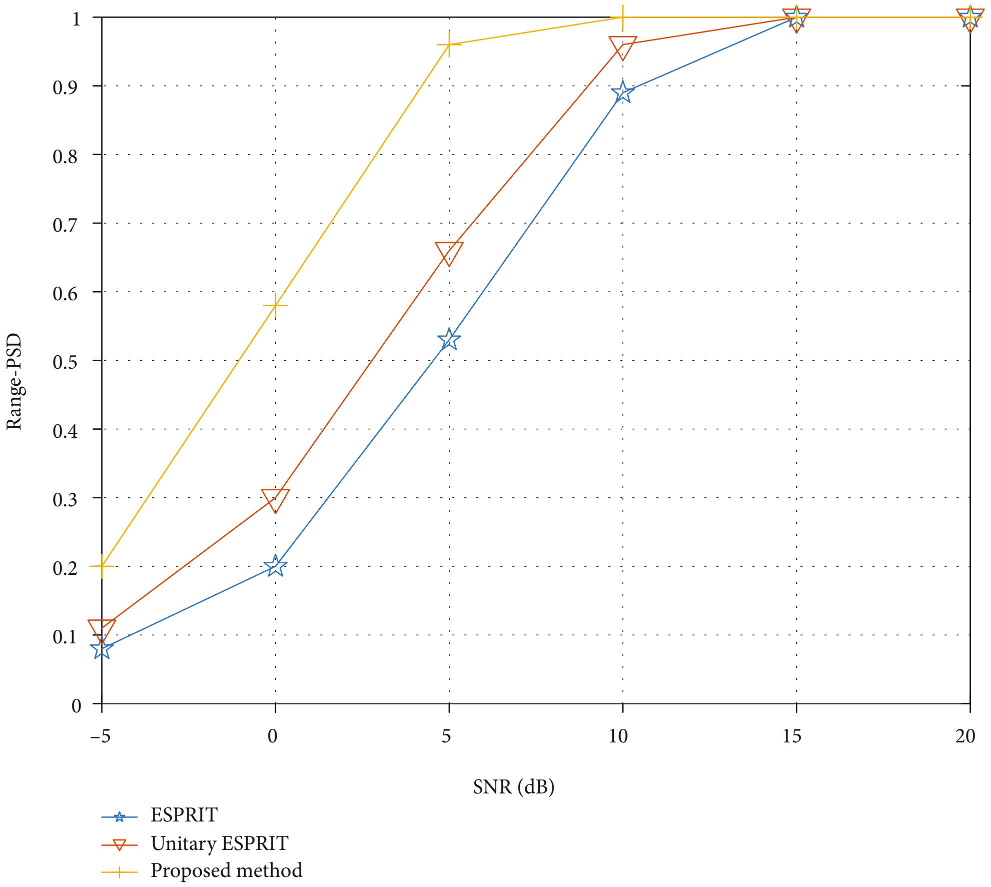

Figure 9: PSD versus SNR for range estimation.

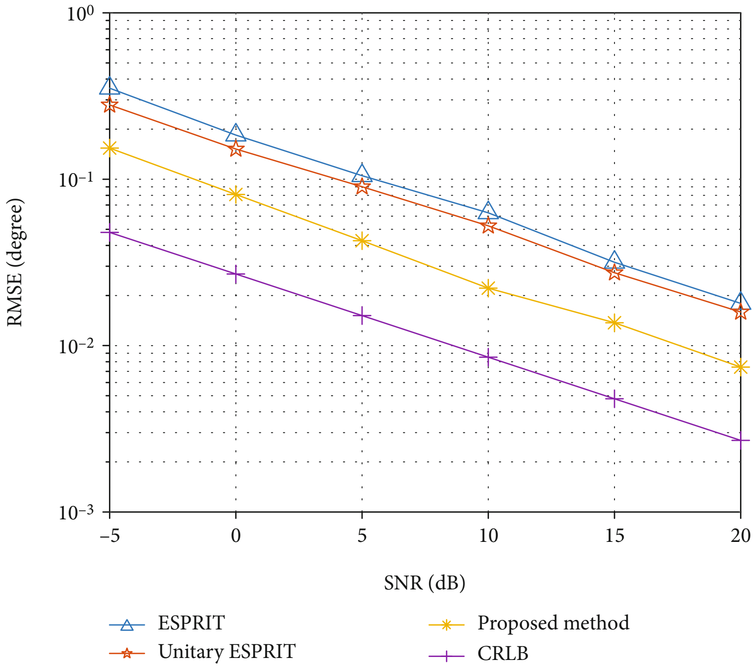

Figure 10: RMSE results of angle estimation versus SNR.

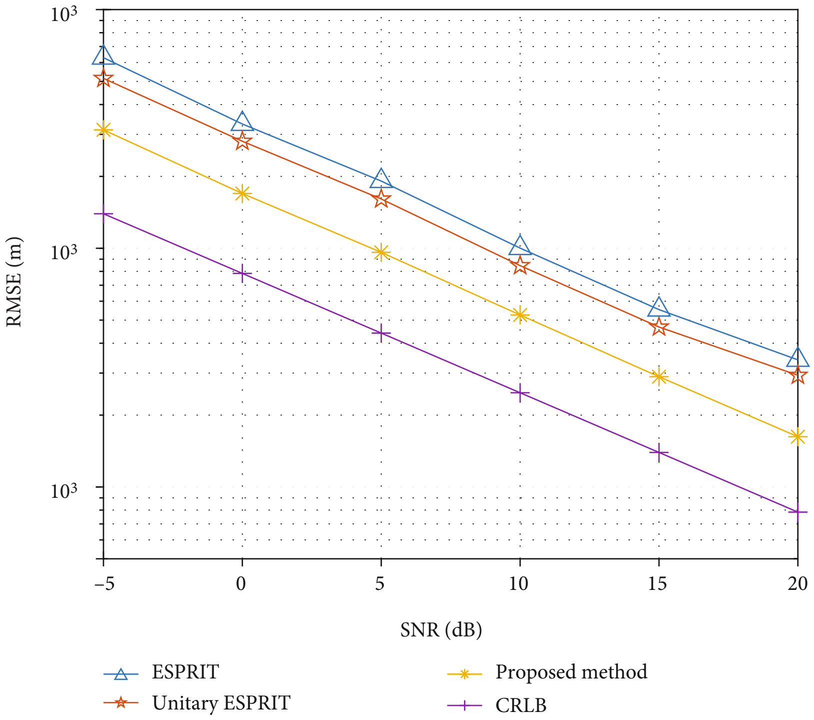

Figure 11: RMSE results of range estimation versus SNR.

algorithm in Figures 6 and 7, where SNR = 0 dB and $L = 50$. According to the results in Figure 5, we can conclude that the proposed algorithm can achieve precise matching of the range and angle. The angles and ranges of the two targets can be accurately estimated with a small number of snapshots. As can be seen from Figures 6 and 7, the parameter estimates of the proposed algorithm can considerably approach the real ones.

Figures 8 and 9 give the probability of successful detection (PSD) versus SNR for angle and range estimation in different algorithms, respectively, where $L = 50$. With regard to angle estimation, we define the successful detection if the

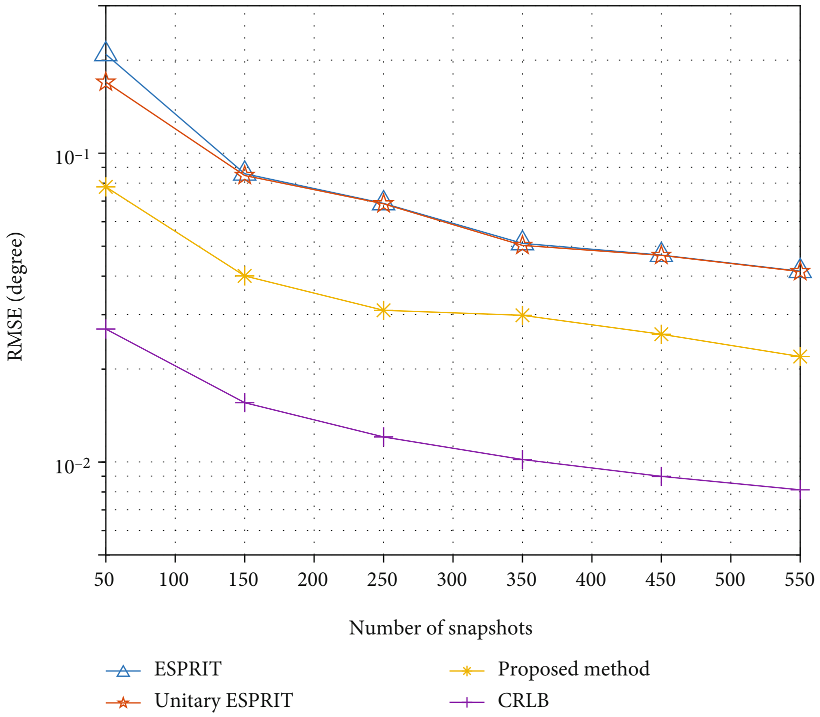

FIGURE 12: RMSE results of angle estimation versus snapshots.

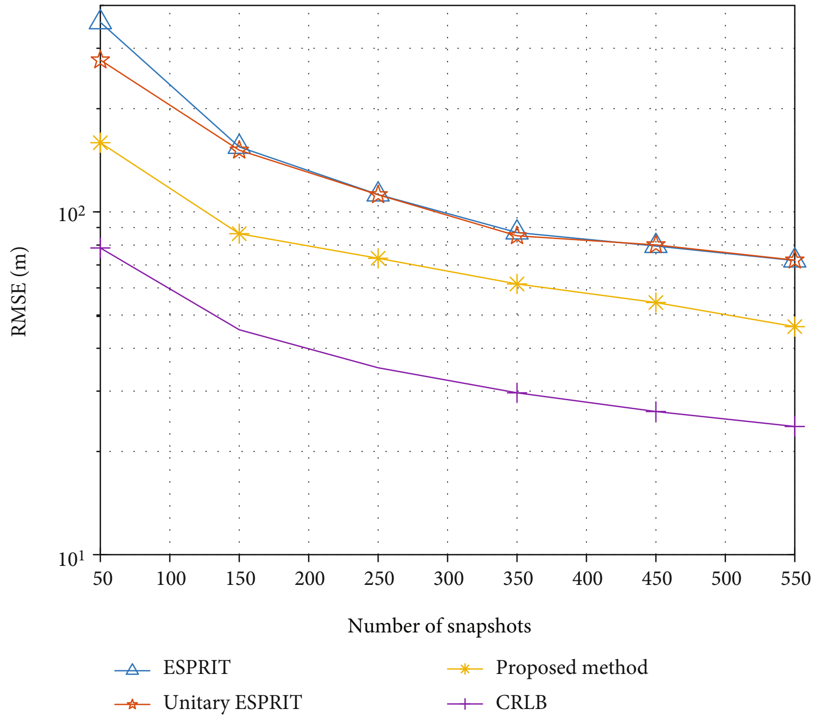

FIGURE 13: RMSE results of range estimation versus snapshots.

estimation result θ_{ik} satisfies $|\theta_{ik} - \theta_k| \leq 0.1^\circ$. The detection is successful if the estimation result r_{ik} satisfies $|r_{ik} - r_k| \leq 200$ m with regard to range estimation. As can be seen from Figures 8 and 9, the PSD of the proposed algorithm is superior to those of other algorithms under the same condition. Besides, with the increase of SNR, the proposed algorithm can achieve the PSD of 100% when SNR = 10 dB.

Figures 10 and 11 give the RMSE results of the angle and range estimates versus SNR with $L = 50$, respectively. It is obvious that the RMSE results gradually decrease with the increase of SNR, and in particular, the algorithm outperforms the other methods in range and angle estimation.

Figures 12 and 13 depict the RMSE results of the angle and range estimates versus snapshots, respectively, where SNR = 0 dB. As can be seen from Figures 12 and 13, the RMSE results of all algorithms improve with the increase of snapshots. Besides, the proposed method can obtain more accurate range and angle estimates than the other methods with the same number of snapshots.

6. Conclusion

In this paper, an accurate sparse recovery algorithm based on a double-pulse FDA-MIMO radar is proposed. In the proposed algorithm, we decouple the range and angle parameters of the FDA-MIMO radar with two pulses. Grid partition will bring high computational complexity. Therefore, we adopt an iterative grid refinement method to alleviate the above limitation on parameter estimation and propose a new iteration criterion to improve the error between real parameters and their estimates to get a trade-off between the high-precision grid and the atomic correlation. Compared with the subspace-based algorithms, the proposed algorithm performs better in simulation. Massive simulation results have certified that the proposed algorithm is prominent for parameter estimation of FDA-MIMO radar.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Acknowledgments

This work is supported by the Key Research and Development Project of Hainan Province (No. ZDYF2019011), the National Natural Science Foundation of China (Nos. 61701144, 61861015, and 61961013), the Program of Hainan Association for Science and Technology Plans to Youth R&D Innovation (No. QCXM201706), the scientific research projects of University in Hainan Province (No. Hnky2018ZD-4), the Young Elite Scientists Sponsorship Program by CAST (No. 2018QNRC001), and the Scientific Research Setup Fund of Hainan University (No. KYQD (ZR) 1731).

References

[1] C. Gentile, N. Alsindi, R. Raulefs, and C. Teolis, *Geoloc Techniques: Principles and Applications*, Springer, 2012.

[2] D. Ribas, P. Ridao, and J. Neira, *Underwater Slam for Structured Environments Using an Imaging Sonar*, Springer, 2010.

[3] M. Skolnik, *Radar Handbook Third Edition*, McGraw-Hill, 2008.

[4] H. Krim and M. Viberg, “Two decades of array signal processing research: the parametric approach,” *IEEE Signal Processing Magazine*, vol. 13, no. 4, pp. 67–94, 1996.

[5] E. Fishler, A. Haimovich, R. Blum, D. Chizhik, L. Cimini, and R. Valenzuela, “MIMO radar: an idea whose time has come,” in *Proceedings of the 2004 IEEE Radar Conference (IEEE Cat. No.04CH37509)*, pp. 71–78, Philadelphia, PA, USA, 2004.

[6] D. W. Bliss and K. W. Forsythe, “Multiple-input multiple-output (MIMO) radar and imaging: degrees of freedom and resolution,” in *The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003*, pp. 54–59, Pacific Grove, CA, USA, 2003.

[7] L. Wan, X. Kong, and F. Xia, “Joint range-Doppler-angle estimation for intelligent tracking of moving aerial targets,” *IEEE Internet of Things Journal*, vol. 5, no. 3, pp. 1625–1636, 2018.

[8] H. Huang, Y. Song, J. Yang, G. Gui, and F. Adachi, “Deep-learningbased millimeter-wave massive MIMO for hybrid precoding,” *IEEE Transactions on Vehicular Technology*, vol. 63, no. 3, pp. 3027–3032, 2019.

[9] H. Huang, J. Yang, H. Huang, Y. Song, and G. Gui, “Deep learning for super-resolution channel estimation and DOA estimation based massive MIMO system,” *IEEE Transactions on Vehicular Technology*, vol. 67, no. 9, pp. 8549–8560, 2018.

[10] X. Wang, L. Wan, M. Huang, C. Shen, and K. Zhang, “Polarization channel estimation for circular and non-circular signals in massive MIMO systems,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 13, no. 5, pp. 1001–1016, 2019.

[11] J. J. Zhang and A. Papandreou-Suppappola, “MIMO radar with frequency diversity,” in *2009 International Waveform Diversity and Design Conference*, pp. 208–212, Kissimmee, FL, USA, 2009.

[12] P. F. Sammartino, C. J. Baker, and H. D. Griffiths, “Frequency diverse MIMO techniques for radar,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 49, no. 1, pp. 201–222, 2013.

[13] J. Xu, G. Liao, S. Zhu, L. Huang, and H. C. So, “Joint range and angle estimation using MIMO radar with frequency diverse array,” *IEEE Transactions on Signal Processing*, vol. 63, no. 13, pp. 3396–3410, 2015.

[14] P. F. Sammartino, C. J. Baker, and H. D. Griffiths, “Range-angle dependent waveform,” in *2010 IEEE Radar Conference*, pp. 511–515, Washington, DC, USA, 2010.

[15] P. Antonik, M. C. Wicks, H. D. Griffiths, and C. J. Baker, “Frequency diverse array radars,” in *2006 IEEE Conference on Radar*, pp. 215–217, Verona, NY, USA, 2006.

[16] W. Wang, “Frequency diverse array antenna: new opportunities,” *IEEE Antennas and Propagation Magazine*, vol. 57, no. 2, pp. 145–152, 2015.

[17] W. Wang, “Overview of frequency diverse array in radar and navigation applications,” *IET Radar Sonar Navigation*, vol. 10, no. 6, pp. 1001–1012, 2016.

[18] W. Wang, “Range-angle dependent transmit beampattern synthesis for linear frequency diverse arrays,” *IEEE Transactions on Antennas and Propagation*, vol. 61, no. 8, pp. 4073–4081, 2013.

[19] B. Li, W. Bai, and G. Zheng, “Successive ESPRIT algorithm for joint DOA-range-polarization estimation with polarization sensitive FDA-MIMO radar,” *IEEE Access*, vol. 6, pp. 36376–36382, 2018.

[20] F. Liu, X. Wang, M. Huang, and L. Wan, “A novel unitary ESPRIT algorithm for monostatic FDA-MIMO radar,” *Sensors*, vol. 20, p. 827, 2020.

[21] J. Xiong, W. Wang, and K. Gao, “FDA-MIMO radar range–angle estimation: CRLB, MSE, and resolution analysis,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 54, no. 1, pp. 284–294, 2018.

[22] X. Wang, L. Wang, X. Li, and G. Bi, "Nuclear norm minimization framework for DOA estimation in MIMO radar," *Signal Processing*, vol. 135, pp. 147–152, 2017.

[23] C. Zhou, Y. Gu, X. Fan, Z. Shi, G. Mao, and Y. D. Zhang, "Direction-of-arrival estimation for coprime array via virtual array interpolation," *IEEE Transactions on Signal Processing*, vol. 66, no. 22, pp. 5956–5971, 2018.

[24] Z. Shi, C. Zhou, Y. Gu, N. A. Goodman, and F. Qu, "Source estimation using coprime array: a sparse reconstruction perspective," *IEEE Sensors Journal*, vol. 17, no. 3, pp. 755–765, 2017.

[25] X. Wang, W. Wang, J. Liu, and X. Li, "A sparse representation scheme for angle estimation in monostatic MIMO radar," *Signal Processing*, vol. 104, pp. 258–263, 2014.

[26] L. Wan, G. Han, L. Shu, and N. Feng, "The critical patients localization algorithm using sparse representation for mixed signals in emergency healthcare system," *IEEE Systems Journal*, vol. 12, no. 1, pp. 52–63, 2018.

[27] M. Tipping, "Sparse Bayesian learning and the relevance vector machine," *Mach Learn Res*, vol. 1, pp. 211–244, 2001.

[28] L. Zhao, G. Bi, L. Wang, and H. Zhang, "An improved autocalibration algorithm based on sparse Bayesian learning framework," *IEEE Signal Processing Letters*, vol. 20, no. 9, 2013.

[29] D. Malioutov, M. Cetin, and A. S. Willsky, "A sparse signal reconstruction perspective for source localization with sensor arrays," *IEEE Transactions on Signal Processing*, vol. 53, no. 8, pp. 3010–3022, 2005.

[30] J. Yin and T. Chen, "Direction-of-arrival estimation using a sparse representation of array covariance vectors," *IEEE Transactions on Signal Processing*, vol. 59, no. 9, pp. 4489–4493, 2011.

[31] W. Wang and H. Shao, "Range-angle localization of targets by a double-pulse frequency diverse array radar," *IEEE Journal of Selected Topics in Signal Processing*, vol. 8, no. 1, pp. 106–114, 2014.

[32] H. Wang, L. Wan, M. Dong, K. Ota, and X. Wang, "Assistant vehicle localization based on three collaborative base stations via SBL-based robust DOA estimation," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 5766–5777, 2019.

[33] D. Meng, X. Wang, M. Huang, L. Wan, and B. Zhang, "Robust weighted subspace fitting for DOA estimation via block sparse recovery," *IEEE Communications Letters*, vol. 24, no. 3, pp. 563–567, 2020.

[34] X. Wang, L. Wan, M. Huang, C. Shen, Z. Han, and T. Zhu, "Low-complexity channel estimation for circular and noncircular signals in virtual MIMO vehicle communication systems," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 4, pp. 3916–3928, 2020.

[35] P. Stoica and R. Moses, *Spectral Analysis of Signals*, Prentice Hall, 2005.

Hindawi
Wireless Communications and Mobile Computing
Volume 2020, Article ID 6684293, 16 pages
https://doi.org/10.1155/2020/6684293

Research Article

Deep Reinforcement Learning-Based Collaborative Video Caching and Transcoding in Clustered and Intelligent Edge B5G Networks

Zheng Wan[1] **and Yan Li**[1,2]

[1] *School of Information Management, Jiangxi University of Finance and Economics, No. 665, West Yuping Road, Nanchang, Jiangxi 330032, China*
[2] *School of Information Engineering, Nanchang Institute of Technology, No. 289, Tianxiang Road, Nanchang, Jiangxi 330099, China*

Correspondence should be addressed to Yan Li; yli@nit.edu.cn

Received 14 October 2020; Revised 7 November 2020; Accepted 25 November 2020; Published 12 December 2020

Academic Editor: Lisheng Fan

In the next-generation wireless communications system of Beyond 5G networks, video streaming services have held a surprising proportion of the whole network traffic. Furthermore, the user preference and demand towards a specific video might be different because of the heterogeneity of users' processing capabilities and the variation of network condition. Thus, it is a complicated decision problem with high-dimensional state spaces to choose appropriate quality videos according to users' actual network condition. To address this issue, in this paper, a Content Distribution Network and Cluster-based Mobile Edge Computing framework has been proposed to enhance the ability of caching and computing and promote the collaboration among edge severs. Then, we develop a novel deep reinforcement learning-based framework to automatically obtain the intracluster collaborative caching and transcoding decisions, which are executed based on video popularity, user requirement prediction, and abilities of edge servers. Simulation results demonstrate that the quality of video streaming service can be significantly improved by using the designed deep reinforcement learning-based algorithm with less backhaul consumption and processing costs.

1. Introduction

Beyond fifth-generation (B5G) networks is the next-generation wireless communications systems. They are desired to provide rather reliable services with super high transmission rate, ultralow latency, very little energy loss, excellent quality of experience (QoE), and much enhanced security [1]. Due to providing mobile edge computing and edge caching capabilities together with machine learning, edge intelligence is emerging as a new concept and has extremely high potential in addressing the new challenges in B5G networks [2, 3]. In wireless communication networks, video streaming services have hold a surprising proportion of the whole network traffic. In particular, because of the impact of the epidemic at 2019-nCoV in this year, it has greater dependence and demand on online video streaming services, such as online meeting, online teaching, and online shopping.

In recent years, the number of smart devices has been explosively grown, which led to unprecedented increase in the demand on video streaming service. In video streaming service, it generally requires higher data rates and bigger system capacity. The overall mobile data traffic has experienced 17-fold growth from 2012-2017 as summarized in Cisco Visual Networking Index [4]. Mobile videos account for more than half of this data traffic and are predicted to further grow by 2022, accounting for 79% of the total data traffic. Due to the immense demands of mobile videos, mobile network operators can not be enough to satisfy the users' demands on high-quality video streaming services.

To address this issue, firstly, edge video caching has been recognized as a promising solution to reduce the data traffic, because edge video caching can bring videos closer to the users, which will reduce data traffic going through the backhaul links and the time required for video delivery [5]. Motivated by serving the users better, different edge caching strategies have been studied recently. Secondly, a good video QoE is very important to users. In a full range of user mobile devices, the source video streams are needed to be transcoded into multiple representations. But the video transcoding is also an extremely computation intensive and time-consuming work [6].

Recently, mobile edge computing (MEC) has been introduced as an emerging paradigm in the edge of the cellular Radio Access Network (C-RAN) [7–12]. The MEC servers are implemented particularly at the BSs in the mobile edge computing platforms, enabling video streaming services in close-proximity to the mobile users. Due to this position, MEC presents a unique opportunity to not only perform edge caching but also implement edge processing.

Due to the heterogeneity of users' processing capabilities and the variation of network condition, the user preference and demand towards a specific video might be different. For example, users with better network condition usually prefer high-resolution videos while users with poor network condition may desire for appropriate quality videos according to their actual network condition. Based on this phenomenon, adaptive bitrate (ABR) streaming [13, 16] has been widely used to improve the quality of delivered video. In ABR streaming, the bitrate of the streaming video will be chosen according to the users' specific request and actual network condition. A video content is encoded into multiple layers with different bitrates, satisfying different users' requirement. Then, each video layer will be further segmented into many small video chunks, which contains several seconds of the video content. Thus, users can dynamically adjust video layer for different video chunks, depending on their actual network conditions. So, it is a complicated decision problem with high-dimensional state spaces to choose appropriate quality videos according to users' actual network condition. There are obvious advantages in deploying ABR streaming locally at multi-MEC servers in RAN, such as avoiding the long latency and reducing the prestorage pressure at RAN [14–18]. Then, the required video layer of mobile users can be transcoded in an on-demand fashion, which can improve ABR streaming performance over mobile edge computing networks when it is directly served from a local MEC server.

Deep learning has a strong perception ability. It is mainly used to solve classification and regression problems by capturing and analyzing data features [19–22], but it does not have the ability to make decisions. Reinforcement learning [23] has the ability to make decisions, but it is helpless to perceive problems and cannot handle high-dimensional data. Reinforcement learning is actually an agent that learns the best decision sequence during the interaction with the environment. In order to deal with the complicated control and decision problems with high-dimensional state spaces, a promising solution has been given in recent development of deep reinforcement learning (DRL) [24]. DRL consists of two modules: deep learning and reinforcement learning. It uses deep learning to extract features from complex high-dimensional data and transform it into a low-dimensional feature space. Then the low-dimensional feature state space inputs into reinforcement learning to make decisions for seeking more rewards. The goal of DRL is to enable an agent to take the best action in the current state to maximize long-term gains in the environment [25, 26]. And the interaction between the agent's action and state is learned by leveraging the deep neural network (DNN). Due to these characteristics, DRL becomes a powerful tool in robotics, wireless communication, etc. [27–29]. Since the advent of deep Q network (DQN) [30–32] in 2013, a large number of algorithms and papers to solve practical application problems have appeared in the field of deep reinforcement learning. The basic idea behind many reinforcement learning algorithms is to estimate the Q value function by using the Bellman equation as an iterative update. Such value iteration algorithms converge to the optimal Q value function.

This paper intends to propose a video transmission model combining MEC and Content Distribution Network (CDN) technology, which interconnects the CDN network with the MEC network through the CDN tips. Also, we focus on exploiting MEC storage and processing capabilities to improve the performance of high-quality streaming services. We aim to solve the collaborative caching and transcoding for multi-MEC servers by using the DRL algorithm in mobile edge computing system. Specifically, the main contributions of this paper are as follows:

(i) A CDN and Cluster-based Mobile Edge Computing (2C-MEC) system model has been proposed, which promotes cooperation among MEC servers and reduces unnecessary backhaul consumption and processing costs. We design a MEC-enabled collaborative caching and transcoding for multi-MEC servers in the 2C-MEC system by leveraging video caching and transcoding in the vicinity of RAN at multi-MEC servers

(ii) The optimization problem of collaborative caching and transcoding for multi-MEC servers can be formulated as a stochastic Markov decision process to maximize the time-averaged Deep Q-Network (DQN) reward. The reward is defined as the weighted sum of the cache hit rate, user perceived QoE, the cost of performing transcoding, and transmission at multi-MEC servers. Then, we develop a DRL-based algorithm to automatically obtain the intracluster collaborative caching and transcoding decisions, which are executed based on video popularity, user requirement prediction, and abilities of MEC servers

(iii) Simulation results demonstrate that video streaming service can be significantly improved by using the proposed DRL-based algorithm compared with the scheme that video transcoding is not implemented at the MEC servers, with less backhaul consumption and processing costs

The remainder of this paper is organized as follows. Section 2 presents a related work. Section 3 describes the framework design of system and Section 4 formulates the optimization problem. The DRL-based algorithm is presented in

Section 5. Section 6 presents the simulation results and analysis, followed by conclusions in Section 7.

2. Related Work

The research on the application of DRL to wireless network transmission optimization in the MEC environment is extensive studied recently. It can be seen that the research in this area mainly began in 2018, increasing quickly year by year after 2018. Furthermore, the application of DRL in video transmission optimization under MEC environment is less. The current research in this area includes the following categories: DRL-based caching strategy, DRL-based real-time transcoding scheduling decision, DRL-based wireless network communication resource allocation [33–37], and DRL-based offloading and service migration of computing tasks [38–43]. In this paper, we mainly focus on the first two topics, trying to satisfy the requests of quality for user's streaming service.

2.1. DRL-Based Caching Strategy. For edge video caching at MEC servers, video caching policy is driven by video popularity. Therefore, knowing the video popularity is key to solve the video caching problem. To avoid such drawbacks, combining DRL methods are introduced to implement video cache strategies, which is an important research direction [44–47]. In order to reduce the traffic load of backhaul and transmission latency, Wei et al. [48] proposed the Q-Learning-based collaborative cache algorithm to solve the intelligent baseband unit pool cache problem. Yang et al. [49] considered the task offloading decision, cache allocation, and computation allocation problems in single MEC sever; a DRL algorithm was proposed to solve this optimization problem with low complexity. Zhong et al. [50, 51] presented a DRL-based framework with Wolpertinger architecture for content caching at the single MEC. They proposed deep actor-critic reinforcement learning-based policies for both centralized and decentralized content caching, aiming at maximizing the cache hit rate in centralized edge caching and the cache hit rate and transmission delay as performance metrics in decentralized edge caching. Gursoy et al. [52] designed a deep actor-critic RL-based multiagent framework for the edge caching problem in both a multicell network and a single-cell network with D2D communication.

Applying DRL to cache technology mainly solves the problem of cache content location decision, cache update strategy, and cache content delivery. It implements resource allocation and cache scheduling by using deep learning to analyze and learn network information. Then corresponding video content and bitrate versions are cached to improve cache hit radio and utilization of cache resources. However, the lack of transcoding on the network edge will reduce the video cache hit rate.

2.2. DRL-Based Transcoding Scheduling Strategy. The user's demand towards a specific video might be different because of the heterogeneity of their actual network condition. To address this issue, transcoding in network edge has been widely used to improve the quality of delivered video on the wireless networks. To achieve accurate QoE, Liu et al. [53] and Zhang et al. [54] presented deep learning-based QoE prediction called DeepQoE. Then in [53], the authors designed a content-aware bitrate adaptation policy with the objective to prefetch a higher resolution version for video clips that is in line with viewers' interests. Zhang et al. [54] also developed a DeepQoE-based ABR system to verify that their framework can be easily applied to multimedia communication service. To address the challenge of how to allocate bitrate budgets for different parts of the video with different users' interest, Gao et al. [55] proposed a content-of-interest-based rate adaptation scheme for ABR. They designed a deep learning approach for recognizing the interestingness of the video content and a DQN approach for rate adaptation according to incorporating video interestingness information. Considering joint computation and communication for ABR streaming, Guo et al. [56] presented a joint video transcoding and quality adaptation framework for ABR streaming. Inspired by recent advances of blockchain technology, Liu et al. [57] proposed a novel DRL-based transcoder selection framework for blockchain-enabled D2D transcoding systems where video transcoding has been widely adopted in live streaming services, to bridge the resolution and format gap between content producers and consumers. To accommodate personalized QoE with minimized system cost, Wang et al. [58] proposed DeepCast, which is an edge-assisted crowdcast framework. It makes intelligent decisions at edges based on the massive amount of real-time information from the network and viewers. In [59], using DRL to train a neural network model for resource provisioning, Pang et al. designed a joint resource provisioning and task scheduling approach for transcoding live streams in the cloud.

The application of DRL in transcoding scheduling decisions mainly focuses on making intelligent real-time transcoding decisions at the network edge based on a large amount of real-time information from the network and customers. In order to meet the high-quality video service experience of requirements of different users, DRL-based transcoding scheduling strategy will aim at achieving personalized QoE with minimized system cost.

2.3. Our Vision and Motivation. Inspired by the success of DRL in solving complicated control problems, DRL-based methods are commonly used in caching and transcoding strategy for MEC system. But there are still some issues which are needed to be resolved. (i) At present, there are many systems mainly studying single-MEC server. However, single-MEC server does not have enough storage and computing ability to satisfy the needs of different users. (ii) There are few researches on the cooperation mode and efficiency of multi-MEC servers. The completion of intensive tasks requires efficient collaboration among multi-MEC servers. (iii) In multi-MEC servers' system, the load balance among MEC servers and the resource utilization of the MEC server are basically not considered. (iv) According to users' network conditions, adaptively collaborative caching and transcoding methods in ABR streaming are needed further explored.

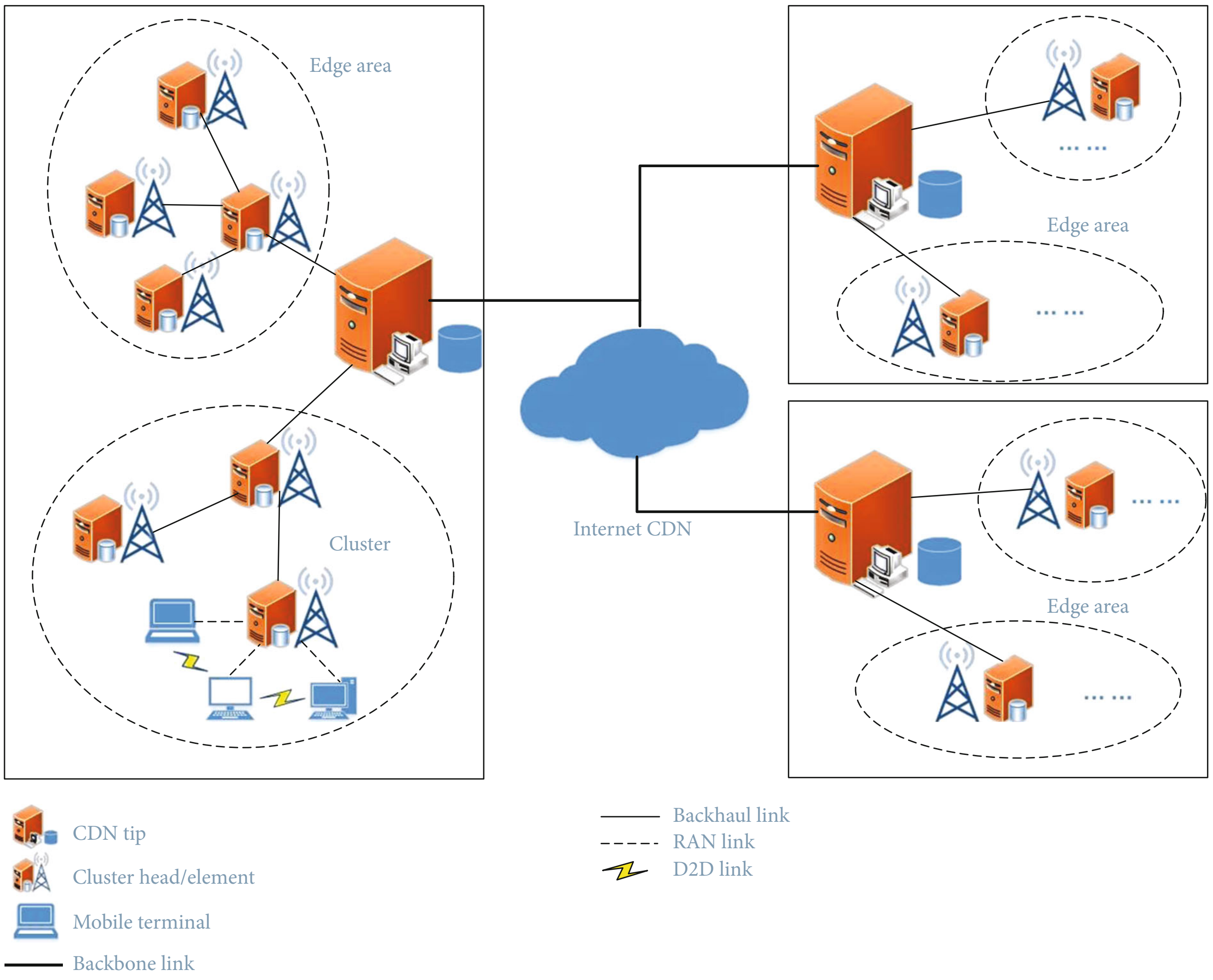

Figure 1: A CDN and Cluster-based Mobile Edge Computing (2C-MEC) system.

To address these issues, in this paper, a CDN and Cluster-based Mobile Edge Computing (2C-MEC) system model has been presented, which promotes cooperation among MEC servers and reduces unnecessary backhaul consumption and processing costs. Then, aiming to exploit MEC storage and processing capabilities to improve performance of high-quality streaming services, we focus on solving the collaborative caching and transcoding for multi-MEC servers by using the DRL algorithm in the 2C-MEC system model.

3. Framework Design of System

3.1. 2C-MEC System Model. In order to meet the transmission requirements of real video services in the internet, the video transmission strategy based on mobile edge computing must consider a heterogeneous wireless access network environment and popular video transmission technology. As shown in Figure 1, this paper intends to propose a video transmission model combining Cluster-based MEC and CDN technology, which is called as a CDN and Cluster-based Mobile Edge Computing system.

The video transmission model-based mobile edge computing is seamlessly connected with the current popular video transmission CDN technology. In this model, the edge area consists of the CDN tips (that is, the "edge node" in the CDN, in order to distinguish it from the edge computing node, called "CDN tip" in this paper) and many edge computing nodes in the local area (may be deployed at small base stations, macro base stations, and locations higher than the macro base stations). Thereby, the computing, storage, and communication capabilities of edge computing nodes are used to assist in the deployment of sparse CDN tips to optimize wireless video transmission across the entire network.

Due to the large number of edge nodes and the large difference in capabilities among them, a hierarchical management model is proposed to cluster edge nodes. The communication protocols within and among clusters can draw on the related research of sensor networks and P2P networks. The influencing factors of edge node clustering strategy include edge node capabilities, geographic location distribution, number and activity of users. The 2C-MEC system can promote mutual cooperation among MEC servers and reduces unnecessary backhaul consumption and processing costs.

Based on the proposed Cluster-based Mobile Edge Computing framework, on the one hand, the storage and computing capabilities of the MEC servers have been improved. The 2C-MEC system enables the MEC servers' collaboration within the cluster to have sufficient storage and computing power to meet users' needs. On the other hand, the collaboration among MEC servers is promoted. Under this framework, it is possible to pursue the multi-MEC collaboration method within the cluster, which focuses on exploring the effective ways of multi-MEC servers' collaboration of caching and transcoding. On the contrary, existed studies focused on "cloud-edge" collaboration or "edge-edge" collaboration.

In this paper, we plan to design the edge node clustering algorithm based on the following ideas: (i) firstly, the cluster division is based on the principle of proximity to geographic location. (ii) Secondly, the overall service capabilities of the nodes in the cluster should match their users' needs, and the edge service capabilities among different clusters should be balanced to a certain extent. (iii) Thirdly, if the edge node is located in the intersection area of two clusters, the appropriate cluster is selected based on the similarity of the video access preferences of the users managed by this node and the video access preferences of the users managed by other nodes in one cluster. (iv) Finally, after the clustering is completed, we can comprehensively consider the computing, storage, communication capabilities of the edge node, and its communication delay with other nodes in the cluster to elect this cluster's head.

In a word, the 2C-MEC system model proposed in this paper is compatible with popular CDN technology, resulting in conveniently utilization of its research results in cache replacement, content prefetching and load balancing. Furthermore, the ability of MEC to utilize heterogeneous edge nodes with different capabilities and deployments further improves the quality of video transmission.

3.2. Rebuffer Model. In order to keep continuous playback in video streaming service, a playback buffer is usually deployed at the user device, in which the video chunks are downloaded into. The rebuffer model used in this paper comes from the reference [60]. Let $B(t)$ denote the bitrate of the chunk at time stage t for the user. And $W(t)$ denotes the wireless transmission rate (bits/second) of user experienced during time stage t. Then, the buffer occupancy rate $L(t)$ is defined as follows:

$$L(t) = \frac{\text{Buffer occupancy}}{\text{Buffer size}}. \tag{1}$$

When $B(t)/W(t) < 1$, the new video chunk is put into the buffer at rate of less than 1; then, the buffer decreases. In another way, if more than one chunk is played before the next chunk arrives, then, the buffer is depleted and the rebuffering is happened. So, in the rebuffer model, the term of rebuffering time and buffered video time are usually introduced, which are used in Reference [56]. A video has some chunks; each chunk also contains a fixed duration of video, such as D seconds of video. Let $T(t)$ denote the buffered video time at playback buffer at the beginning of time stage t. In the rebuffer model, we assume that one chunk will be downloaded into the buffer at one time. The total downloading time of one chunk during time stage t, denoted by $d(t)$, can be expressed as

$$d(t) = \frac{B(t) * D}{W(t)}. \tag{2}$$

Furthermore, the video rebuffering time of playback buffer during time stage t is denoted as $R(t)$. Then, we can get

$$\begin{aligned} R(t) &= \max\left(d(t) - T(t), 0\right), \\ T(t+1) &= D + \max\left(T(t) - d(t), 0\right). \end{aligned} \tag{3}$$

3.3. Video Quality Rate Model. In video processing, Peak Signal to Noise Ratio metric (PSNR) is the de facto standard criterion to provide objective quality evaluation between the original frame and the compressed one. In the video quality evaluation, the video quality rate $q(t)$ of a video coded at rate $B(t)$ can be approximated by a logarithmic function [61] as follows:

$$q(t) = \beta \log\left(B(\mathrm{t})\right), \tag{4}$$

where the β value can be obtained from the video encoder during the encoding in video source. Generally, the mentioned quality rate $q(t)$ is a nondecreasing function, which means a higher bitrate may be a high definition video while a lower bitrate may be a standard definition video.

Then, let $B_u(i, t) \in \{B_1, B_2, \cdots, B_{\max}\}$ and $B_{\max}$ be the set of all video layers after video transcoding and the highest video level at the MEC servers, respectively. And $B_u(i, t)$ denotes the bitrate assigned to user i at timeslot t.

3.4. Cache Hit Rate Model. In our setting, requests by all users are served by the MEC severs; all video have the same size, and there are no priorities for different users, while there are popularities for different videos. Videos popularity distribution is always the key to solve the video caching problem. Considering the changing popularities, the probability that the requests of video v is defined as Z_v, which follows the Zipf distribution [16] as follows:

$$Z_v = \frac{v^{-\alpha}}{\sum_{v=1}^{V} v^{-\alpha}}, \tag{5}$$

where $\alpha > 0$ is the parameter of Zipf distribution which indicates the skewness degree. According to our setting, the video streaming service quality of content caching can be evaluated in terms of the cache hit rate. The cache hit rate $\mathrm{CRH}(t)$ in T requests during time stage t is defined [40] as

$$\mathrm{CRH}(t) = \frac{\sum_{i=1}^{T} 1(H_i)}{T}, \tag{6}$$

where indicator function $1(H_i)$ is defined as

$$1(H_i) = \begin{cases} 1, & H_i \in C_T, \\ 0, & H_i \notin C_T, \end{cases} \tag{7}$$

where C_T represents the cache state during this period; if there is the cache of video, the request H_i can hit in the cache.

3.5. System Cost Model. In the system cost model, most of the operational cost consists of bandwidth cost and transcoding cost in video streaming service. The fraction of other service cost is negligible comparing with the above two kinds of cost.

Then, the bandwidth cost $C_b(t)$ [62] of all MEC servers in the cluster can be obtained by the following formula:

$$\begin{aligned} C_b(t) &= \sum_{n=1}^{M} P(n,t)\cdot W(n,t), \\ W(n,t) &= \sum_{i\in U^t} B_u(i,t)\cdot I^t(i,n), \quad n\in\{0,\cdots,M\text{-}1\}, \end{aligned} \tag{8}$$

where U^t and M are the user group and the numbers of severs in the cluster at time stage t. And $I^t(i,n)$ is an indicator that represents whether user i is connected to MEC server n at the time stage t. Respectively, $P(n,t)$ and $W(n,t)$ be the unit bandwidth price and the amount of bandwidth usage in the MEC server n.

Beside the bandwidth cost, the video streaming service also needs to consider the transcoding cost. Based on the definition and description of video transcoding in [56, 62], the transcoding cost is closely related to the input bit-rate, target bit-rate, the video length, and the number of CPU cores needed for transcoding according to the video pricing model. Then, we define the transcoding cost incurred at time stage t as

$$O(t)=\sigma*(L_{\max}-l)*T_v*N_{\text{cpu}}, l\in\{L_1,L_2,\cdots,L_{\max}\}, \tag{9}$$

where σ is an adjustable parameter and symbols of l, T_v, and N_{cpu} represent the level of input video, the video length, and the number of CPU cores required for transcoding, respectively.

In order to simplify the problem formulation, in our system cost model, the operational cost mainly consisted of bandwidth cost and transcoding cost. Since bandwidth cost and transcoding cost have different measurement units, bandwidth cost reflects the network transmission capacity, while transcoding cost reflects the computing power of the MEC node; it is not easy to unify the corresponding dimensional units. However, in the comparison of simulation experiments, only the cost of comparing different environments is required. Therefore, like the design in Reference [62], the bandwidth cost and transcoding cost can be regarded as values without a unit of measurement, and there is no need to consider the details of the unit of measurement. The operational cost can be expressed as

$$C(t)=C_b(t)+O(t). \tag{10}$$

4. Optimization Problem Formulation

Based on using the DRL algorithm for resource optimization in the 2C-MEC system, we describe the three basic elements of reinforcement learning. They are the state, action, and reward of the collaborative video caching and transcoding optimization problem.

4.1. State Space. The state at time stage t is jointly determined by the four tuples, the current bandwidth cost $C_b(t)$, the current buffer occupancy rate $L(t)$, the current rebuffer time $R(t)$, and the current video quality $q(t)$. Then, the state space $S(t)$ at time stage t can be defined as follows:

$$S(t)=\{C_b(t),L(t),R(t),q(t)\}, \tag{11}$$

where the state space is denoted as S.

4.2. Action Space. The control action for the agent is to select the video caching strategy and video transcoding strategy for the next requested video chunk according to the current system state. In this network, the action at each time stage t is the joint video cache updating, cache$(M(t),U(t))$, and video transcoding layer adaption decision, $B_u(i,t)$.

So, the action is selected from the action set $A(t)$, in which $M(t)$, $U(t)$, and $B_u(i,t)$ represent the number of MEC severs selected in the cluster, the decision of video cache updating, and the target video layer, respectively. Then, the action space can be described as

$$A(t)=\{M(t),U(t),B_u(i,t)\}, \tag{12}$$

where the action space is denoted as A.

In practice, since the numbers of MEC severs in a cluster and the set of all video layers are not large; also, the decision of video cache updating is only yes or no; the number of possible actions in the state space set for the collaborative video caching and transcoding problem can be not very large.

4.3. Reward. The reward should reflect the objective of the framework, which, in our case, is to reduce the operational cost and desire best QoE for users by solving the collaborative caching and transcoding for multi-MEC servers. In our paper, we define the reward function during time stage t, denoted by $r(t)$, as follows:

$$\begin{aligned} r(t)=\omega_1\text{CRH}_{sl}(t)+\lambda q(t)-\omega_2\|q(t)-q(t-1)\|-\omega_3R(t) \\ -\omega_4C_b(t)-\omega_5O(t). \end{aligned} \tag{13}$$

The first term on the right-hand side of (13) is the weighted sum of the short and long-term cache hit rate. Considering the number of requests for local video in the next epoch, the short-term cache hit rate $\text{CRH}_s(t)$ can be either 0 or 1. Thus, let the total normalized number of requests for local video within the last 20 requests as the long-term cache hit rate $\text{CRH}_l(t)\in[0,1]$. The total cache hit rate $\text{CRH}_{sl}(t)$ for each step is defined as the weighted sum of the short and long-term cache hit rate, which is defined as

$$\text{CRH}_{sl}(t)=\text{CRH}_s(t)+\mu*\text{CRH}_l(t), \tag{14}$$

where μ is the weight to balance the short and long-term cache hit rate.

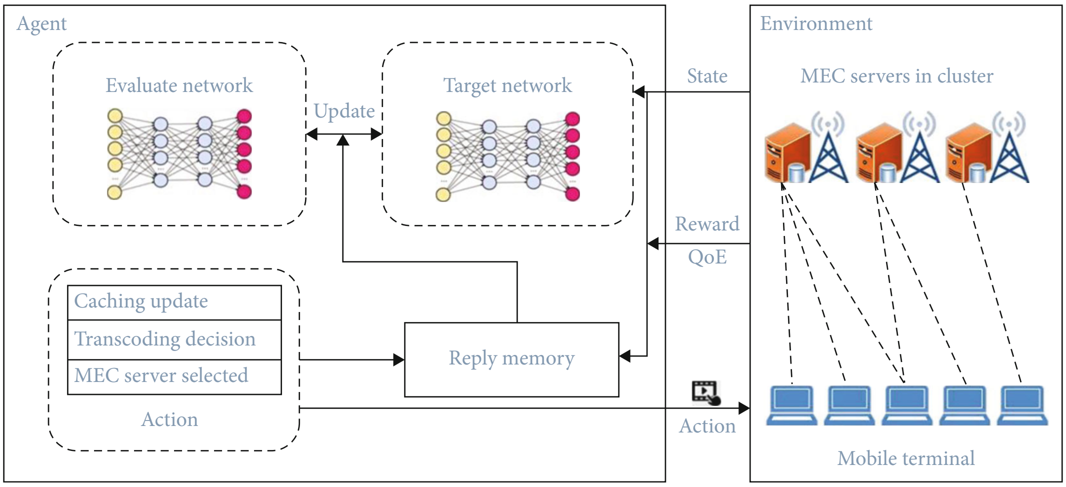

Figure 2: The design of DRL-based intracluster collaborative caching and transcoding framework.

The second, third, and fourth terms on the right-hand side of (13) are video quality, video quality variation, and video rebuffering time, respectively. The fifth and sixth terms on the right-hand side of (13) are two penalty terms for the bandwidth cost and transcoding cost in each step. The total cache hit rate, video quality, video quality variation, and video playback rebuffering time are directly associated with user perceived QoE in the video streaming service. And the weights $\omega_1, \lambda, \omega_2, \omega_3, \omega_4, \omega_5$ are the weighting parameters.

4.4. Problem Formulation. In this paper, our objective is to derive the jointly optimal video caching policy and video transcoding policy for maximizing the rewards in video streaming service. Future rewards and present rewards have different importance and weights because of the uncertainly of system dynamics. The objective of the joint video caching policy and video transcoding policy is to maximize the expected average reward. Then, we can formulate the dynamic optimization problem as a Markov decision process (MDP) as follows:

$$\begin{aligned} \max_{M(t),U(t),B_u(i,t)} \quad & J(t) = E\left[\sum_{t=0}^{T-1} \gamma^t r(t)\right] \\ \text{s.t.} \quad & C1: M(t) \in \{0, 1, \cdots, M\}, \quad \forall t \\ & C2: U(t) \in \{0, 1\}, \quad \forall t \\ & C3: B_u(i, t) \in \{B_1, B_2, \cdots, B_{\max}\}, \quad \forall t, \end{aligned} \tag{15}$$

where $\gamma \in (0, 1]$ is the discount factor.

It is impractical for the optimization problem with a large number of states in state space. But the DRL algorithm has been proved a useful mathematical tool for large-scale optimization problem which does not need any prior knowledge of state transition probability. Based on this, we propose a DRL-based algorithm to solve the optimization problem in formulation (15). Thus, the design of DRL-based intracluster collaborative caching and transcoding framework is shown in Figure 2.

5. DRL-Based Intracluster Collaborative Caching and Transcoding Algorithm

5.1. Deep Reinforcement Learning-Based Collaborative Video Caching and Transcoding for Multi-MEC Servers. Based on DQN's excellent performance when dealing with discrete state space and action space, we adopt DQN for learning the intracluster collaborative caching and transcoding policy. Specifically, as illustrated in Figure 2, the inputs of the deep neural network are the video service system states listed in Equation (11), and the outputs of the network are the Q value function, $Q(s, a; \theta)$, for each action listed in Equation (12).

We illustrate the details of the DRL-based learning algorithm for collaborative caching and transcoding for multi-MEC servers in Algorithm 1.

6. Simulation Results and Analysis

In this section, firstly, we illustrate the experiment settings. Then, the computer simulations are carried out to demonstrate the performance of the proposed DRL algorithm of collaborative caching and transcoding for multi-MEC servers in mobile edge computing wireless networks.

6.1. Experimental Settings

6.1.1. Data Generation. In our experiments, the user data of requests is generated randomly, while the video data of users' requests is generated according to the Zipf distribution. We have collected different numbers of requests in one episode as the testing data, such as 30, 40, and 50. In order to make the experiment more comprehensive, we generate two types of data sets. Firstly, the video data set in users' different-number requests was generated with unchanged popularity distribution with Zipf parameter set as 1.3. Then, the video data set in users' same-number requests was generated with a varying Zipf parameter.

```
Initialization:
  Initialize replay memory D to capacity $N$
  Initialize Q network and target Q network with random weights
  Initialize MEC service matrix $V$ of requests
for episode =1, $M$ do
  Generate the user requests data
  Observe the initial state $s_1$ as illustrated in Eq. (11)
  for t =1, $T$ do
    Give a random probability $\varsigma \in [0, 1]$
    Choose action $A(t)$ which listed in Eq. (12) as $A(t) = \begin{cases} a^*(t) = \arg\max_{a} Q(s, a; \theta), \varsigma > \varepsilon \\ a(t) \neq a^*(t), \textit{randomly select } a(t), \textit{others} \end{cases}$
    Based the action $A(t)$, execute the transcoding policy and the caching updated
    Observe the reward $r(t)$, state $s(t+1)$
    Store the transition $(s(t), A(t), r(t), s(t+1))$ in D
    Update MEC service matrix $V$ of requests
    Sample random minibatch of transitions
      $(s(t), A(t), r(t), s(t+1))$ from D
    Set $y_j = \begin{cases} r_j & \text{for terminal } s' \\ r_j + \gamma \max_{a'} Q(s', a'; \theta_{i-1})|s, a) & \text{for non-terminal } s' \end{cases}$
    Perform a gradient descent step according to equation: $\begin{matrix} L_i(\theta_i) = \mathbb{E}_{s,a\sim\rho(\cdot)}[(y_i - Q(s, a; \theta_i))^2] \\ y_i = r + \gamma \max_{a'} Q(s', a'; \theta_{i-1})|s, a) \end{matrix}$.
    Update the parameters in the Q network
    Reset the parameters in the target Q network every G time stages
  end for
end for
```

Algorithm 1: Deep reinforcement learning algorithm for collaborative video caching and transcoding (DRL-CCT).

6.1.2. Parameter Setting. In our experiments, we set 7 MEC severs in one cluster, which serve 30 users in this region and provide about 50 videos for users' requests. Then, we set $D = 10s$, $\beta = 6.5$, $\alpha = 1.3$, $\mu = 0.6$, $\sigma = 1.2$, the weights associated with cache hit rate and QoE in the reward function are set as $\omega_1 = 1$, $\lambda = 0.9$, $\omega_2 = 0.9$, $\omega_3 = 0.1$, and the weights associated with cost penalty in the reward function are set as $\omega_4 = 0.1$, $\omega_5 = 0.1$.

In the experiment, there are four video layers of the video, with $B_{\max} = 10$ Mbps as the highest layer at the MEC server. The bitrates of the three transcoded layers are $B_1 = 1$ Mbps, $B_2 = 2$ Mbps, and $B_3 = 4$ Mbps, and the set of available CPU cores at MEC is $\{2, 4, 6, 8\}$. Video transcoding from $B_{\max}$ to B_1, B_2, and B_3 needs 2, 4, and 6 CPU cycles, respectively. With the number of caching strategy being 2 (yes or no), the number of videos' bitrates being 4, and the number of MEC severs in one cluster being 7, the number of actions in action set A is $2 \times 4 \times 7 = 56$.

6.1.3. Deep Neural Network for DQN. We use a fully connected neural network with 2 hidden layers, 256 and 512 in size. The loss function is the mean square error. The naive ε-greedy strategy is used for exploration, and the probability of randomly choosing an action during training is ε. As the learning progresses, the degree of exploration continues to shrink. The learning rate is 0.01, the size of experience replay in DQN is 2000, the attenuation parameter used to update the target Q network is 0.9, and the batch size in stochastic batch gradient descent is 32. The experiments are implemented using Python and TensorFlow.

6.2. Simulation Results. In this section, we compare the proposed DRL algorithm (called DRL-CCT) with the latest baseline methods, such as the method (called caching only at network edge) in Reference [51] and the method (called transcoding only at network edge) in Reference [56]. In our experimental framework, we simulated the above methods according to the setting form of the reward function in the above literature. Also, we compare the proposed DRL algorithm with the algorithm of DRL-CCT without transcoding policy. Especially due to the characteristics of deep reinforcement learning, for our proposed algorithm, all reported results were obtained from average of 20 algorithm executions.

Figure 3 shows the convergence performance of the DRL-CCT algorithm under the set of full weight in the different learning rates. With continuous learning, the average reward gradually stabilizes. Compared with the balanced method of the algorithm in Reference [56], the average reward of the algorithm we proposed converges faster, and the subsequent fluctuations are slightly larger. But in contrast, the deep network used in our DRL-CCT algorithm is more concise and efficient. The convergence performance is influenced by learning rate. The performance of the learning rate 0.01 is better than the performance of the learning rates 0.1, 0.001, and 0.0001. The convergence performance becomes worse

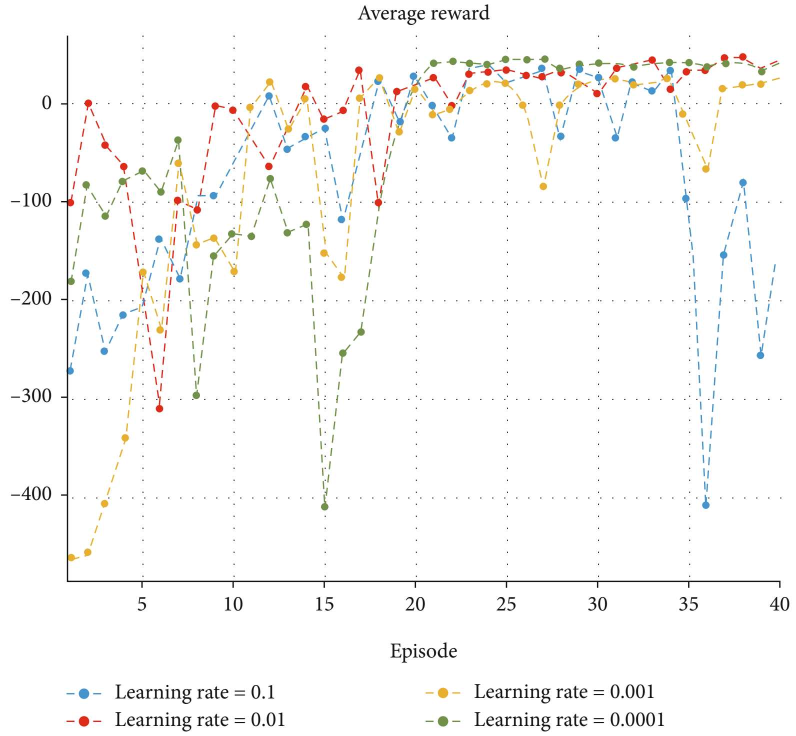

Figure 3: The convergence performance of DRL-CCT algorithm in the different learning rate.

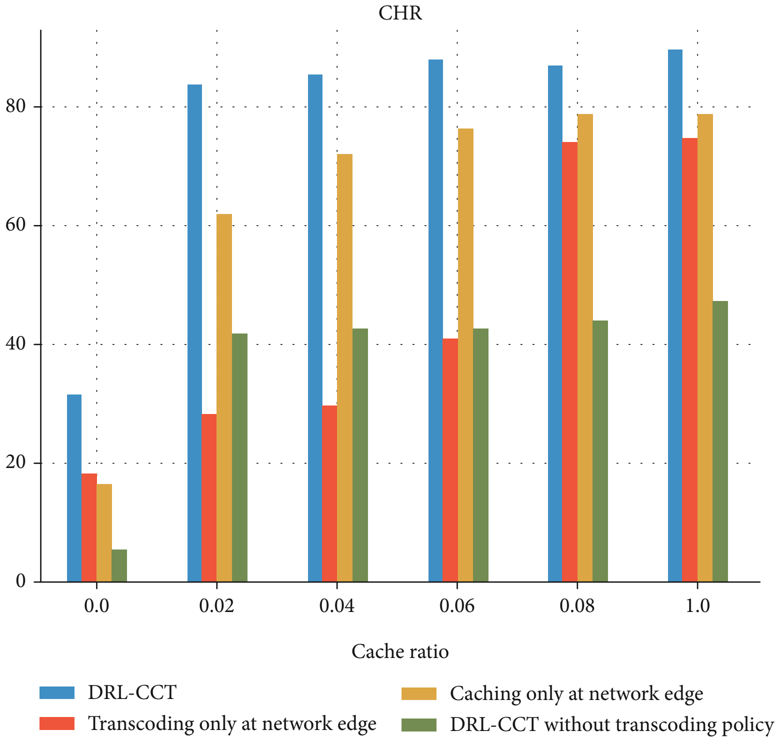

Figure 4: Cache hit rate vs. cache ratio.

in the learning rate 0.1, owing to a large update step such that the average reward converges to a local optimal solution. In fact, an appropriate learning rate depends on the state of the environment in the current optimization process.

Figure 4 gives the comparison of cache hit rates in different algorithms at the same cache ratios. Compared with the other algorithms, the DRL-CCT algorithm has a higher cache hit rate. Since the 2C-MEC system model has been proposed,

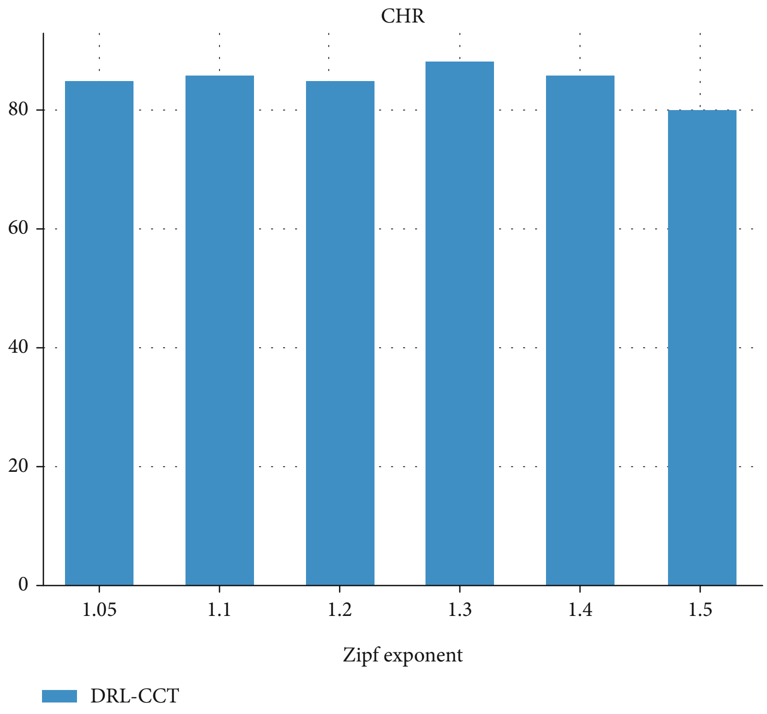

Figure 5: Cache hit rate vs. Zipf exponent.

the cluster-based video cache hit rate is definitely better than the video cache hit rate based on a single MEC server, especially when the cache ratio is relatively small. In addition, the performance in the cache hit rate of DRL-CCT without transcoding policy algorithm is the worst one because only the highest version of the video is cached in the MEC. Owing to the absence of transcoding function at network edge, the MEC server has to return to the source server for extraction when the user requests for other version of the video, which results in low cache hit rate.

In Figure 5, we study the cache hit rate as a function of the Zipf exponent. As Zipf exponent increases, cache hit rates achieved by the caching policy increase first and then decrease. This is due to the fact that with larger Zipf exponent, the video popularity distribution is more concentrated, and therefore, the popularity of the files is skewed. Consequently, caching these more popular videos leads to an increase first in the cache hit rates. Then, the cache hit rates have a fall. It is because that the DRL-CCT algorithm stores the most popular files initially when the number of popular files gets small. However, it eventually experiences diminishing returns as Zipf exponent is further increased, and the larger the Zipf exponent, the smaller the influence of less popular files is.

As for average QoE performance in Figure 6, DRL-CCT is much better than “transcoding only” and the other two algorithms. Due to the long rebuffering time, the average QoE value of the DRL-CCT without transcoding algorithm and “caching only” algorithm are below zero all the time. Compared with these methods which has no joint caching and transcoding at the edge, DRL-CCT has the highest QoE, which means users can get much better experience in video streaming services. It can be seen from Figure 7 that when there is no transcoding function at the network edge, the bandwidth cost is greater than the DRL-CCT algorithm, because the uncached video has to be extracted from the source server which leads to consume a lot of bandwidth cost. The difference of bandwidth cost performance between “transcoding only” algorithm and DRL-CCT algorithm is slight in the latter stage.

The average bandwidth cost and QoE performance in DRL-CCT algorithm with different experimental settings are shown in Figures 8–11. Figures 8 and 9 are the performance for different request numbers in an episode. It can be seen from Figure 8 that as the number of user requests in a time slot increases, the average bandwidth cost of each MEC will continue to increase. This is because the number of MEC servers is fixed. When the number of user requests has increased, the number of user requests served by each MEC must increase, which directly leads to an increase in the average bandwidth cost of each MEC. The following conclusions can be directly obtained in Figure 9 that the change in the number of requests from different users in a time slot does not have a great impact on the average QoE of the users, and the QoE value of the video streaming service is stable in a good range.

Then, Figures 10 and 11 are the performance for different MEC numbers within a cluster at network edge. According to Figure 10, on the premise that the number of user requests in a time slot is determined, when the number of MEC nodes in the edge cluster decreases, the average bandwidth cost of each MEC will increase at the beginning. However, as the deep reinforcement learning process progresses, the average bandwidth cost of each MEC will tend to stabilize. This is due to

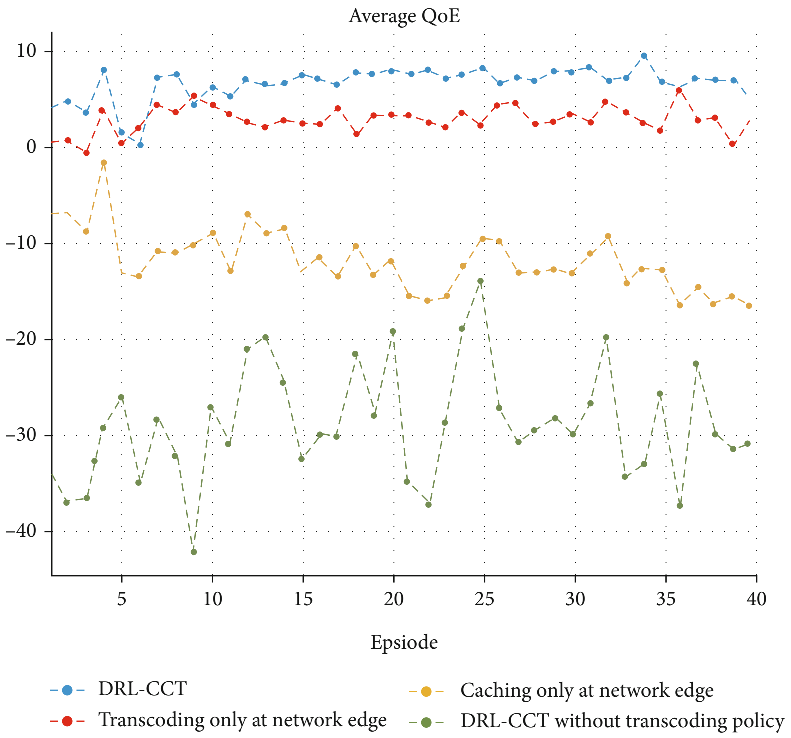

FIGURE 6: The QoE performance in different algorithms.

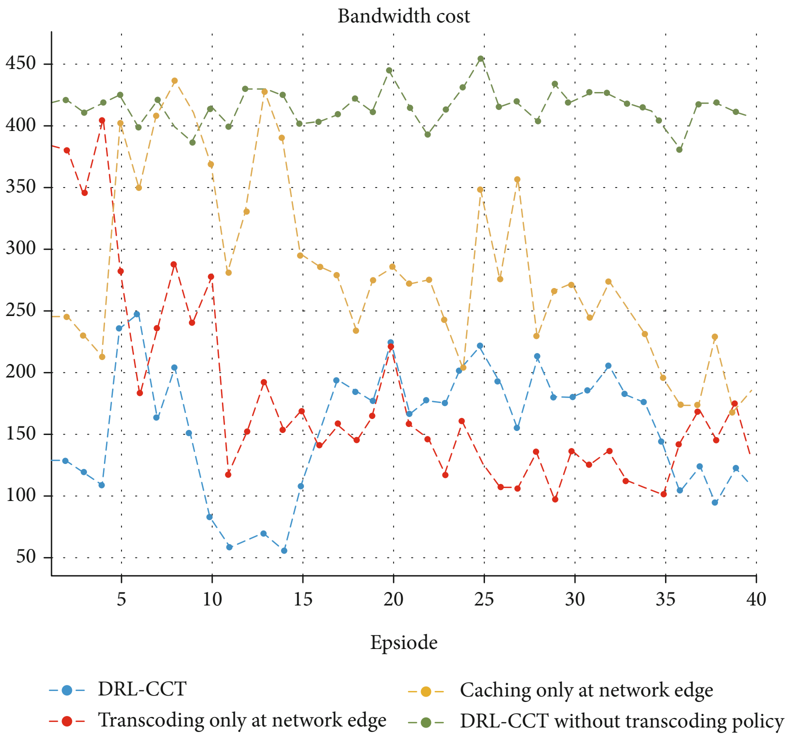

FIGURE 7: The bandwidth cost performance in different algorithms.

the adaptive decision-making function of deep reinforcement learning, which continuously optimizes the MEC load distribution in one edge cluster. In Figure 11, the same as in Figure 9, the average QoE performance of the system has always been relatively stable, indicating that the proposed method has excellent robustness to environmental changes.

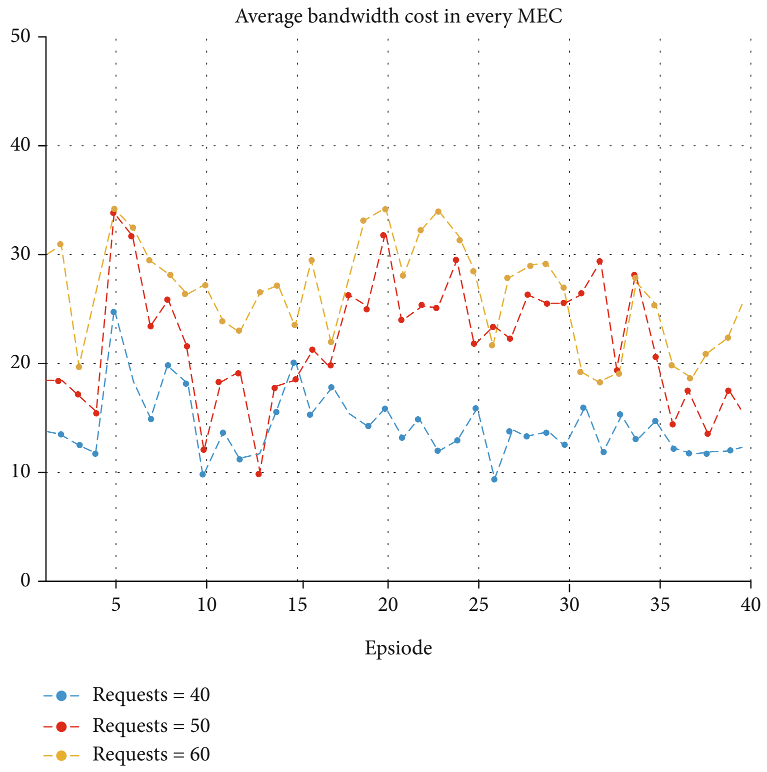

FIGURE 8: The average bandwidth cost in the DRL-CCT algorithm at different request numbers in an episode.

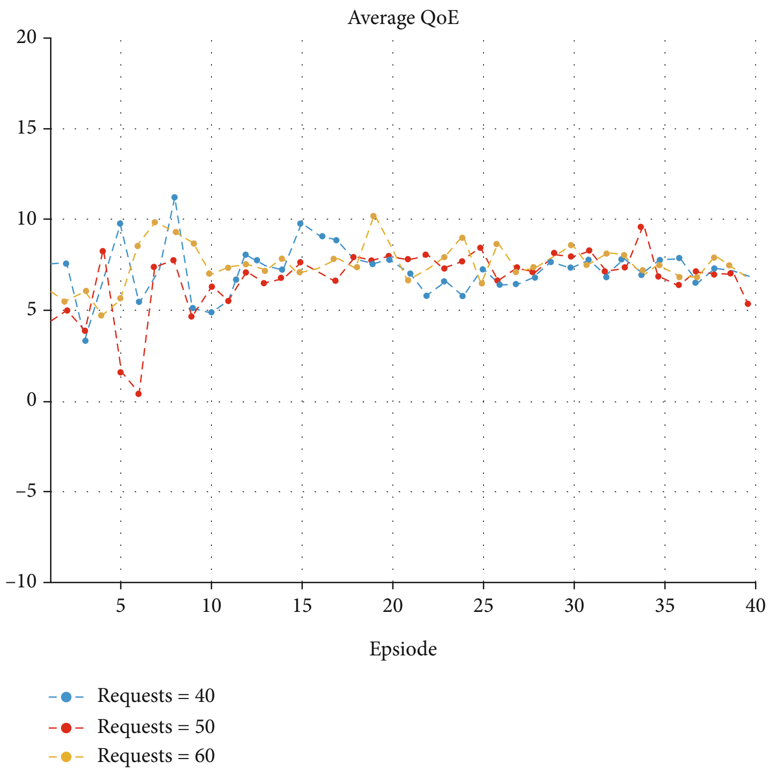

FIGURE 9: The average QoE performance in the DRL-CCT algorithm at different requests numbers in an episode.

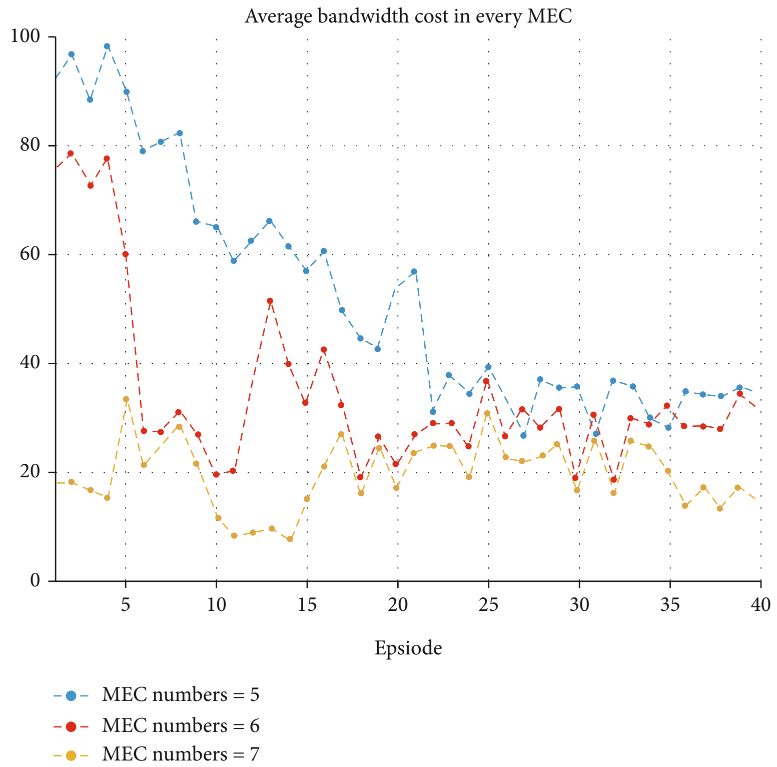

FIGURE 10: The average bandwidth cost in DRL-CCT algorithm at different MEC numbers with in a cluster.

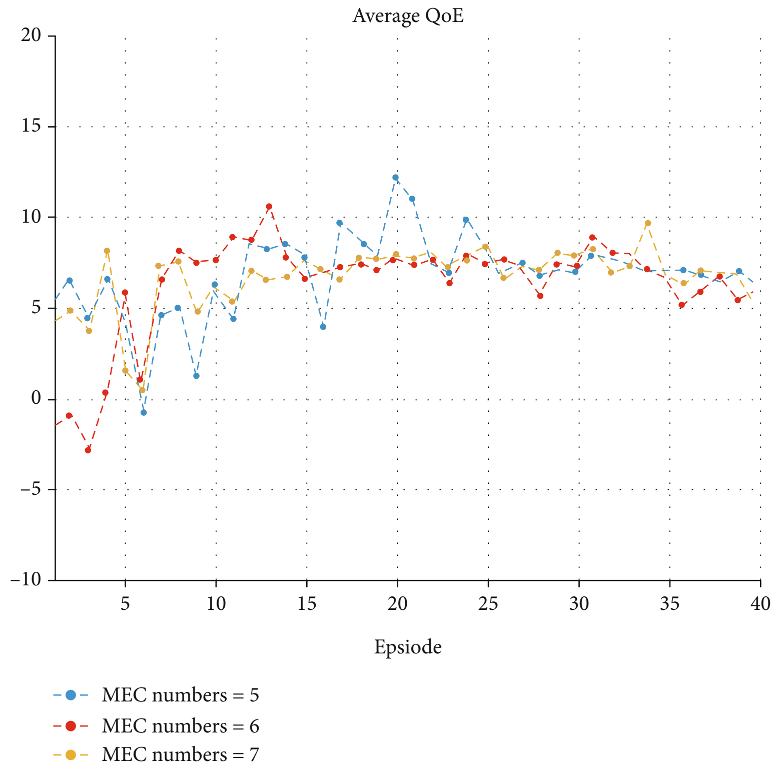

FIGURE 11: The average QoE performance in DRL-CCT algorithm at different MEC numbers with in a cluster.

7. Conclusions

In this paper, we first propose a CDN and Cluster-based Mobile Edge Computing system that can enhance the ability of caching and computing and promote the collaboration among MEC severs in one cluster. In addition, we formulate a novel deep reinforcement learning based framework to automatically obtain the intracluster collaborative caching and transcoding decisions, which are executed based on video popularity, user requirement prediction, and abilities of MEC servers. Then, numerical results are presented to validate the effectiveness of the proposed method.

Under the framework of the 2C-MEC system, this paper mainly researches on promoting the collaboration among MEC servers in the cluster. In the future work, intercluster collaboration needs to be considered when intracluster computing and storage capabilities are insufficient. If it is assumed that the terminal has caching and computing capabilities, it is also possible to consider "edge-end" collaboration, "end-end" collaboration, and other collaboration modes to implement a multidimensional collaboration model of "cloud-edge-end" among different agents. At the same time, load balancing among MEC servers in the mobile edge cluster still needs further research to explore efficient ways to solve the contradiction between the balance of MEC servers and the improvement of user QoE.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The work was supported by the National Natural Science Foundation of China (No. 61961021), the Science and Technology Project of Jiangxi Education Department (No. GJJ180251 and No. GJJ171011), and the Innovation Special Fund for Individual Graduate Student of Jiangxi University of Finance and Economics (2020 Annual, No. 24).

References

[1] N. Kato, B. Mao, F. Tang, Y. Kawamoto, and J. Liu, "Ten challenges in advancing machine learning technologies toward 6G," *IEEE Wireless Communications*, vol. 27, no. 3, pp. 96–103, 2020.

[2] K. Zhang, Y. Zhu, S. Maharjan, and Y. Zhang, "Edge intelligence and blockchain empowered 5G beyond for the industrial Internet of things," *IEEE Network Magazine*, vol. 33, no. 5, pp. 12–19, 2019.

[3] Y. Dai, D. Xu, S. Maharjan, G. Qiao, and Y. Zhang, "Artificial intelligence empowered edge computing and caching for internet of vehicles," *IEEE Wireless Communications Magazine*, vol. 26, no. 3, pp. 12–18, 2019.

[4] Cisco, *Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2017-2022 White Paper*, 2019, https://www.cisco.com/c/dam/m/en_in/innovation/enterprise/assets/mobile-white-paper-c11-520862.pdf.

[5] S. M. Azimi, O. Simeone, A. Sengupta, and R. Tandon, "Online edge caching and wireless delivery in fog-aided networks with dynamic content popularity," *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 6, pp. 1189–1202, 2018.

[6] G. Gao, Y. Wen, and J. Cai, "vcache: supporting cost-efficient adaptive bitrate streaming," *IEEE Multimedia*, vol. 24, no. 3, pp. 19–27, 2017.

[7] Y. C. Hu, M. Patel, D. Sabella, N. Sprecher, and V. Young, "Mobile edge computing-a key technology towards 5G," *ETSI White Paper*, vol. 11, 2015.

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

[9] A. Ahmed and E. Ahmed, "A survey on mobile edge computing," in *Proc. IEEE Int. Conf. On Intelligent Systems and Control (ISCO)*, Nanjing, China, 2016.

[10] J. Liu, Y. Mao, J. Zhang, and K. B. Letaief, "Delay-optimal computation task scheduling for mobile-edge computing systems," in *2016 IEEE International Symposium on Information Theory (ISIT)*, pp. 1451–1455, Barcelona, 2016.

[11] Y. Mao, J. Zhang, and K. B. Letaief, "Dynamic computation offloading for mobile-edge computing with energy harvesting devices," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 12, pp. 3590–3605, 2016.

[12] T. X. Tran, A. Hajisami, P. Pandey, and D. Pompili, "Collaborative mobile edge computing in 5g networks: new paradigms, scenarios, and challenges," *IEEE Communications Magazine*, vol. 55, no. 4, pp. 54–61, 2017.

[13] Y. Sánchez dela Fuente, T. Schierl, C. Hellge et al., "iDASH: improved dynamic adaptive streaming over HTTP using scalable video coding," *Proceeding of ACM Multimedia Systems*, pp. 23–25, 2011.

[14] A. Mehrabi, M. Siekkinen, and A. Ylä-Jääski, "Edge computing assisted adaptive mobile video streaming," *IEEE Transactions on Mobile Computing*, vol. 18, no. 4, pp. 787–800, 2019.

[15] D. Wang, Y. Peng, X. Ma et al., "Adaptive wireless video streaming based on edge computing: opportunities and approaches," *IEEE Transactions on Services Computing*, vol. 12, no. 5, pp. 685–697, 2019.

[16] T. X. Tran and D. Pompili, "Adaptive bitrate video caching and processing in mobile-edge computing networks," *IEEE Transactions on Mobile Computing*, vol. 18, no. 9, pp. 1965–1978, 2019.

[17] J. Yao, T. Han, and N. Ansari, "On mobile edge caching," *IEEE Communications Surveys & Tutorials*, vol. 21, 2019.

[18] S. Safavat, N. N. Sapavath Naveen, and D. B. Rawat, "Recent advances in mobile edge computing and content caching," *Digital Communications and Networks*, vol. 6, 2020.

[19] K. He, Z. Wang, W. Huang, D. Deng, J. Xia, and L. Fan, "Generic deep learning-based linear detectors for MIMO systems over correlated noise environments," *IEEE Access*, vol. 8, pp. 29922–29929, 2020.

[20] J. Xia, L. Fan, W. Xu et al., "Secure cache-aided multi-relay networks in the presence of multiple eavesdroppers," *IEEE Transactions on Communications*, vol. 67, no. 11, pp. 7672–7685, 2019.

[21] H. Liu, C. Lin, J. Cui, L. Fan, X. Xie, and B. F. Spencer, "Detection and localization of rebar in concrete by deep learning

using ground penetrating radar," *Automation in Construction*, vol. 118, 2020.

[22] K. He, Z. Wang, D. Li, F. Zhu, and L. Fan, "Ultra-reliable MU-MIMO detector based on deep learning for 5G/B5G-enabled IoT," *Physical Communication*, vol. 43, p. 101181, 2020.

[23] S. S. Mousavi, M. Schukat, and E. Howley, "Deep reinforcement learning: an overview," in *Proceedings of SAI Intelligent Systems Conference*, pp. 426–440, Cham, 2016.

[24] S. S. Mousavi, M. Schukat, and E. Howley, *Deep Reinforcement Learning: An Overview*, Intelligent Systems Conference 2018 (IntelliSys 2018), London, United Kingdom, 2018.

[25] Y. He, Z. Zhang, F. R. Yu et al., "Deep-reinforcement-learning-based optimization for cache-enabled opportunistic interference alignment wireless networks," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 11, pp. 10433–10445, 2017.

[26] X. Wang, Y. Han, V. C. M. Leung, D. Niyato, X. Yan, and X. Chen, "Convergence of edge computing and deep learning: a comprehensive survey," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 2, pp. 869–904, 2020.

[27] D. Guo, L. Tang, X. Zhang, and Y. Liang, "Joint optimization of handover control and power allocation based on multi-agent deep reinforcement learning," *IEEE Transactions on Vehicular Technology*, vol. 69, 2020.

[28] S. Lai, "Intelligent secure mobile edge computing for beyond 5G wireless networks," *Physical Communication*, vol. 99, pp. 1–8, 2020.

[29] R. Zhao, "Deep reinforcement learning based mobile edge computing for intelligent Internet of things," *Physical Communication*, vol. 43, article 101184, 2020.

[30] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, MIT Press, Cambridge, MA, 2018.

[31] V. Mnih, K. Kavukcuoglu, D. Silver et al., "Playing atari with deep reinforcement learning," *NIPS Deep Learning Workshop*, 2013.

[32] V. Mnih, K. Kavukcuoglu, D. Silver et al., "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[33] B. Guo, X. Zhang, Y. Wang, and H. Yang, "Deep-Q-network-based multimedia multi-service QoS optimization for mobile edge computing systems," *IEEE Access*, vol. 7, pp. 160961–160972, 2019.

[34] I. AlQerm and B. Shihada, "Energy efficient power allocation in multi-tier 5g networks using enhanced online learning," *IEEE Transactions on Vehicular Technology*, vol. 66, 2017.

[35] X. He, K. Wang, H. Huang, T. Miyazaki, Y. Wang, and S. Guo, "Green resource allocation based on deep reinforcement learning in content-centric iot," *IEEE Transactions on Emerging Topics in Computing*, vol. 8, 2020.

[36] Y. S. Nasir and D. Guo, "Multi-agent deep reinforcement learning for dynamic power allocation in wireless networks," *IEEE Journal on Selected Areas in Communications*, vol. 37, 2019.

[37] H. TY, N. Zhao, and H. Yin, "Integrated networking, caching and computing for connected vehicles: a deep reinforcement learning approach," *IEEE Transactions on Vehicular Technology*, vol. 99, no. 10, 2017.

[38] J. Li, H. Gao, T. Lv, and Y. Lu, "Deep reinforcement learning based computation offloading and resource allocation for mec," *IEEE Wireless Communications and Networking Conference (WCNC)*, 2018, pp. 1–6, Barcelona, Spain, 2018.

[39] L. Huang, S. Bi, and Y. J. Zhang, "Deep reinforcement learning for online computation offloading in wireless powered mobile-edge computing networks," *IEEE Transactions on Mobile Computing*, vol. 99, 2018.

[40] X. Chen, H. Zhang, C. Wu, S. Mao, Y. Ji, and M. Bennis, "Optimized computation offloading performance in virtual edge computing systems via deep reinforcement learning," *IEEE Internet of Things Journal*, vol. 6, 2018.

[41] S. Park, J. Kim, D. Kwon, M. Shin, and J. Kim, "Joint offloading and streaming in mobile edges: a deep reinforcement learning approach," in *2019 IEEE VTS Asia Pacific Wireless Communications Symposium (APWCS)*, Singapore, Singapore, 2019.

[42] H. Zhang, W. Wu, C. Wang, M. Li, and R. Yang, "Deep reinforcement learning-based offloading decision optimization in mobile edge computing," in *2019 IEEE Wireless Communications and Networking Conference (WCNC)*, Marrakesh, Morocco, Morocco, 2019.

[43] Z. Cheng and Z. Zheng, "Task migration for mobile edge computing using deep reinforcement learning," *Future Generation Computer Systems*, vol. 96, pp. 111–118, 2019.

[44] S. Wang, X. Zhang, Y. Zhang, L. Wang, J. Yang, and W. Wang, "A survey on mobile edge networks: convergence of computing, caching and communications," *IEEE Access*, vol. 5, no. 3, pp. 6757–6779, 2017.

[45] Z. Zhang, Y. Zheng, C. Li, Y. Huang, and L. Yang, "Cache-enabled adaptive bit rate streaming via deep self-transfer reinforcement learning," in *2018 10th International Conference on Wireless Communications and Signal Processing (WCSP) IEEE*, Hangzhou, China, 2018.

[46] L. Lei, L. You, G. Dai, T. X. Vu, D. Yuan, and S. Chatzinotas, "A deep learning approach for optimizing content delivering in cache-enabled hetnet," in *Wireless Communication Systems (ISWCS)*, pp. 449–453, Bologna, Italy, 2017.

[47] L. Lei, X. Xiong, H. Lu, and K. Zheng, "Collaborative edge caching through service function chaining: architecture and challenges," *IEEE Wireless Communications*, vol. 25, no. 3, pp. 94–102, 2018.

[48] C. H. Wei, Y. W. Hung, and F. L. Chin, "Q-learning based collaborative cache allocation in mobile edge computing," *Future Generation Computer Systems*, vol. 102, pp. 603–610, 2020.

[49] Z. Yang, Y. Liu, Y. Chen, and G. Tyson, "Deep reinforcement learning in cache-aided MEC networks," in *ICC 2019-2019 IEEE International Conference on Communications (ICC)*, Shanghai, China, 2019.

[50] Z. Chen, M. C. Gursoy, and S. Velipasalar, "A deep reinforcement learning-based framework for content caching," in *2018 52nd Annual Conference on Information Sciences and Systems (CISS)*, Princeton University, NJ, USA, 2018.

[51] C. Zhong, M. Cenk Gursoy, and S. Velipasalar, "Deep reinforcement learning based edge caching in wireless networks," *IEEE Transactions on Cognitive Communications and Networking*, vol. 6, no. 1, pp. 48–61, 2020.

[52] M. C. Gursoy, C. Zhong, and S. Velipasalar, "Deep multi-agent reinforcement learning for cooperative edge caching," in *Machine Learning for Future Wireless Communications*, pp. 439–457, Shanghai, China, China, 2020.

[53] L. Liu, H. Hu, Y. Luo, and Y. Wen, "When wireless video streaming meets AI: a deep learning approach," *IEEE Wireless Communications*, vol. 27, 2019.

[54] H. Zhang, L. Dong, G. Gao, H. Hu, Y. Wen, and K. Guan, "DeepQoE: a multimodal learning framework for video quality

of experience (QoE) prediction," *IEEE Transactions on Multimedia*, vol. 22, 2020.

[55] G. Gao, L. Dong, H. Zhang, Y. Wen, and W. Zeng, "Content-aware personalised rate adaptation for adaptive streaming via deep video analysis," in *ICC 2019-2019 IEEE International Conference on Communications (ICC)*, Shanghai, China, China, 2019.

[56] Y. Guo, F. R. Yu, J. An, K. Yang, C. Yu, and V. C. M. Leung, "Adaptive bitrate streaming in wireless networks with transcoding at network edge using deep reinforcement learning," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 4, pp. 3879–3892, 2020.

[57] M. Liu, Y. Teng, F. R. Yu, V. C. M. Leung, and M. Song, "A deep reinforcement learning-based transcoder selection framework for blockchain-enabled wireless D2D transcoding," *IEEE Transactions on Communications*, vol. 68, no. 6, pp. 3426–3439, 2020.

[58] F. Wang, C. Zhang, F. Wang et al., "Intelligent edge-assisted crowdcast with deep reinforcement learning for personalized QoE," in *IEEE INFOCOM 2019*, Paris, France, 2019.

[59] Z. Pang, L. Sun, T. Huang, Z. Wang, and S. Yang, "Towards QoS-aware cloud live transcoding: a deep reinforcement learning approach," in *2019 IEEE International Conference on Multimedia and Expo (ICME)*, pp. 670–675, Shanghai, China, China, 2019.

[60] T. Y. Huang, R. Johari, N. Mc Keown, M. Trunnell, and M. Watson, "A buffer-based approach to rate adaptation: evidence from a large video streaming service," *acm special interest group on data communication*, vol. 44, no. 4, pp. 187–198, 2015.

[61] M. Chen, M. Ponec, S. Sengupta, J. Li, and P. A. Chou, "Utility maximization in peer-to-peer systems with applications to video conferencing," *IEEE ACM Transactions on Networking*, vol. 20, no. 6, pp. 1681–1694, 2012.

[62] Y. Zheng, D. Wu, Y. Ke, C. Yang, M. Chen, and G. Zhang, "Online cloud transcoding and distribution for crowdsourced live game video streaming," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 27, no. 8, pp. 1777–1789, 2017.

Hindawi
Wireless Communications and Mobile Computing
Volume 2020, Article ID 8836592, 11 pages
https://doi.org/10.1155/2020/8836592

Research Article

A Smart Cache Content Update Policy Based on Deep Reinforcement Learning

Lincan Li,[1] Chiew Foong Kwong,[1] Qianyu Liu,[2] and Jing Wang[1]

[1] *Department of Electrical and Electronic Engineering, University of Nottingham Ningbo China, 315100 Ningbo, China*
[2] *International Doctoral Innovation Centre, University of Nottingham Ningbo China, 315100 Ningbo, China*

Correspondence should be addressed to Chiew Foong Kwong; chiew-foong.kwong@nottingham.edu.cn

Received 10 May 2020; Revised 10 October 2020; Accepted 15 October 2020; Published 9 November 2020

Academic Editor: Lisheng Fan

This paper proposes a DRL-based cache content update policy in the cache-enabled network to improve the cache hit ratio and reduce the average latency. In contrast to the existing policies, a more practical cache scenario is considered in this work, in which the content requests vary by both time and location. Considering the constraint of the limited cache capacity, the dynamic content update problem is modeled as a Markov decision process (MDP). Besides that, the deep Q-learning network (DQN) algorithm is utilised to solve the MDP problem. Specifically, the neural network is optimised to approximate the Q value where the training data are chosen from the experience replay memory. The DQN agent derives the optimal policy for the cache decision. Compared with the existing policies, the simulation results show that our proposed policy is 56%–64% improved in terms of the cache hit ratio and 56%–59% decreased in terms of the average latency.

1. Introduction

The recent rapid evolution of mobile communication techniques and the proliferation of smart mobile devices have caused an exponential growth in mobile network traffic [1] [2]. According to Cisco [3], global mobile network traffic will reach 77 exabytes each month by 2022. As such, it will lead to data traffic congestion of the backhaul [4]. To mitigate this, a cache-enabled technique has emerged that is regarded as an effective method that can alleviate data traffic congestion [5]. In a cache-enabled network, a portion of the popular content is cached at the edge of the network at base stations (BSs) or user terminals (UTs), where users can directly access and download the cached content from the edge rather than from the core network via backhaul links. Consequently, data traffic congestion of the backhaul can be reduced and content retrieval from the edge can be faster than from the remote core network [6, 7].

However, because of the limited cache capacity, it is necessary to update cache content to ensure that cache-enabled networks always store the most popular content [8]. The most two common content update policies are the least frequently used (LFU) policy and the least recently used (LRU) policy [9]. LRU frequently stores the content with the latest access time, and LFU frequently stores content with the largest cumulative request times. Besides, as described in [10], a heterogeneous cache structure is proposed, in which the most popular contents are stored at small BSs and the less popular contents are stored at macro BSs. The combination of small BSs and macro BSs can maximise the network capacity and satisfy the content transmission demand. In [11], an optimal cooperative cache policy that can increase the cache hit ratio was presented. The cache hit ratio is utilised to describe how frequently content is requested by mobile users. In [9], an adaptive cache policy was proposed that can reduce user access latencies. In [12], an edge cache policy was proposed to reduce the average content delivery latency. However, conventional methods lack adaptive ability in dynamic cache scenarios. The reason is that they assume that the content popularity distribution is known or can be accurately predicted, which is difficult to achieve in dynamic caching scenarios. In this case, due to an inaccurate distribution of content popularity, the conventional methods have poor cache performances, since their performances are highly dependent on the accurate distribution of content popularity.

Motivated by the deep reinforcement learning (DRL) approach in solving the dynamic problem [13], DRL has been applied into cache policies to improve the cache performance of dynamic cache scenarios. In [14], a DRL approach was proposed to reduce the transmission cost by jointly considering proactive cache and content recommendations. In [15], a cache content update policy based on DRL was proposed to improve energy efficiency. In [16], a DRL model was utilised to minimise transmission latencies. For specific, reinforcement learning (RL) is applied to obtain the optimal cache policy. In [17], a DRL-based policy was proposed to minimise system power consumption. In [18], a deep Q-learning network (DQN) algorithm, one branch of DRL, is applied to do the network slicing decision and allocates the spectrum resources for the content delivery. In [19], a DQN-based mobile edge computing network is proposed, in which several computation tasks are offloaded from the user terminals to the computational access points. Although DQN has attracted significant attention in the cache-enabled network, there are very little works done in applying DQN into the cache content update phase. Moreover, most of the previously mentioned DRL-based cache policies assume the content requests as a time-varying variable. They did not adopt more practical scenarios in which the content requests are varied in both time and location, also known as spatiotemporally varying scenarios.

Inspired by the aforementioned literature, in this paper, a DQN-based content update policy at BSs is proposed to increase the cache hit ratio and reduce average latency, as well as considering spatiotemporally varying scenarios in which content requests vary by both time and location. The reasons to apply DQN are as follows: (1) DQN has a faster convergence speed than the conventional DRL policies, e.g., advanced actor-critic (A2C) and deep deterministic policy gradient (DDPG) [14]. (2) DQN can adapt to the varying scenarios, as long as the dynamic problem is correctly modeled and the DQN agent is allowed to continuously learn experience from the environment [18]. The main contributions are summarised as follows:

(i) The dynamic cache content update problem is formulated as a Markov decision process (MDP) problem, which is solved by a DQN algorithm. Specifically, the neural network is utilised to approximate the Q value and the DQN agent is used to decide whether or not to cache the requested content

(ii) Our proposed policy is compared with LRU, LFU and DRL [20] policies and the simulation results demonstrate that our proposed policy has the best cache performance in terms of the cache hit ratios and average latencies

The rest of this paper is organised as follows. The system model and problem formulation are introduced in Section 2. The detailed elements of the MDP framework and the principles of the DQN-based cache content update policy are discussed in Section 3. The simulation results are shown in Section 4, and the conclusion is provided in Section 5.

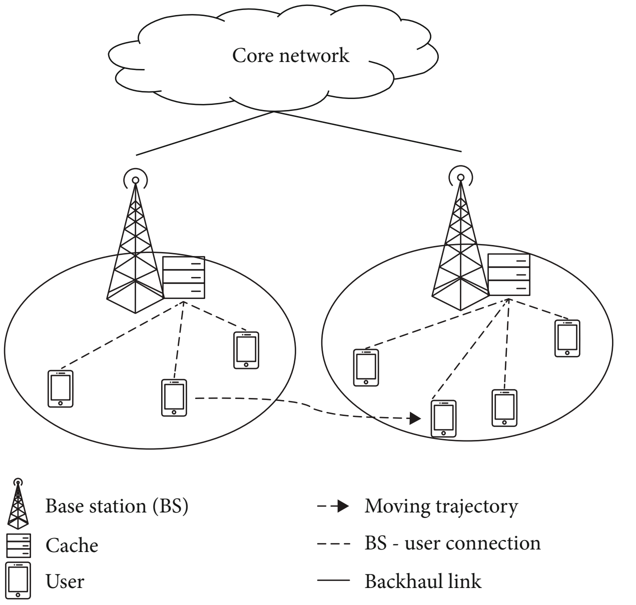

FIGURE 1: The system model of the cache-enabled network.

2. System Model and Problem Formulation

In this section, the system model and the problem of how to maximise the cache hit ratio and minimise average latency are introduced.

2.1. System Model. As shown in Figure 1, the cache-enabled system includes one core network, $\mathscr{M}$ cache-enabled BSs, and $\mho$ mobile users. Each BS can store $\mathscr{H}$ contents at most. The total content library $\mathscr{W} = \{1, 2, \cdots, \omega\}$ contains ω kinds of contents and each content has the same size $\mathrm{Size}_{\mathrm{df}}$. The core network is assumed that has enough capacity to store the entire contents. Each BS covers a circular cellular region with a fixed radius, and all of the mobile users in its cellular region can connect with the serving BS (the BS where users connect). Mobile users can directly retrieve their requested content from the serving BS if the content is cached locally (the requested content is already cached at the serving BS); otherwise, the requested content must be retrieved from the core network. The i^{th} BS is regarded as a DQN agent and receives the spatiotemporal content requests $R^i = \{R_1^i, R_2^i, \cdots, R_t^i, \cdots\}$, where R_t^i is the current content request at the i^{th} BS. From the received content requests, the DQN agent can decide when and where (which BS) to cache the content or not. If cached, the DQN agent further decides which cached content is replaced by the currently requested content; otherwise, the cached contents remain the same. The action space of the i^{th} BS is defined as $A^i = \{A_0^i, A_1^i, A_2^i, \cdots, A_{\mathscr{H}}^i\}$ and A^i uses one hot code. $A_0^i = 1$ means that the cached content remains the same, and $A_v^i = 1$ means that the v^{th} cached content is replaced by the currently requested content, where $v \epsilon \{1, 2, \cdots, \mathscr{H}\}$. In summary, at each time slot

t, each BS receives numerous content requests including the user preference content and location information, and each DQN agent executes one action from the corresponding action space to maximise the cache hit ratio and minimise the average latency.

2.2. Problem Formulation. The problem in this study consists of two subproblems: maximising the cache hit ratio and minimising average latency.

2.2.1. Maximising the cache hit ratio. The cache hit ratio is utilised to describe the probability of the requested content at the local cache. The system cache hit ratio $\mathscr{P}_{\text{hit\_ratio}}$ is formulated for N requests as follows:

$$\mathscr{P}_{\text{hit\_ratio}} = \frac{\sum_{i=1}^{\mho}\sum_{t=1}^{N} F\left(R_t^i\right)}{\mho \times N}, \tag{1}$$

where $F(R_t^i)$ is a function to test whether the requested content is cached locally. The definition of $F(R_t^i)$ is as follows:

$$F\left(R_t^i\right) = \begin{cases} 1, \text{if } R_t^i \text{ is cached locally} \\ 0, \text{otherwise} \end{cases} \tag{2}$$

Maximising the cache hit ratio is expressed as follows:

$$\begin{aligned} &\boldsymbol{P\_1} : \text{Max } \mathscr{P}_{\text{hit\_ratio}} \\ &s.t. \sum_{t=1}^{T} F\left(R_t^i\right) \leq \mathscr{H} \end{aligned} \tag{3}$$

2.2.2. Minimising the average latency. The latency is an indicator that evaluates the cache content update policy's performance. The latency is the time when content is transmitted from one location to another. The latency consists of the transmission latency $\mathscr{T}_{\text{tr}}$, propagation latency $\mathscr{T}_{\text{prp}}$, processing latency $\mathscr{T}_{\text{pro}}$, and queue latency $\mathscr{T}_{\text{qu}}$. From [20], the expression of the latency $\mathscr{T}$ is given as:

$$\mathscr{T} = \mathscr{T}_{\text{tr}} + \mathscr{T}_{\text{prp}} + \mathscr{T}_{\text{pro}} + \mathscr{T}_{\text{qu}}. \tag{4}$$

Normally in the content update process, the destination of the content packet is determinate, and the content packet is assumed that does not need to wait for transmission. Hence, the processing and queue latencies can be neglected during the content update process [20, 21], and the expression of the latency can be optimised as follows:

$$\begin{aligned} \mathscr{T}_{\text{tr}} &= \frac{\text{Size}_{\text{df}}}{v_{\text{tr}}}, \\ \mathscr{T}_{\text{prp}} &= D^* \times \frac{d}{\mathscr{R}}, \\ \mathscr{T} = \mathscr{T}_{\text{tr}} + \mathscr{T}_{\text{prp}} &= \frac{\text{Size}_{\text{df}}}{v_{\text{tr}}} + D^* \times \frac{d}{\mathscr{R}}, \end{aligned} \tag{5}$$

where Size_{df} is the content size, v_{tr} is the content transmission rate, $\mathscr{R}$ is the maximal coverage radius of the serving BS or core network, d is the distance between the user and the serving BS or between the serving BS and the core network, and D^* is the maximal propagation latency between the user and the serving BS or between the serving BS and the core network. To meet the requirement of the fifth-generation (5G) communication [22], the indicator $D^*$is expressed as follows:

$$D^* = \begin{cases} {D^*}_{\text{user}-\text{BS}} = 0.5 \sim 1.5\,\text{ms, if it is a user} - \text{BS connection} \\ {D^*}_{\text{BS}-\text{core}} = 10 \sim 20\,\text{ms, if it is a BS} - \text{core network connection} \end{cases}, \tag{6}$$

where ${D^*}_{\text{user}-\text{BS}}$ is the maximal propagation latency between the user and the serving BS, and ${D^*}_{\text{BS}-\text{core}}$ is the maximal propagation latency between the serving BS and the core network.

In more detail, if the requested content is cached locally, the content can be directly retrieved from the serving BS. Thus, for a hit content request, we consider the maximal propagation latency between the user and the serving BS ${D^*}_{\text{user}-\text{BS}}$, the distance between the user and the serving BS $d_{\text{user}-\text{BS}}$, and the maximal coverage radius of the serving BS $\mathscr{R}_{\text{BS}}$. The definition of the hit content $\mathscr{T}_{\text{hit}}$ latency is as follows:

$$\mathscr{T}_{\text{hit}} = \frac{\text{Size}_{\text{df}}}{v_{\text{tran}}} + {D^*}_{\text{user}-\text{BS}} \times \frac{d_{\text{user}-\text{BS}}}{\mathscr{R}_{\text{BS}}}. \tag{7}$$

If the requested content is missed at the serving BS, the serving BS needs to first retrieve the requested content from the core network and then deliver the requested content to the corresponding user. Hence, for a missed content request, we consider the maximal propagation latency between the user and the serving BS ${D^*}_{\text{user}-\text{BS}}$, the maximal propagation latency between the serving BS and the core network ${D^*}_{\text{BS}-\text{core}}$, the distance between the user and the serving BS $d_{\text{user}-\text{BS}}$, the distance between the serving BS and the core network $d_{\text{BS}-\text{core}}$, the maximal coverage radius of the serving BS $\mathscr{R}_{\text{BS}}$, and the maximal coverage radius of the core network $\mathscr{R}_{\text{core}}$. The definition of the latency of missed content $\mathscr{T}_{\text{miss}}$ is as follows:

$$\mathscr{T}_{\text{miss}} = \frac{\text{Size}_{\text{df}}}{v_{\text{tran}}} + {D^*}_{\text{user}-\text{BS}} \times \frac{d_{\text{user}-\text{BS}}}{\mathscr{R}_{\text{BS}}} + {D^*}_{\text{BS}-\text{core}} \times \frac{d_{\text{BS}-\text{core}}}{\mathscr{R}_{\text{core}}}. \tag{8}$$

The system latency $\mathscr{T}_{\text{sy}}$ is the sum of the latency of all of the hit content requests and all of the missed content requests. The average latency $\mathscr{T}_{\text{ave}}$ is the system latency divided by the number of content requests E. The $\mathscr{T}_{\text{sy}}$ and $\mathscr{T}_{\text{ave}}$ are defined as follows:

$$\begin{aligned} \mathscr{T}_{\text{sy}} &= \left[E \times \mathscr{P}_{\text{hit}_{\text{ratio}}} \times \mathscr{T}_{\text{hit}} + E \times \left(1 - \mathscr{P}_{\text{hit}_{\text{ratio}}}\right) \times \mathscr{T}_{\text{miss}}\right], \\ \mathscr{T}_{\text{ave}} &= \left[\mathscr{P}_{\text{hit}_{\text{ratio}}} \times \mathscr{T}_{\text{hit}} + \left(1 - \mathscr{P}_{\text{hit}_{\text{ratio}}}\right) \times \mathscr{T}_{\text{miss}}\right]. \end{aligned} \tag{9}$$

The problem on how to minimise the average latency can be formulated as follows:

$$\begin{aligned} \boldsymbol{P}_2 &: \text{Min } \mathscr{T}_{\text{ave}}, \\ s.t.&\mathscr{P}_{\text{hit\_ratio}} \in [0, 1]. \end{aligned} \tag{10}$$

3. A Deep Q-Learning Network-Based Cache Content Update Policy

The related elements of the deep Q-learning network will be introduced in Section 3.1. The principle of the DQN algorithm and the workflow of our proposed cache policy will be provided in Section 3.2.

3.1. The Description of the Related Elements of the Deep Q-Learning Network. The principle of the DQN can be regarded as a Markov decision process (MDP) [23, 24]. To apply the DQN to the cache content update problem, the related notations under the DQN framework are described.

3.1.1. State Space. In time slot t, the instant state consists of the currently cached content, the currently requested content and its corresponding user, the user's next location, and the current time. In time slot t, the current instant state s_t is defined as

$$s_t = \left\{c_t^i, R_t^i, j_t, L_t^j\right\}, \tag{11}$$

where c_t^i is the cached content at the i^{th} DQN agent, R_t^i is the currently requested content, j_t is the unique name of the mobile user currently requesting the content, L_t^j is the next location of the j^{th} user, $i \in \{1, 2, \cdots, \mathscr{M}\}$, and $j \in \{1, 2, \cdots, \mho\}$.

The state space S is the set of all of the instant states over a time period. It is defined as

$$S = \{s_0, s_1, s_2, \cdots, s_t, \cdots\}. \tag{12}$$

3.1.2. Action Space. In each time slot t, the i^{th} DQN agent decides whether or not to cache the currently requested content. If yes, the DQN agent decides which cached content is replaced by the currently requested content; otherwise, the cached content remains the same. The action space of the i $^{\text{th}}$ DQN A^i is defined as

$$A^i = \left\{A_0^i, A_1^i, A_2^i, \cdots, A_{\mathscr{H}}^i\right\}, \tag{13}$$

where A^i uses one hot code, which means only one action can be executed in a time slot. In this study, $A_0^i = 1$ means the cached content remains the same and $A_v^i = 1$ means the v^{th} cached content is replaced by the currently requested content, where $v \epsilon \{1, 2, \cdots, \mathscr{H}\}$ and $\mathscr{H}$ is the maximal capacity of the i^{th} BS.

3.1.3. Reward and Value Functions. The reward r_t is the instant cache hit ratio in the time slot t. Specifically, reward $r_t = 1$ when the currently requested content is hit in the next state s_{t+1}; otherwise, $r_t = 0$. The policy $\pi(s) = \mathscr{P}(a \mid s_t)$ is a map that shows the probability of the execution of action a_t under the current state s_t, and $a_t \in A^i$. The MDP evaluates and optimises the policy based on the value function, which is defined as the expected value of cumulative discounted rewards received over the entire process following the policy [25]. There are two definitions of value functions: one is the state value function and the other is the state-action value function. The state value function is the expected value of a discounted cumulative reward in the current state s_t when the agent follows the policy. The state value function is defined as follows:

$$V_\pi(s) = E_\pi\left[\sum_{u=0}^{\infty} \gamma^u r_{t+u+1} \mid s_t\right]. \tag{14}$$

The state-action value function is the expected value of the discounted cumulative reward from the current state s_t and action a_t is based on the policy used to choose one action. The definition of the state-action value function is

$$Q_\pi(s, a) = E_\pi\left[\sum_{u=0}^{\infty} \gamma^u r_{t+u+1} \mid (s_t, a_t)\right], \tag{15}$$

where $\gamma \epsilon [0, 1]$ is a discount factor that affects the future reward from the current state s_t. The target of the MDP is finding the optimal $\pi(s)$ and $\pi^*(s)$ that can obtain the maximal value function.

3.2. The Cache Content Update Based on the Deep Q-Learning Network

3.2.1. Principle of the DQN Framework. DQN is an effective hybrid framework of neural networks and Q-learning. In this framework, the neural network is applied to predict the Q values rather than recording the Q values in a Q table. However, the DQN will not be efficient when considering only the combination of Q-learning and the neural network. The following two characteristics improve the DQN framework's efficiency.

(i) The DQN has two neural networks with the same structures operating in different parameters, the evaluation network and the target network. The parameters of the evaluation and target networks are defined as θ and θ^-, respectively. The evaluation network uses the latest parameter θ to predict the current state-action Q values $Q_{(s_t, a_t, \theta)}$, where θ is updated in each iteration. The target network uses the parameter θ^- to predict the next state-action Q value $Q_{(s_{t+1}, a_{t+1}, \theta^-)}$, where θ^- is updated over a period time. The target network can solve the correlation of the Q value with the Q target value, which makes the DQN easier to converge

(ii) DQN has an experience replay memory with a limited capacity. The current state s_t, action a_t, reward r_t, and next state s_{t+1} are stored in format $(s_t$, a_t,

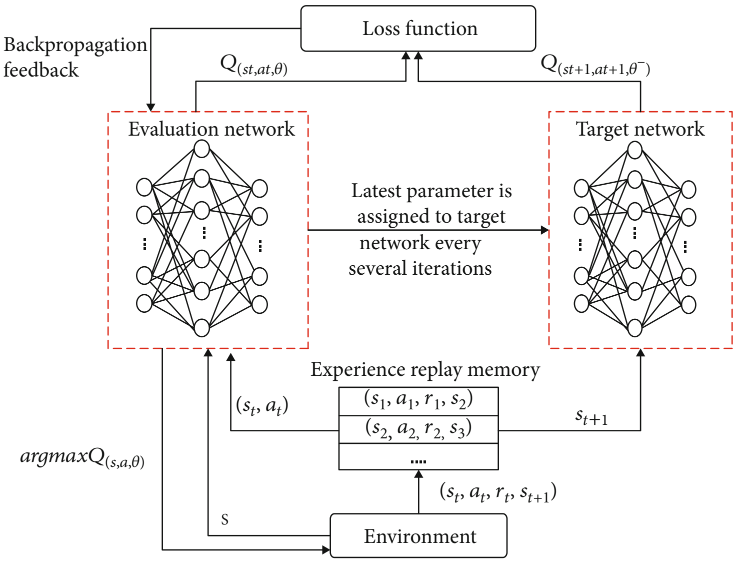

Figure 2: Flow chart of the deep Q-learning network.

r_t, and s_{t+1}) into the memory as experiences. Once the capacity is full, new received experiences will replace earlier experiences. During the training stage, the training data are randomly selected from the experience replay memory. The random selection disorganises the experience correlation, which solves the neural network's overfitting issue

The neural network enables $Q_{(s_t,a_t,\theta)} \approx Q_{(s_t,a_t)}$ [26]. According to [5], the evaluation of $Q_{(s_t,a_t)}$ is derived from Q-learning as:

$$Q_{(s_t,a_t)} = Q_{(s_t,a_t)} + \ell\left[r_t + \gamma * \max Q_{(s_{t+1},a_{t+1})} - Q_{(s_t,a_t)}\right], \tag{16}$$

where ℓ is the learning rate $\epsilon\,(0,1)$, and γ is the discount factor $\epsilon\,[0,1]$.

The neural network can be trained via the minimisation of the loss function. The loss function Loss (θ) is defined as:

$$\begin{aligned}\text{Loss}\,(\theta) &= E\left[\left(Q_{(s_{t+1},a_{t+1},\theta^-)} - Q_{(s_t,a_t,\theta)}\right)^2\right] \\ &= E\left[\left(r_t + \gamma * \max Q_{(s_{t+1},a_{t+1},\theta^-)} - Q_{(s_t,a_t,\theta)}\right)^2\right],\end{aligned} \tag{17}$$

where $r_t + \gamma * \max Q_{(s_{t+1},a_{t+1},\theta^-)}$ is the target network's Q value and $Q_{(s_t,a_t,\theta)}$ is the evaluation's Q value.

The detailed optimisation of the evaluation network and target network is shown in Figure 2. In each training step, the evaluation network receives a backpropagated loss function based on a batch of experiences randomly selected from the experience replay memory. The parameter of the evaluation network θ is then updated by the minimisation of the loss function via the stochastic gradient descent (SGD) function. After several steps, the parameter of the target network θ^- is updated by assigning the latest parameter θ to θ^-. After a training period, the two neural works are stably trained.

3.2.2. The Workflow of the Cache Content Update Policy Based on DQN. In each decision epoch, the i^{th} DQN agent receives a content request. If the content is cached locally, the serving BS delivers the requested content to the corresponding user. If the content is missed at the serving BS, the serving BS retrieves the requested content from the core network and then delivers the content to the corresponding user. Subsequently, the requested content is cached at the serving BS when the cache capacity is not full. If the cache capacity is full, the optimised evaluation network outputs the Q value of all of the actions, and the DQN agent selects an action a_k with the maximal Q value. After the execution of the action a_k, the new instant reward is calculated into the target network's Q value and a new loss function is obtained based on Eq. (17). The parameters θ and θ^- are then updated based on the minimisation of the new loss function. After a training period, the best policy $\pi^*(s)$ that can maximise the cache hit ratio and minimise the average latency is derived. The DQN-based cache content update policy is shown in Algorithm 1.

4. Results and Discussion

In this study, we consider a cache-enabled network with 4 BSs and 10 mobile users and ensure that each user is covered by a BS. For simplicity, the users are distributed along with the edge of the serving BS, and each BS has the maximal communication distance with the core network, and hence, the rate $\mathscr{d}/\mathscr{R}$ is 1. Besides, there is no overlap between any two BSs to avoid the handover between any two BSs. Furthermore, each content has the same size (2,000 bits), and the content transmission rate is 35 Mbit/s. The neural network has three layers, the input layer, hidden layer, and output layer. The hidden layer has 512 neurons, and the number of

```
The DQN-based cache content update algorithm.
Input: The feature of the state $s_t$
Initialise the parameter $\theta$ and $\theta^-$ and instant reward $\mathscr{r}_t$ =0
for step =1, Y do
  for $t$ = 1, $T$ do
    Receive a content request
    if the content request is cached locally, then
    BS directly delivers the requested content to the user end epoch
    elif
    The cache capacity is not full, then
    BS retrieves the requested content from the core network and delivers the requested content to the user
    The requested content is cached locally end epoch
    elif
    The cache capacity is full, then
    observe the current state $s_t$
    randomly generate a value $\rho$
    if $\rho < \varepsilon$, then
      randomly select an action $a_t$ from the action spaces
      else
      $a_t$ = argmax$Q_{(s_t,a_t,\theta)}$
      end if
      execute $a_t$, receive the reward $\mathscr{r}_t$, next state $s_{t+1}$
      store ($s_t$, $a_t$, $\mathscr{r}_t$, $s_{t+1}$) into the experience replay memory
      randomly selects a mini-batch of the experiences
      update the parameter of the evaluation $\theta$ via the minimisation of the backpropagated loss
      update the parameter of the evaluation $\theta^-$ in several time slots
      end if
  end for
end for
```

Algorithm 1:

neurons at the input and output layers is $(\mathscr{H}+3)$ and $(\mathscr{H}+1)$, respectively. The maximal cache capacity $\mathscr{H}$ is described in each experiment. The learning rate ℓ is 0.9, the greedy parameter ε is 0.9, and the discount factor γ is 0.1. The content requests of the i^{th} user are generated following the Zipf distribution law as

$$\mathscr{p}(\delta, \mathscr{K}, B) = \frac{\delta^{-\mathscr{K}}}{\sum_{b=1}^{B} b^{-\mathscr{K}}}, \tag{18}$$

where δ is the content rank, $\mathscr{K}$ is the Zipf parameter, and B is the total number of content requests. In each experiment, we assume that the total number of content requests is 7,200.

Figure 3 investigates the cache hit ratios of the LFU policy, LRU policy, DRL policy in [20], and our proposed policy. The Zipf parameters vary from 1.1 to 1.8, the users' locations are fixed, and the cache can store 288 types of contents at most. As the Zipf parameter $\mathscr{K}$ increases, the four policies' cache hit ratios increase. This occurs because as the Zipf parameter increases, there is less content with larger probabilities of content requests. In other words, the popular content becomes more popular, the unpopular content becomes less popular, and the type of content decreases. Considering the same cache capacity, the cached content is more popular, and therefore, the cache hit ratio increases. Our proposed policy has the highest cache hit ratio regardless of the Zipf parameter. The simulation demonstrates that the effect of the popular content in the cache hit ratio increases as the Zipf parameter increases. Thus, our proposed policy is superior to the three other policies.

Figure 4 investigates the effect of the cache capacity on the cache hit ratio. Here, the Zipf parameter is 1.4, and the mobile users' locations are fixed. The varied cache capacity is 36, 72, 108, 144, 180, 216, 252, and 288. As demonstrated, our proposed policy is superior to the three other policies since our proposed policy has the highest cache hit ratio. In addition, as the cache capacity increases, the cache hit ratios of the four policies continuously increase. When the capacity is 288, the cache hit ratios of the four policies are remarkably close. This occurs because the popular content dominates the cache hit ratio, and the cache capacity is high enough to store all of the popular contents.

The cache hit ratio under spatiotemporally varying scenarios is shown in Figure 5. In the experiment, the cache can store 216 types of contents at most, the Zipf parameters are randomly generated from 1.2 to 1.6 every 20,000 time slots, and the users are initially fixed and randomly change their locations among the four BSs after the 20,000 time slot. When the users' locations and the Zipf parameters are fixed, the gaps between our proposed policy and the three other policies are gradually stable. This occurs because all of the policies are optimally trained. After time slot 20,000, the four policies immediately decrease. This occurs because the

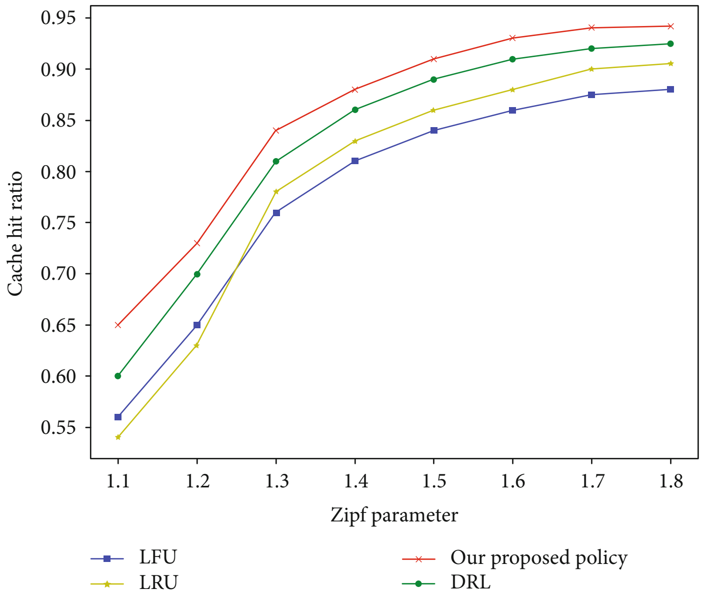

Figure 3: The cache hit ratio vs. the varying Zipf parameters. We assume that the Zipf parameters = 1.1, 1.2, 1.3, 1.4, 1.5, 1.6, 1.7, and 1.8.

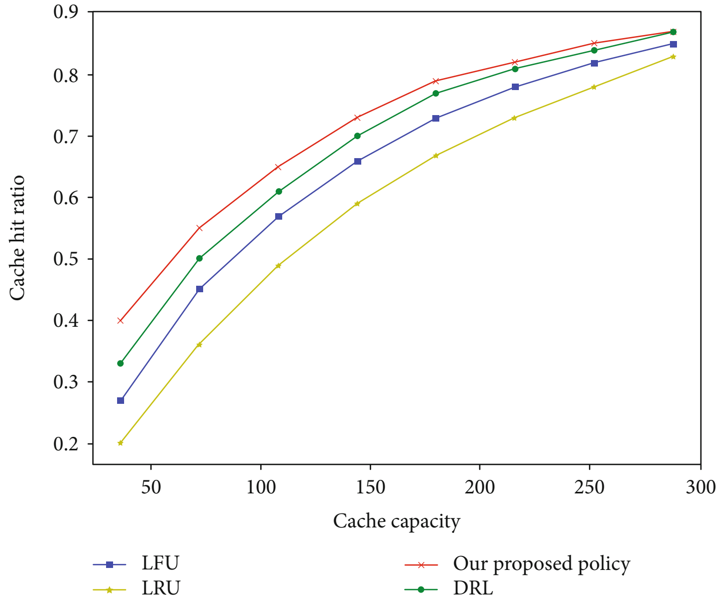

Figure 4: The cache hit ratio vs. the cache capacity. The varied cache capacity is 36, 72, 108, 144, 180, 216, 252, and 288.

content popularity changes with the random movement of the users and random generation of the Zipf parameters. Later, our proposed policy's curve slowly increases, while the three other policies' curves continuously decrease. The gaps between our proposed policy's curve and the other policies' curves continuously increase. Our proposed policy eventually improves by at least 56% compared with the three other policies. The growth ratio g is derived based on $g = C_{\text{our}} - C_{\text{existing}}/C_{\text{existing}}$, in which C_{our} and C_{existing} is the cache hit ratio of our proposed policy and any one of the other three policies, respectively. This significant improvement occurs because our proposed policy considers the effect of the users' random distribution and the random generation of Zipf parameters. Therefore, our proposed policy quickly adapts to spatiotemporally varying content requests. Consequently, we conclude that our proposed policy is superior for managing spatiotemporally varying problems.

Figure 6 demonstrates the four policies' average latencies under different Zipf parameters. Here, the Zipf parameters vary from 1.1 to 1.8, the mobile users' locations are fixed, and the cache can store 288 types of contents at most. As demonstrated, our proposed policy always has the lowest cache hit ratio compared with the other three policies. Thus, our proposed policy has the best cache hit ratio. The higher

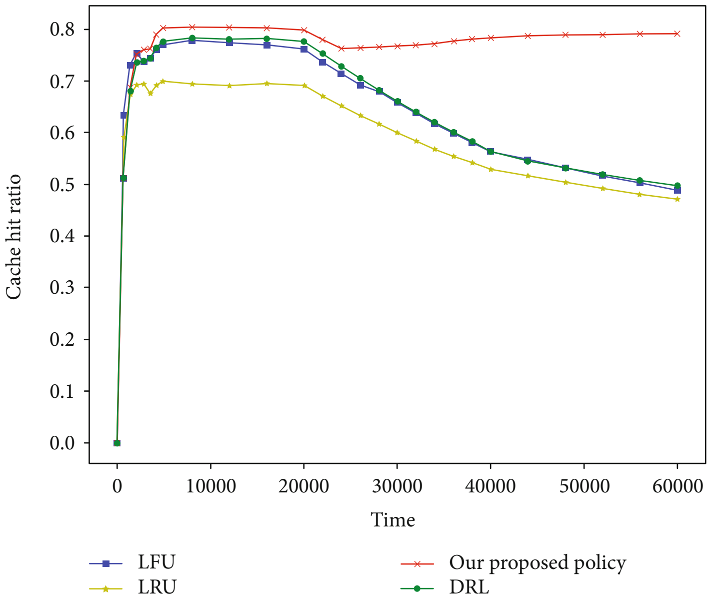

Figure 5: The cache hit ratio under spatiotemporally varying scenarios in which the mobile users randomly change their locations after the 20,000 time slot.

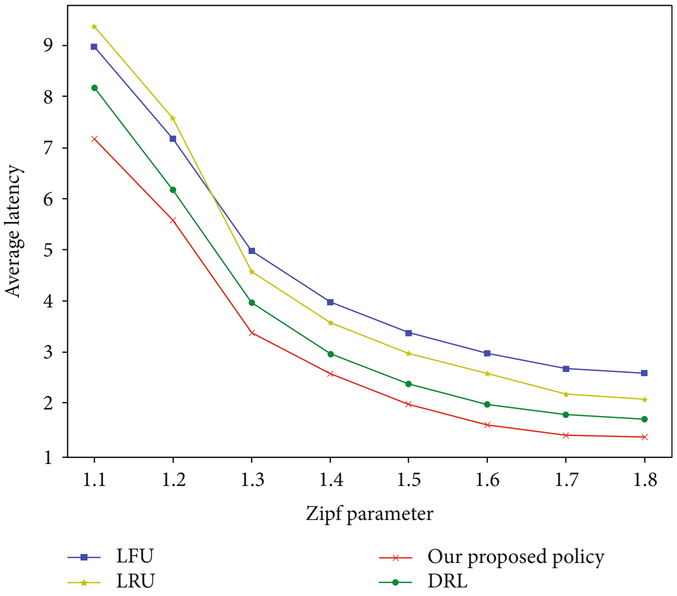

Figure 6: The average latency vs. the varying Zipf parameters. We assume that the Zipf parameters = 1.1, 1.2, 1.3, 1.4, 1.5, 1.6, 1.7, and 1.8.

the cache hit ratio is, the more contents can be retrieved locally. The local latency from the BS is much smaller than the remote latency from the core network. Therefore, our proposed policy performs better than the other three policies in terms of the average latency.

As shown in Figure 7, we investigate the effect of the cache capacity on the average latency. In this simulation, the Zipf parameter is 1.4, and the mobile users' locations are fixed. The cache capacity is 36, 72, 108, 144, 180, 216, 252, and 288. The higher the cache capacity, the lower the average latency of each policy. This occurs because more contents can be cached locally as the cache capacity increases. In addition, the slope of each policy gradually decreases. This occurs because all of the policies aim to cache the most popular contents via their limited cache capacity. As the cache capacity further increases, more contents are cached, while the recently cached contents are less popular than the initially cached contents. Consequently, the average latency increases less when caching less popular contents. Furthermore, our proposed policy has the minimal average latency regardless of the cache capacity.

Figure 8 shows the average latency under spatiotemporally varying scenarios. In the experiment, the cache is assumed that can store 216 types of contents at most, the Zipf

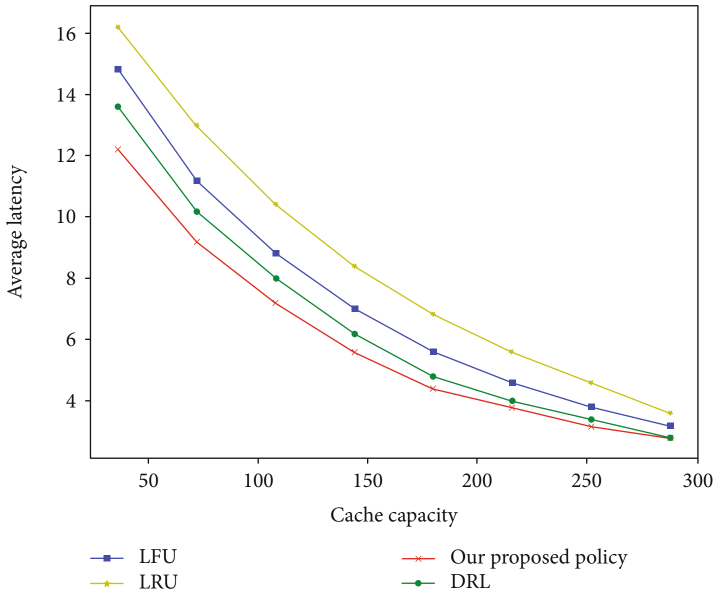

Figure 7: The average latency vs. the varying cache capacity. We assume that the cache capacity is 36, 72, 108, 144, 180, 216, 252, and 288.

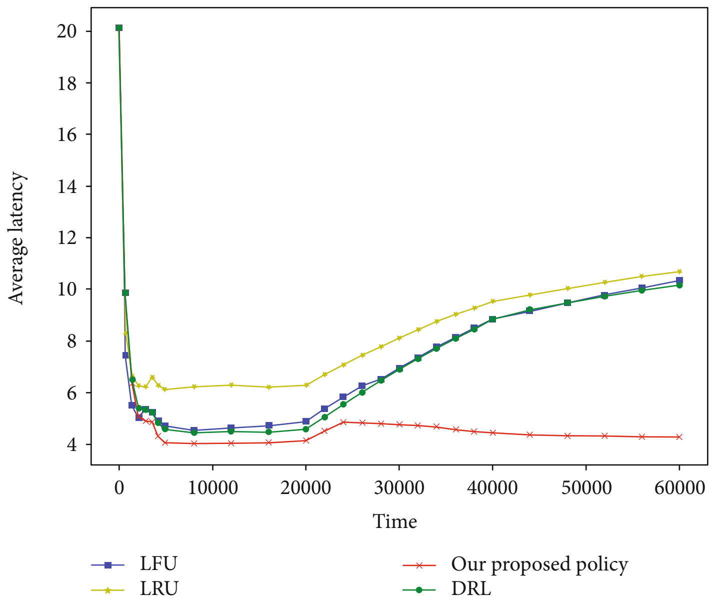

Figure 8: The average latency under spatiotemporally varying scenarios in which the mobile users randomly change their locations after the 20,000 time slot.

parameter is randomly generated from 1.2 to 1.6 every 20,000 time slots, and the users are initially fixed and randomly change their locations among the four BSs after the 20,000 time slot. In the first 20,000 time slots, each policy finally has a stable cache performance after a training period. Once the users randomly move among the four BSs, the LRU, LFU, and DRL policies' curves immediately increase, and our proposed policy's curve first slightly increases and then gradually decreases. More specifically, our proposed policy achieves a 56%-59% decrease compared to the three other policies. The reduction rate $\mathfrak{y}$ is derived based on $\mathfrak{y} = L_{\text{existing}} - L_{\text{our}}/L_{\text{existing}}$, in which L_{our} and L_{existing} is the latency of our proposed policy and any one of the other three policies, respectively. The decrease occurs because our proposed policy considers the effect of the dynamic changes in the user distribution and Zipf parameters on the latency, while the other three policies do not. The simulation demonstrates that our proposed policy can perform stably under spatiotemporally varying scenarios.

5. Conclusions

In this study, a DRL-based cache content update policy is proposed with the objective to maximise the cache hit ratio and minimise the average latency. Compared to the existing policies, a more practical cache scenario is considered, in which the content requests vary spatiotemporally. The dynamic content update problem is formulated as an MDP problem, and DQN is applied to solve this MDP problem. Specifically, the neural network is trained to approximate the Q value, in which the training data are chosen from the experience replay memory. The DQN agent derives the optimal policy from the neural network for the cache decision. Compared with the existing policies, e.g., the LFU, LRU, and DRL [20] policies, the simulation results show that our proposed DRL-based cache content update policy has the best cache performance in the considered spatiotemporally varying scenario and is 56%–64% improved in terms of the cache hit ratio and 56%–59% decreased in terms of the average latency.

Data Availability

Content requests were described in the simulation section.

Conflicts of Interest

Lincan Li, Chiew Foong Kwong, Qianyu Liu, and Jing Wang declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This study was supported by Ningbo Natural Science Programme (NBNSP), project code 2018A610095.

References

[1] M. Chen, Y. Hao, L. Hu, K. Huang, and V. K. N. Lau, "Green and mobility-aware caching in 5G networks," *IEEE Transactions on Wireless Communications*, vol. 16, no. 12, pp. 8347–8361, 2017.

[2] D. Deng, J. Xia, L. Fan, and X. Li, "Link selection in buffer-aided cooperative networks for green IoT," *IEEE Access*, vol. 8, pp. 30763–30771, 2020.

[3] Cisco, *Cisco visual networking index: global mobile data traffic forecast update, 2017–2022*, White Paper, 2019.

[4] C. Zhong, M. C. Gursoy, and S. Velipasalar, "A deep reinforcement learning-based framework for content caching," in *2018 52nd Annual Conference on Information Sciences and Systems (CISS)*, pp. 1–6, Princeton, NJ, USA, March 2018.

[5] Z. Zhao, W. Zhou, D. Deng, J. Xia, and L. Fan, "Intelligent mobile edge computing with pricing in Internet of Things," *IEEE Access*, vol. 8, pp. 37727–37735, 2020.

[6] H. Zhu, Y. Cao, X. Wei, W. Wang, T. Jiang, and S. Jin, "Caching transient data for Internet of Things: a deep reinforcement learning approach," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 2074–2083, 2019.

[7] Y. Wu, S. Yao, Y. Yang, Z. Hu, and C.-X. Wang, "Semigradient-based cooperative caching algorithm for mobile social networks," in *2016 IEEE Global Communications Conference (GLOBECOM)*, pp. 1–6, Washington, DC, USA, December 2016.

[8] P. Wu, J. Li, L. Shi, M. Ding, K. Cai, and F. Yang, "Dynamic content update for wireless edge caching via deep reinforcement learning," *IEEE Communications Letters*, vol. 23, no. 10, pp. 1773–1777, 2019.

[9] X. Zhang and Y. Cao, "A cooperation-driven ICN-based caching scheme for mobile content chunk delivery at RAN," in *2017 13th International Wireless Communications and Mobile Computing Conference (IWCMC)*, pp. 1437–1442, Valencia, Spain, June 2017.

[10] S. Zhang, N. Zhang, P. Yang, and X. Shen, "Cost-effective cache deployment in mobile heterogeneous networks," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 12, pp. 11264–11276, 2017.

[11] R. Wang, F. Hajiaghajani, and S. Biswas, "Distributed caching in mobile networks with heterogeneous content demand," in *2017 14th IEEE Annual Consumer Communications & Networking Conference (CCNC)*, pp. 172–178, Las Vegas, NV, USA, Jan 2017.

[12] Z. Luo, M. LiWang, Z. Lin, L. Huang, X. Du, and M. Guizani, "Energy-efficient caching for mobile edge computing in 5G networks," *Applied Sciences*, vol. 7, no. 6, p. 557, 2017.

[13] D. Guo, L. Tang, X. Zhang, and Y.-C. Liang, "Joint optimization of handover control and power allocation based on multi-agent deep reinforcement learning," *IEEE Transactions on Vehicular Technology*, pp. 1–1, 2020.

[14] D. Liu and C. Yang, "A deep reinforcement learning approach to proactive content pushing and recommendation for mobile users," *IEEE Access*, vol. 7, pp. 83120–83136, 2019.

[15] W. Li, J. Wang, G. Zhang, L. Li, Z. Dang, and S. Li, "A reinforcement learning based smart cache strategy for cache-aided ultra-dense network," *IEEE Access*, vol. 7, pp. 39390–39401, 2019.

[16] Y. Wei, Z. Zhang, F. R. Yu, and Z. Han, "Joint user scheduling and content caching strategy for mobile edge networks using deep reinforcement learning," in *2018 IEEE International Conference on Communications Workshops (ICC Workshops)*, pp. 1–6, Kansas City, MO, USA, May 2018.

[17] Y. Sun, M. Peng, and S. Mao, "Deep reinforcement learning-based mode selection and resource management for green fog radio access networks," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 1960–1971, 2019.

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

[19] R. Zhao, X. Wang, J. Xia, and L. Fan, "Deep reinforcement learning based mobile edge computing for intelligent Internet of Things," *Physical Communication*, vol. 43, article 101184, 2020.

[20] F. Jiang, Z. Yuan, C. Sun, and J. Wang, "Deep Q-learning-based content caching with update strategy for fog radio access networks," *IEEE Access*, vol. 7, pp. 97505–97514, 2019.

[21] C. Wang, W. Li, D. Li, M. Song, C. Dong, and X. Wang, "Edge caching via content offloading in heterogeneous mobile opportunistic networks," in *2018 IEEE 24th International Conference on Parallel and Distributed Systems (ICPADS)*, pp. 787–794, Singapore, Singapore, December 2018.

[22] N. Wang, G. Shen, S. K. Bose, and W. Shao, "Zone-based cooperative content caching and delivery for radio access network with mobile edge computing," *IEEE Access*, vol. 7, pp. 4031–4044, 2019.

[23] X. He, K. Wang, H. Huang, T. Miyazaki, Y. Wang, and S. Guo, "Green resource allocation based on deep reinforcement

learning in content-centric IoT," *IEEE Transactions on Emerging Topics in Computing*, vol. 8, no. 3, pp. 781–796, 2020.

[24] N. C. Luong, D. T. Hoang, S. Gong et al., "Applications of deep reinforcement learning in communications and networking: a survey," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 4, pp. 3133–3174, 2019.

[25] Y. He, N. Zhao, and H. Yin, "Integrated networking, caching, and computing for connected vehicles: a deep reinforcement learning approach," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 1, pp. 44–55, 2018.

[26] F. Xu, F. Yang, S. Bao, and C. Zhao, "DQN inspired joint computing and caching resource allocation approach for software defined information-centric Internet of Things network," *IEEE Access*, vol. 7, pp. 61987–61996, 2019.

Hindawi
Wireless Communications and Mobile Computing
Volume 2020, Article ID 8858298, 11 pages
https://doi.org/10.1155/2020/8858298

Research Article

Research on Multinode Collaborative Computing Offloading Algorithm Based on Minimization of Energy Consumption

Dongsheng Han, Yu Liu, and Junhong Ni

Department of Electronic and Communication Engineering, North China Electric Power University, Baoding, 071003 Hebei, China

Correspondence should be addressed to Yu Liu; 2644270950@qq.com

Received 14 July 2020; Revised 16 October 2020; Accepted 21 October 2020; Published 6 November 2020

Academic Editor: Lisheng Fan

Mobile edge computing (MEC) nodes are deployed at positions close to users to address excessive latency and converging flows. Nevertheless, the distributed deployment of MEC nodes and offload of computational tasks among several nodes consume additional energy. Accordingly, how to reduce the energy consumption of edge computing networks while satisfying latency and quality of service (QoS) demands has become an important challenge that hinders the application of MEC. This paper built a local-edge-cloud edge computing network and proposes a multinode collaborative computing offloading algorithm. It can be applied to smart homes, realize the development of green channels, and support local users of Internet of Things (IoT) to decompose computational tasks and offload them to multiple MEC or cloud nodes. The simulation analysis reveals that the new local-edge-cloud edge computing offload method not only reduces network energy consumption more effectively compared with traditional computing offload methods but also ensures the implementation of more data samples.

1. Introduction

With the continuous development of the Internet of Things (IoT) technology in recent years, IoT network equipment has developed perception and communication abilities, and the user end of the network can extend to information exchange and communication between any goods in daily life [1]. IoT technology has also been used in various aspects of industrial production and daily life. In transportation and network performance optimization [2], IoT has been used in smart homes, smart industries, and smart cities, among others. Previous studies have largely focused on the application scene of smart homes. The local user ends of IoT in smart homes can take the form of any good. Therefore, IoT contains diversified user data, whereas intelligent electrical apparatus requires a rapid and effective processing of task data [3]. In this case, a fast, efficient, and safe task processing mode needs to be devised to meet the demands of users with a large data size or high sensitivity to latency. Given that the traditional single-cloud model cannot meet such demands, the concept of mobile edge computing (MEC) has been proposed based on cloud computing [4]. MEC is a new computing model, and MEC nodes are widely distributed in the vicinity of the client to provide intelligent services for local users. Edge nodes can be installed on the edge server (e.g., vehicles and UAV) to meet the linkage demands of different users [5]. Combined with MEC, a multinode cooperation of data tasks is realized by transmitting data between the local users of IoT and MEC nodes wherein the local user data of IoT are offloaded to nearby MEC servers, thereby addressing the limited computing capability of these users and reducing their computing task pressures. However, MEC nodes have a limited computing capacity, thereby requiring a cooperation among multiple MEC nodes to handle computing tasks with a large data size.

To solve the network energy consumption problem under a large data size at the user ends of IoT, this study initially analyzes and selects MEC nodes in a local-edge-cloud edge computing network model while considering the distances between the MEC nodes and user ends, the channel characteristics, and the CPU energy consumption.

The main contributions can be summarized as follows:

(1) The local-edge-cloud edge computing network model proposed in this paper supports the local user ends of IoT in their parallel offloading of a computing

task to multiple MEC nodes or a cloud. This study takes both network computation and transmission into account, considered from the three layers of local, edge, and cloud

(2) Latency cannot be directly accumulated due to the parallel data transmission. Instead, the time for receiving and processing data at different nodes is analyzed to determine the network latency. An integral linear programming problem that targets the optimization of network energy consumption is formulated, and single-user task offloading is analyzed by using the branch-and-bound (BB) algorithm to minimize the overall network energy consumption

(3) The simulation results show that the demands for MEC nodes increase along with the size of offload data at the local user ends of IoT. Moreover, the multinode collaborative model is significantly superior over the traditional computing offloading algorithm in terms of energy consumption and latency, especially under large offload data sizes

The rest of this study is organized as follows. In Section 2, related work is introduced. Section 3 introduces the proposed model. Section 4 discusses in detail the construction of an objective function for the multinode computing offload model and the BB algorithm used in the optimization. Section 5 analyzes the simulation results. Section 6 concludes the paper.

2. Related Work

The local user ends of IoT can offload computing tasks to MEC nodes via global and partial offloading. In global offloading, the entire computing task is offloaded to an MEC node. Liu et al. [6] used the 1D searching algorithm to reduce implementation latency to the maximum extent and gave comprehensive considerations to the queuing state in the application buffer zone and the available processing capacity. However, edge nodes have inadequate computing capacities and experience long transmission latency. To address this problem, this paper proposes a partial offload method that implements parts of the computing task at the local position and offloads the other parts to the MEC for implementation. Further details on partial offloading can be found in [7]. In a partial offload program, the distribution positions of data tasks need to be determined; tasks are successively transferred to each node to execute tasks after the user is partitioned. In [8], Yang et al. proposed the concept of task zoning, which determines offload modules and implementation methods, that is, whether the tasks are implemented at local positions or offloaded to MEC and cloud nodes. Meanwhile, Zhao et al. [9] transformed the partial offloading problem into a nonlinear constraint problem and adopted a linear programming approach to solve this problem and realize the goal of optimal processing. Given their diversity, network data of different sizes are generated. Accordingly, resource limits have become key problems in the offload process that have been discussed in [9–11]. For instance, Zhao [10] analyzed resource limits from the perspectives of network capacity and data allocation, chose an appropriate position for data processing, and guaranteed the smooth implementation of additional data tasks. In [11], a data task was segmented by employing a partial offload method, and this task was transmitted successively to MEC and cloud nodes for implementation, thereby overcoming resource limits. To address the limitations in node quantity and processing ability, You and Huang [12] proposed an optimal resource allocation strategy for a time division multiple access system to process the queuing of tasks and ensure resource processing efficiency. Aiming at the complex resource allocation problem, Ref. [13] proposed an intelligent resource allocation framework to solve the complex resource allocation problem of collaborative mobile edge computing network. The resource allocation scheme was determined according to the edge computing server's computing capacity, channel quality, resource utilization, and latency constraints.

When users have a large number of computing tasks, a single MEC node cannot meet the demand of processing offload tasks from the user end even if the partial offload method is applied. As a result, several nodes must be selected in the collaborative processing of offload tasks. Fan et al. [14] adopted a multinode collaboration method that allows nearby MEC nodes to share the computing pressure of the target node when the computing task at the user ends is too large for a single MEC node. They also designed an algorithm for solving the optimization problem by using an interior point method and a logarithmic potential barrier function to optimize the energy consumption problem of the multinode collaboration system. This multinode collaboration method is mainly used to address the inadequate computing capacity of single nodes. Based on a dynamic and self-configuring multiequipment mobile cloud system, Habak et al. [15] implemented relevant computing tasks and expanded the range of the cloud system by using the surrounding vacant mobile equipment as MEC servers with an aim to solve the problem where the network load exceeds the computing capacity of nodes. In a multinode collaboration method, the computing task should be allocated to multiple nodes, but this action involves the allocation and deployment of nodes. Reference [16] considers link selection in collaborative networks. Based on the characteristics of two branches in the system, the buffer-assisted relay combination technology is used to provide accurate expression of interrupt probability for the common channel interference network to evaluate the transmission performance of the network. In [17], the authors selected the deployment positions of MEC nodes, such as LTE micro sites and gathering stations of multiwireless access technology communities. With the continuous popularization of MEC technology, multinode collaborative technology has been increasingly used in practice.

In the above studies, users offload the computing tasks completely or partially to one or several MEC nodes, optimize the network structure, increase the task processing capability of the network, and explore resource optimization in a multinode collaborative network structure. Nevertheless, MEC nodes are extensively distributed in ranges of local user nodes, and several MEC nodes in a wireless network are selected to participate in computing. Involving more nodes in a network will increase its overall energy consumption. After the introduction of the green MEC philosophy, network energy consumption has become a key concern among researchers.

In the multinode task allocation model, MEC nodes that implement the computing tasks are chosen reasonably to reduce network energy consumption. Zhang et al. [18] applied a single-user mobile edge computing offload (MECO) approach to the MEC network model, where network energy consumption is treated as the optimization goal, and the appropriate offload strategy is determined by comprehensively changing the number of CPU periods and network transmission rate. However, this study only considers the single-user MECO model. Meanwhile, the authors in [19] fully considered energy consumption and latency of end users in the multiuser MECO distributed computing offload model and realized an optimal allocation of resources in the computing offload process by using game theory. Reference [20] constructs an intelligent edge computing network based on pricing. When the user is offloading data, latency and price are taken as performance indicators, stochastic game method is used to determine the user signal processing scheme, and offloading strategy is designed to reduce latency and price. In [21], to cope with energy shortage in a heterogeneous network, a shared link was established among multiple base stations (BS) and was extended to the macro and micro domains for analysis. At the same time, in the heterogeneous network, due to the complex distribution of base stations and users, multilayer switching and power distribution need to be considered. In Ref. [22], there is the switching and power distribution problem in the two-layer heterogeneous network composed of macro station and millimeter wave. A multiagent augmented learning algorithm based on the proximal policy optimization is developed to realize the interaction between multiuser devices. Ng et al. [23] proposed an offload priority function by considering quantitative equality, transmission channel, and local computing situations. By analyzing this offload priority function, the optimal network resource allocation was realized, and the overall network energy consumption was used as the measurement index.

In sum, many studies have examined multinode collaboration and data offloading. Users transmit data to multiple nodes in a step-by-step manner before their implementation. When the data size at the user ends is relatively large, then the step-by-step transmission leads to significant latency, thereby destroying the latency constraints of users and consuming a considerable amount of network energy. On this basis, the superiority of the model created in this paper is more prominent.

3. System Model

Figure 1 illustrates a local-edge-cloud edge computing network that has K local user ends of IoT served by N wireless eNodeBs. Each eNodeB is equipped with one MEC server or N MEC nodes. The computing task from the local user ends of IoT can be implemented in site, partially offloaded to the MEC nodes, or partially transmitted to the cloud server through the routers at eNodeB. Before offloading tasks, the local user ends of the IoT segment these tasks while following certain rules, and the segments choose the appropriate MEC nodes or cloud servers for task offloading based on the latency, energy consumption, computing capacity of MEC nodes, and other parameters. In Ref. [6], the sequential transmission of segmented task blocks will cause a certain latency waste. Based on the above, this paper makes improvements by transferring the segmented task block to the appropriate node to perform tasks synchronously, determining the optimal assignment location of the task at the user end, transmitting and processing the task at the same time, and processing more data under the same latency constraint. Without loss of generality, this study hypothesizes that the computing task of local user UE_1 at a moment can be segmented into N task blocks. Task block 1 can be implemented at the local user ends of IoT. Offloading to and implementing at nodes MEC_1, MEC_2, and MEC_3 are optional for task block 2, task blocks 3, 4, and 5, and task blocks 6 and 7, respectively. Given that the computing capacity of MEC nodes cannot meet the demands of residual task blocks, these blocks are transmitted to the cloud for implementation. A parallel offloading of multiple task blocks is applied to reduce the network latency and overall network energy consumption.

3.1. Network Energy Consumption. In studying the local-edge-cloud edge computing network model, the computing and transmission capacity of the network should be considered to minimize the network energy consumption because the data from the local user ends of IoT are offloaded simultaneously and implemented at multiple nodes. Therefore, "network energy consumption" in this paper includes the energy consumed for the parallel transmission of computing tasks from the local user ends to the MEC and cloud nodes and the energy consumed for transmitting a computing task from the local user ends to different nodes. The computing model of the local-edge-cloud edge computing network is defined as A_k (R_k, s_k), where R_k is the task value of user k $(k = \{1, 2, \cdots, K\})$ and s_k is the time spent by user k in executing the task. The computing energy consumption of user k can be expressed as

$$E_{\text{com}}^{k} = R_k C_k m_k, \tag{1}$$

where C_k is number of CPU turns needed to execute a computing task per bit of data and m_k is the energy consumed for each CPU turn.

When the computing task cannot be executed completely at the local user ends of IoT, this task must be offloaded to the appropriate nodes, which will consume a certain amount of transmission energy. Transmission energy consumption is related to both the transmission time and transmission power of the task. The transmission energy can be formulated as

$$E_{\text{trans}}^{k} = t_k p_k, \tag{2}$$

where t_k is the transmission time of the computing task of

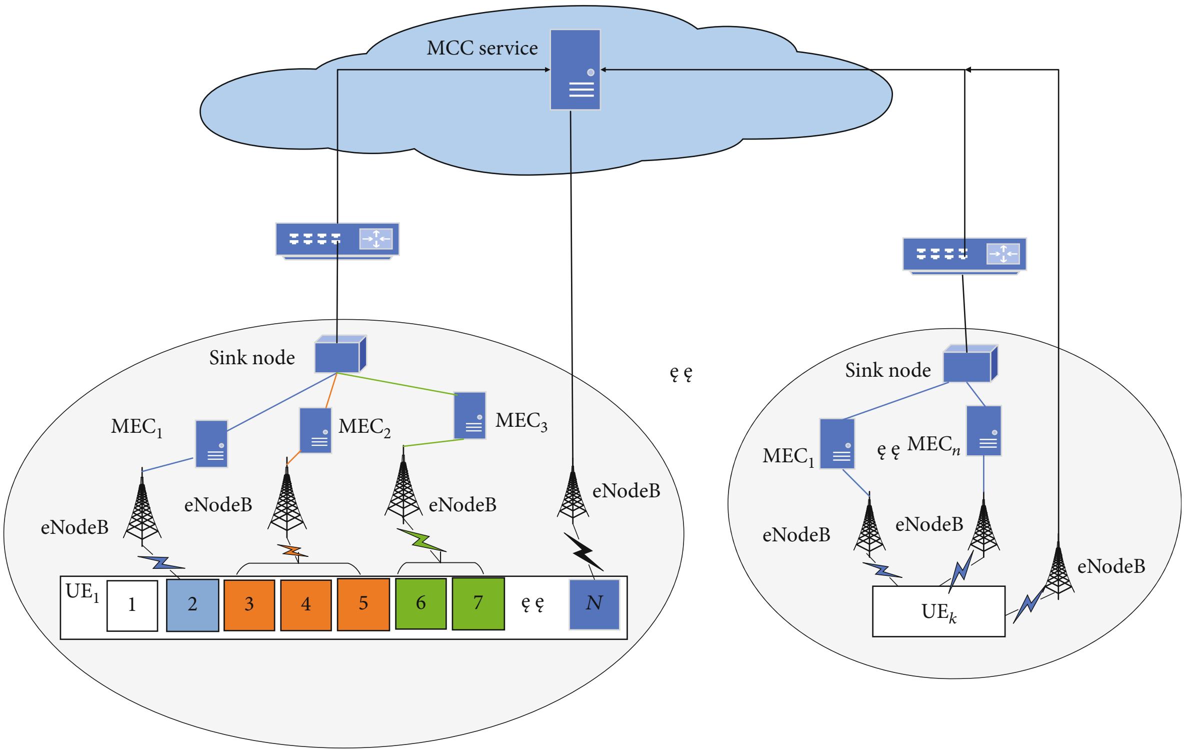

Figure 1: Local-edge-cloud edge computing network model.

user k and p_k is the transmission power between user k and the offload nodes. The overall energy consumed by user k to execute a task is computed as the sum of transmission energy consumption and computing energy consumption:

$$E_{\text{total}}^{k}=E_{\text{com}}^{k}+E_{\text{trans}}^{k}. \tag{3}$$

3.2. Computing Capacity. The number of CPU turns needed for user k to implement 1 bit of task at local users, MEC nodes, and cloud nodes is denoted by C_k^L, C_k^E, C_k^C, respectively. Meanwhile, the energy consumed for each CPU turn in implementing the computing task of user k at local users, MEC nodes, and cloud nodes is denoted by m_k^L, m_k^E, m_k^C. Under the multinode collaboration mode, the data are segmented at the local user ends of IoT, and the segmented data are transmitted to the MEC or MCC nodes for computing. To easily observe the offload condition of segmented tasks, one data unit ϕ (kbit) is set, and the data at the local user ends of IoT are expressed as data units. The data of user k are divided into M_k data units as $R_k=M_k\phi$. For all nodes, parameter ρ is set, where $\rho_{k\to 0}$ denotes the number of data units in the local computing of user k. The network has n MEC nodes, where $n=\{1,2,\cdots,N\}$. $\rho_{k\to n}$ and $\rho_{k\to N+1}$ refer to the number of data units that local user k offloads to MEC$_n$ and the cloud nodes for task execution, respectively. With respect to the selection problem between the local user ends of IoT and MEC nodes, parameter $\beta_{k,m,n}$ indicates that the computing task block m of local user k is offloaded and implemented at node n. In this model, the local user ends of IoT segment the computing task into several blocks and offload them to multiple MEC and cloud nodes. A data unit can only be offloaded to a single node $(\sum_{n=1}^{N}\beta_{k,m,n}=1)$, while one MEC node can receive several data units $(\sum_{m=1}^{M}\beta_{k,m,n}=\rho_{k\to n})$. When $n=0$, the computing task is implemented at the local user ends of IoT, but when $n=N+1$, the computing task is implemented at cloud nodes.

Given that the data are segmented at the local user ends of IoT and transmitted to several nodes simultaneously, the data allocated to different nodes should meet the computing capacities of different nodes. The data of user k are analyzed as

$$C_k^n R_k^n \le F. \tag{4}$$

Let F_0, F_n, F_{N+1} be the computing capacities of the local user ends, MEC nodes, and cloud nodes, that is, the number of CPU turns needed to implement the computing task. In equation (4), $R_k^n=\rho_{k\to n}\phi$, where $n=0$ denotes the size of the task implemented at the local user ends of IoT, $n=\{1,2,\cdots,N\}$ refers to the size of the task implemented at the MEC nodes, and $n=N+1$ refers to the size of the task implemented at the cloud nodes.

3.3. Computing Latency. Computing latency is determined by computing the number of nodes, number of CPU turns, and node computing capacity. When the computing task is executed at the local user ends of IoT, the data computing latency of user k can be expressed as

$$s_k^L=\frac{C_k^L R_k^0}{F_0}. \tag{5}$$

Given that the data are segmented at the local user ends of IoT and are transmitted to several MEC nodes simultaneously for implementation, the computing latency is computed as the maximum computing latency of different nodes. The computing latency of one node can be formulated as

$$s_{k,n}^{E} = \frac{R_k^n C_{k,n}^E}{F_n}. \qquad (6)$$

When the computing task cannot be implemented at the local user ends of IoT and MEC nodes, this task should be transmitted to cloud servers. The computing latency at the cloud nodes can be formulated as

$$s_k^C = \frac{R_k^{N+1} C_k^C}{F_{N+1}}. \qquad (7)$$

3.4. Transmission. The transmission links in a network refer to the wireless communication links between the MEC server and UE, the transmission VLAN among MEC servers, and the transmission links between the MEC and cloud servers. In the network transmission process, the relationship between network computing capacity and transmission capacity should be considered. If the computing capacity is too high, then the channel resources in the network cannot be allocated to the local user ends of IoT, thereby congesting the channels and increasing network latency. Let R_k (bit) be the data size that local user k of IoT needs to process. Specifically, R_k^0 refers to the size of the computing task implemented at the local user ends of IoT, R_k^n is the size of the computing task implemented at the MEC nodes, and R_k^{N+1} is the size of the computing task implemented at the cloud nodes. When the computing task can be implemented at the local user ends of IoT and does not need to be transmitted, no transmission energy is consumed. Transmission energy is only consumed when the computing task is offloaded to the MEC and cloud nodes.

Let t_k denote the transmission time for one data unit ϕ (kbit), where $t_k > 0$. Therefore,

$$t_k = \frac{R_k}{r_k}, \qquad (8)$$

where r_k refers to the data transmission rate from user k to the chosen nodes. The total transmission time in the computing offload process is calculated by the number of bit units that the user offloads to nodes $\rho_{k\to n}$. Suppose that n MEC servers receive data from the user end. These data are segmented at the local user ends of IoT, and data transmission is performed simultaneously. However, $\rho_{k\to n}$ computing tasks will experience $\rho_{k\to n}t_k$ transmission time in the task transmission process of each part. Given that each node has unique basic parameters, the size of the offloaded data also varies. The transmission time from the local user ends of IoT to the nodes shall be taken as the transmission time from the local user ends to the node with the largest offloaded task. This node should meet

$$\sum_{n=1}^{N+1} \rho_{k\to n} t_k \le T, \qquad (9)$$

where T represents the latency in meeting the QoS demands of users.

3.5. Transmission Power. In the transmission from local user k to the chosen nodes, the transmission rate can be expressed as

$$r_k = W \log_2\left(1 + \frac{p_{k,n}\mathrm{PL}_{k,n}}{\sigma^2}\right), \qquad (10)$$

where W is the channel bandwidth, $p_{k,n}$ is the transmission power between local user k and node n, and h_k is the channel characteristics between local user k and node n. The differences in the channel characteristics can be ascribed to the variances in the distances of each node from the local user k. The value of PL_k meets the large-scaled attenuation characteristic and is related to transmission distance. PL_k is expressed as $\mathrm{PL} = \mathrm{PL}_{\mathrm{FS}}(d_0) + 10n \lg (d/d_0) + X_\sigma$, where d is the transmission distance, d_0 is the reference distance, n is the route loss index, and X_σ is a Gaussian random variable with a 0 mean and σ^2 standard deviation. Meanwhile, W_E, W_C represent the bandwidths between the local users of IoT and edge nodes and those between the users and cloud nodes. When $n = \{1, 2, \cdots, N\}$, $p_{k,n}$ and $\mathrm{PL}_{k,n}$ represent the transmission power and loss between user k and MEC_n. When $n = N + 1$, these parameters represent the transmission power and loss between local user k and the cloud nodes.

According to equations (1) and (3), data transmission rate (r_k) can be expressed in two ways. The transmission power from the local user k to node n can be expressed as

$$P_{k,n} = \frac{\left(2^{r_{k,n}/W} - 1\right)\sigma^2}{\mathrm{PL}_{k,n}}. \qquad (11)$$

In sum, to analyze the transmission in the local-edge-cloud edge computing network and computing situations, network energy consumption can be computed as total energy consumption = computing energy consumption + transmission energy consumption. Computing energy consumption includes the computing energy consumed by the local user ends of IoT and by the collaboration between MEC nodes and cloud servers. Meanwhile, transmission energy consumption includes the wireless transmission energy consumption between the local user ends of IoT and MEC nodes and that between the local user ends and cloud servers. Network latency, which includes computing latency and transmission latency, is considered in computing network energy consumption given that the network energy consumption should be minimized under the premise of meeting network latency requirements.

4. Multinode Collaborative Computing Offloading Algorithm

In the local-edge-cloud edge computing network model, one part of the computing task is implemented at the local user ends of IoT, whereas the other parts are offloaded to the appropriate nodes. The data of the user are segmented following certain rules and are offloaded simultaneously to several nodes. Given that MEC nodes are close to the local user ends of IoT, a short data transmission time is achieved. However, the offload positions should be chosen reasonably based on the user demand given the limited computing capacity of MEC nodes.

4.1. Establishment of an Objective Function. According to equation (3), the overall network energy consumption includes computing and transmission energy consumption. In the local-edge-cloud edge computing network model, certain tasks are distributed to all levels. In other words, network energy consumption includes the computing and transmission energy consumption of the local user ends of IoT, MEC nodes, and cloud nodes.

Implementing the computing task at the local user ends of IoT only consumes computing energy. The energy consumed can be formulated as

$$E_L = \sum_{k=1}^{K} \rho_{k\to 0}\phi C_k^L m_k^L. \tag{12}$$

When the computing task is offloaded to edge nodes, several MEC nodes surround the local user ends of IoT. Therefore, the appropriate MEC nodes should be selected. Let the selection parameter be $\rho_{k\to n}$, $n = \{1, 2, \cdots, N\}$, which reflects the selection of MEC nodes. The overall energy consumption of the MEC node includes both computing and transmission energy consumption and can be expressed as

$$E_E = \sum_{k=1}^{K}\sum_{n=1}^{N}\left(\rho_{k\to n}\phi C_{k,n}^E m_{k,n}^E + \rho_{k\to n} t_{k,n} \frac{\sigma^2\left(2^{1/2^{t_{k,n}W_E}} - 1\right)}{\mathrm{PL}_{k,n}^E}\right). \tag{13}$$

When the computing task is partially offloaded to the cloud servers, the overall energy consumption of cloud nodes can be expressed as

$$E_C = \sum_{k=1}^{K} \rho_{k\to N+1}\phi C_k^C m_k^C + \rho_{k\to N+1} t_{k,N+1} \frac{\sigma^2}{PL_{k,N+1}^C}\left(2^{1/t_{k,N+1}W_C} - 1\right). \tag{14}$$

The overall network energy consumption is then computed as the total energy consumed by the local user ends of IoT, MEC nodes, and cloud nodes:

$$E_{\text{total}} = E_L + E_E + E_C. \tag{15}$$

Given that the network model considers the computing and transmission of data from the local user ends of IoT, network latency includes both computing and transmission latencies. The computing latency of the local user ends of IoT, MEC nodes, and cloud nodes should be considered when applying a local-edge-cloud edge computing network model. The computing latency of local user k can be expressed as

$$s_k^L = \frac{C_k^L \rho_{k\to 0}\phi}{F_0}. \tag{16}$$

Unlike in the mutual transmission computing offload model, the computing task is segmented at the local user ends of IoT and are transmitted simultaneously to multiple nodes for processing. Therefore, the computing latency is taken as the maximum computing latency of MEC and cloud nodes:

$$s_k' = \max\left\{\frac{\rho_{k\to n}C_k^n\phi}{F_n},\ n = 0, 1, \cdots, N, N+1\right\}. \tag{17}$$

The overall computing latency of the network is then formulated as

$$s_k = s_k^L + s_k'. \tag{18}$$

Meanwhile, transmission latency mainly involves the wireless transmission links from the local user ends of IoT to the MEC nodes and the VLAN transmission network from the local user ends of IoT to the cloud nodes. Given that the data are segmented at the local user ends of IoT, the appropriate nodes should be selected for the simultaneous transmission of segmented data. When parallel data transmission is applied, the overall transmission latency of the network can be expressed as

$$t_k = \max\left\{\rho_{k\to n}t_{k,n},\ n = 0, 1, \cdots, N, N+1\right\}. \tag{19}$$

The network latency is then computed as the sum of computing latency and transmission latency:

$$D_k = s_k + t_k. \tag{20}$$

The goal of this multinode collaborative computing offload model is to minimize the overall network energy consumption while meeting the time constraints. The optimization system of the multinode collaborative computing offload network is

$$\min \quad E_{\text{total}} \tag{21}$$

$$\text{s.t.} \quad D_k \leq T, \tag{22}$$

$$t_{k,n} > 0, \tag{23}$$

$$\sum_{n=1}^{N} \beta_{k,m,n} = 1, \tag{24}$$

$$\sum_{m=1}^{M} \beta_{k,m,n} = \rho_{k\to n}, \tag{25}$$

$$\rho_{k\to n}\phi C_{k,n}^{E} \le F_n, \quad n = \{1, 2, \cdots, N\}, \tag{26}$$

$$C_k^L \rho_{k\to 0}\phi \le F_0, \tag{27}$$

$$C_k^C \rho_{k\to N}\phi \le F_{N+1}, \tag{28}$$

where (22) and (23) are the limiting conditions of transmission time (with (22) indicating that the network latency is smaller than the latency limit of user ends), (24) and (25) denote the data allocation after the segmentation at the local user ends (with (24) indicating that only one data unit can be offloaded to one MEC node and (25) indicating the number of unit tasks that can be processed by a single MEC node), and (26) to (28) denote the computing capacity limitations of MEC nodes, local user ends of IoT, MEC nodes, and cloud nodes.

To address the above conditions, the size of the computing task blocks offloaded to different nodes in equation (21) is denoted by $\rho_{k\to n}$. The task allocation of nodes under optimal energy consumption is evaluated by analyzing the value of $\rho_{k\to n}$. Given that $\rho_{k\to n}$ determines the number of data units, its value can only be expressed as an integer. Therefore, the optimization problem becomes an integer programming problem.

4.2. Optimization Based on the BB Algorithm. The resource allocation scheme for MEC nodes is determined by using the BB algorithm, which searches all feasible solution spaces for the optimization problem with constraints. During the implementation of this algorithm, all feasible solution spaces are continuously divided into smaller subsets, and a lower or upper bound is calculated as a solution for each subset. With respect to the integer programming problem, the BB algorithm solves the ordinary linear programming problem through simplex and divides the nonintegral decision variables into two proximate integers. The conditions are then listed and added into the original problem. Meanwhile, the constraint vector after updating is solved, from which the upper or lower bound of the numerical value is identified.

In using the BB algorithm to solve the energy consumption optimization problem, equation (21) is taken as the objective function with $\rho_{k\to 0}, \rho_{k\to 1}, \cdots, \rho_{k\to N}, \rho_{k\to N+1}$ as the independent variable. This objective function can be viewed as a linear programming problem that is expressed by its independent variable. The independent variable ρ meets

$$\rho_{k\to 0} + \rho_{k\to 1} + \cdots + \rho_{k\to N} + \rho_{k\to N+1} = M_k. \tag{29}$$

Equation (21) can then be expressed as

$$E_{\text{total}} = v_0\rho_{k\to 0} + v_1\rho_{k\to 1} + \cdots + v_N\rho_{k\to N} + v_{N+1}\rho_{k\to N+1}, \tag{30}$$

where $v_0, v_1, \cdots, v_N, v_{N+1}$ is the coefficient before ρ, and the coefficient vector of independent variables in the objective function can be expressed by $f = [v_0 \; v_1 \; \cdots \; v_N \; v_{N+1}]^T$. The constraint condition (1) for latency in equation (21) is then transformed as

$$D_k = D_k^0\rho_{k\to 0} + D_k^1\rho_{k\to 1} + \cdots + D_k^N\rho_{k\to N} + D_k^{N+1}\rho_{k\to N+1} \le T, \tag{31}$$

where $D_k^0, D_k^1, \cdots, D_k^N, D_k^{N+1}$ is the coefficient before ρ in the constraint condition equation (22). Constraints (5) to (7) in equation (21) can then be transformed into

$$\begin{aligned} a_{10}\rho_{k\to 0} + a_{11}\rho_{k\to 1} + \cdots + a_{1N}\rho_{k\to N} + a_{1N+1}\rho_{k\to N+1} &\le F_n, \\ a_{20}\rho_{k\to 0} + a_{21}\rho_{k\to 1} + \cdots + a_{2N}\rho_{k\to N} + a_{2N+1}\rho_{k\to N+1} &\le F_0, \\ a_{30}\rho_{k\to 0} + a_{31}\rho_{k\to 1} + \cdots + a_{3N}\rho_{k\to N} + a_{3N+1}\rho_{k\to N+1} &\le F_{N+1}. \end{aligned} \tag{32}$$

These equations transform the constraints in equation (21) into a standard form of the independent variable ρ. Let

$$A = \begin{bmatrix} D_k^0 & D_k^1 & \cdots & D_k^N & D_k^{N+1} \\ a_{10} & a_{11} & \cdots & a_{1N} & a_{1N+1} \\ a_{20} & a_{21} & \cdots & a_{2N} & a_{2N+1} \\ a_{30} & a_{31} & \cdots & a_{3N} & a_{3N+1} \\ 1 & 1 & \cdots & 1 & 1 \end{bmatrix}, \tag{33}$$

where A refers to the constraint matrix formed by this constraint equation set. Let$b = [\,T \quad F_n \quad F_0 \quad F_{N+1} \quad M_k\,]^T$, where b refers to the right vector of this constraint equation set. The value ranges of independent variable ρ can be expressed by constraints (6) to (8) of the objective function, which are denoted by lb and ub.

The basic process of the BB algorithm is shown in Table 1.

Since the BB algorithm searches the solution space in a breadth-first way, the original problem is divided into multiple branches to search for the optimal solution at the same time, eliminating a large number of nodes that have no chance to become the best value.

A local-edge-cloud edge computing network has K UE and N MEC nodes, the data of each UE is divided into M task blocks, and it is necessary to determine the allocation strategy of the UE task blocks and the offloading node of the partitioned data. The time complexity of the UE task block allocation process is determined to be $O(K2^M)$. Since multitask blocks are transmitted at the same time, there is no need for sorting by the new allocation strategy, and the optimal solution can be directly searched for the data offloading node. The computational complexity of this process is $O(K(N+2)$ $3^{M-1})$, and the sum of the two is the overall computational complexity of the BB algorithm $O(K(2^M + (N+2)3^{M-1}))$.

Table 1: Basic process of the BB algorithm.

BB algorithm
Input: coefficient vector f of the objective function, inequality constraint matrix A, right vector of inequality constraint b, upper and lower bounds of independent variables lb and ub
Output: minimize network energy consumption (E_{total}) and task allocation to different nodes ($\rho_{k\rightarrow 0}, \rho_{k\rightarrow 1}, \cdots, \rho_{k\rightarrow N}, \rho_{k\rightarrow N+1}$)
(1) Set the optimal solution $\rho = \Phi$, and the optimal upper bound of the function is $F = +\infty$.
(2) Calculate E_{total} for the initial task allocation strategy. Whether the test coefficient under this allocation strategy is nonpositive is determined by simplex.
(3) If the test coefficient is nonpositive, then the independent variable is the optimal value (ρ^*); otherwise, no optimal value is obtained, and $E_{\text{total}} = +\infty$.
(4) The value of the independent variableρ is adjusted by simplex to make all test coefficients nonpositive and meet $E^*_{\text{total}} < F$. All components of ρ^* are integers. Therefore, E^*_{total} and ρ^* are the outputs of the objective function.
(5) Let $E^*_{\text{total}} < F$. Some components of ρ^* are not integers, and the noninteger components of option ρ^* are denoted by ρ^*_k. A dichotomous approach is applied to divide the original lower constraint into two incompatible constraints.
(6) The optimal solutions to the newly formed constraints are solved by simplex. The above steps are repeated until the output x value is an integer.

5. Simulation Results

The multinode computing offloading algorithm proposed in Section 3 is compared with the traditional single cloud offload and multinode mutual transmission computing offloading algorithms. These algorithms are compared under different data sizes with overall network energy consumption as the measurement standard. For the multinode collaborative offload model, three MEC nodes are set, and $\rho_0, \rho_1, \rho_2, \rho_3, \rho_4$ represent the number of user data units at the local user ends of IoT, the three MEC nodes, and the cloud nodes, respectively. The three MEC nodes correspond to different CPU parameters, and their distances from the local user ends of IoT are denoted by d_1, d_2, d_3, respectively, assuming that the network transmission bandwidth meets user demand. The data processing situation at one local user end of IoT is initially analyzed to compare the network energy consumption of the models.

The assumption is that the network bandwidth is large enough to meet user needs; regardless of the limitation of transmission bandwidth, the effect of data transmission rate on network energy consumption is considered. Following $t_{k,i} = L_{E,i}/r_{E,i}$, the data transmission rates in the three cases are shown in Table 2.

The basic parameters used in the simulation are listed in Table 3.

5.1. Energy Consumption. The computing data size is set to $M_k \sim (1000, 2500)$ to analyze the network energy consumption of the three computing offload models. Figure 2 presents the results.

Figure 2 shows that the network energy consumption of the multinode collaborative computing offload model is lower than that of the other two models. Specifically, when the offload data size at the local user ends of IoT is smaller than 1500 kbit, the network energy consumption of the multinode collaborative computing offload model, which involves parallel data transmission, is equal to that of the multinode mutual transmission computing offload model. Otherwise, the network energy consumption of the multinode collaborative computing offload model is lower than that of the multinode mutual transmission computing offload model. The network optimization effect of the proposed model is similar to that of the multinode mutual transmission computing offload model when the offload data size is small. However, the proposed model shows some advantages in network latency that can be attributed to its parallel transmission of computing tasks. Meanwhile, when the offload data size of the network is large, the proposed model significantly outperforms the other two models in terms of network energy consumption and network latency.

The allocations of offload data size among nodes within the range of 1000 kbit to 5000 kbit are shown in Figure 3. The number of nodes for resource allocation gradually increases along with the computing offload data size. When the data size at the local user ends of IoT is not too large, the data can be processed between the local end users and MEC servers and do not need to be offloaded to cloud nodes for execution. A higher number of tasks for processing correspond to higher node number requirements. The proposed algorithm outperforms the other two models when the task data size at the local user ends of IoT is larger and is thereby conducive to optimizing the network.

When the network bandwidth is changed, the effects of information transmission rate on network energy consumption should be considered.

Figure 4 shows that the lowest network energy consumption is achieved under case 3, whereas the lowest and highest transmission rates are observed under cases 1 and 3, respectively. The overall network energy consumption is negatively correlated with network transmission rate. Given that all computing tasks are transmitted simultaneously in the proposed multinode collaborative computing offload model, a higher transmission rate leads to a larger data size for simultaneous transmission and a higher offload quantity at the local user ends of IoT. The simulation results reveal that the total data sizes under cases 1 to 3 are 5000, 7500, and 17500 kbit, respectively. In sum, the overall data size that the network can process increases along with the network transmission rate. At the same computing data size, the network energy consumption decreases along with an increasing transmission rate.

When the data size for processing at the local user ends of IoT is very large, the overall data transmission rate in the network should be increased. Specifically, when the task data size in the computing network ranges from 10000 kbit to 50000 kbit, the data transmission rate should be increased to 2 Gbit/s. The task allocation among nodes is shown in Figure 5.

Table 2: Transmission rates under the three conditions.

	Transmission rate of MEC$_1 r_{E,1}$ (Mbit/s)	Transmission rate of MEC$_2 r_{E,2}$ (Mbit/s)	Transmission rate of MEC$_3 r_{E,3}$ (Mbit/s)	Transmission rate of MCC r_c (Mbit/s)
Case 1	40	80	20	10
Case 2	80	200	40	20
Case 3	200	400	100	40

Table 3: Simulation parameter settings.

Parameter	Symbols	Value
Data volume of user K	R_k	500 kbit+ϕM_k
Data unloading unit	ϕ	1 kbit
Computing capacities of local user ends	F_0	1.26 GHz
Computing capacities of MEC nodes	F_n	16 GHz
Computing capacities of cloud nodes	F_{N+1}	64 GHz
Number of CPU turns required by the local user to calculate 1 bit of data	C_k^L	500 turn/bit
Number of CPU turns required by theMEC$_1$ node to calculate 1 bit of data	$C_{k,1}^E$	200 turn/bit
Number of CPU turns required by theMEC$_2$ node to calculate 1 bit of data	$C_{k,2}^E$	300 turn/bit
Number of CPU turns required by theMEC$_3$ node to calculate 1 bit of data	$C_{k,3}^E$	100 turn/bit
Number of CPU turns required by the cloud node to calculate 1 bit of data	C_k^C	50 turn/bit
CPU energy consumption of the local user per turn	m_k^L	10 W
CPU energy consumption of the MEC$_1$ node per turn	$m_{k,1}^E$	80 W
CPU energy consumption of the MEC$_2$ node per turn	$m_{k,2}^E$	150 W
CPU consumption of the MEC$_3$ node per turn	$m_{k,3}^E$	200 W
CPU consumption of the cloud node per turn	m_k^C	1000 W
Latency constraint of the local user	T	100 ms
Distance between the local user and MEC$_1$ node	d_1	120 m
Distance between the local user and MEC$_2$ node	d_2	100 m
Distance between the local user and MEC$_3$ node	d_3	150 m

The data transmission rate in the network increases when the data size at the local user ends of IoT is very large. Figure 5 shows that the size of data offloaded to node ρ_3 significantly increases along with the size of data at the local user ends given the low CPU energy consumption recorded at ρ_3. When the computing task is very large, the CPU energy consumption becomes a main influencing factor for network energy consumption. The total data size offloaded to the cloud nodes continuously increases along with offload data size, thereby highlighting the superiority of the edge-cloud cooperation mechanism under a large data size.

6. Conclusions

To realize green communication in smart homes, a multinode collaborative computing offload model is proposed in this paper. In this model, the local user ends of IoT segment the computing task following certain rules. Afterward, the segmented data are reasonably distributed and simultaneously transmitted to multiple nodes for implementation. The traditional single-cloud computing offload model and multinode mutual transmission computing offload model are analyzed on this basis. By treating the overall network energy consumption as the optimization goal and latency as the optimization condition, the allocation of resources among MEC nodes is determined by using a BB algorithm. The proposed model is also compared with the two aforementioned traditional models. Under a large offload task size, the proposed multinode collaborative computing offload model achieves the lowest network energy consumption and the best latency characteristics among all models. The CPU parameters of the

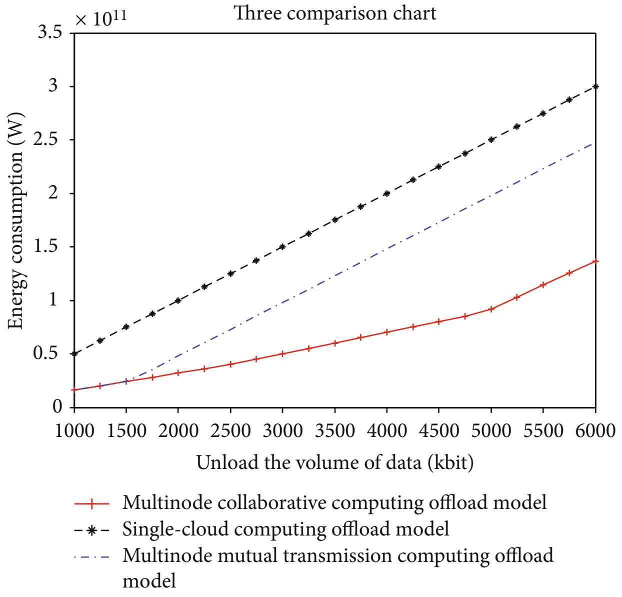

FIGURE 2: Energy consumption of the three offload models.

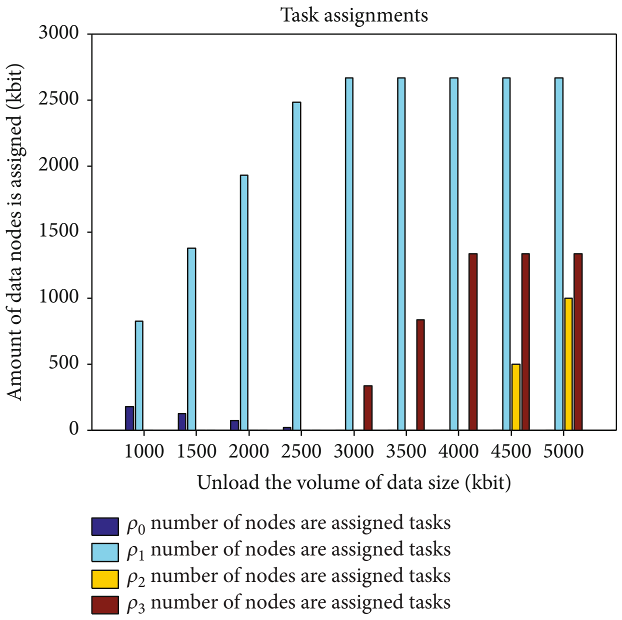

FIGURE 3: Task allocation among nodes in the multinode collaborative computing offload model.

MEC nodes greatly influence the network energy consumption. Under a large data size, the multi-MEC node and edge-cloud collaborative model show improved network characteristics. Meanwhile, both network bandwidth and information transmission rate can influence the data offload performance of the network to some extent. In a multinode collaborative computing offload model, a parallel transmission of segmented data tasks is applied to process large computing tasks at a low overall network energy consumption and high data transmission rate.

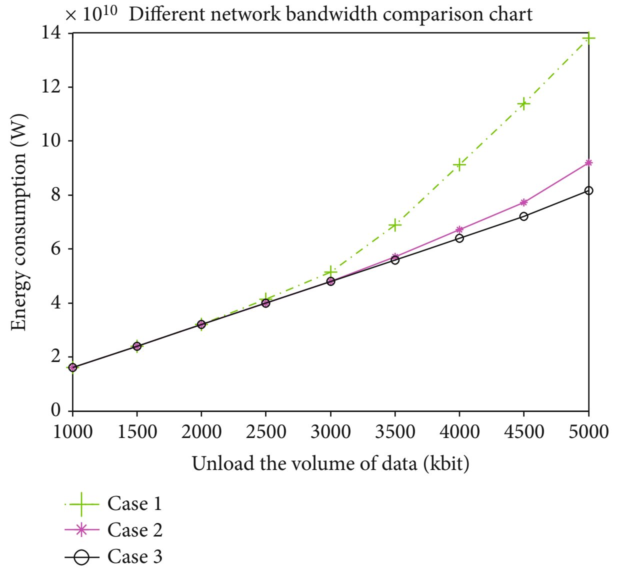

FIGURE 4: Energy consumption under different network bandwidths.

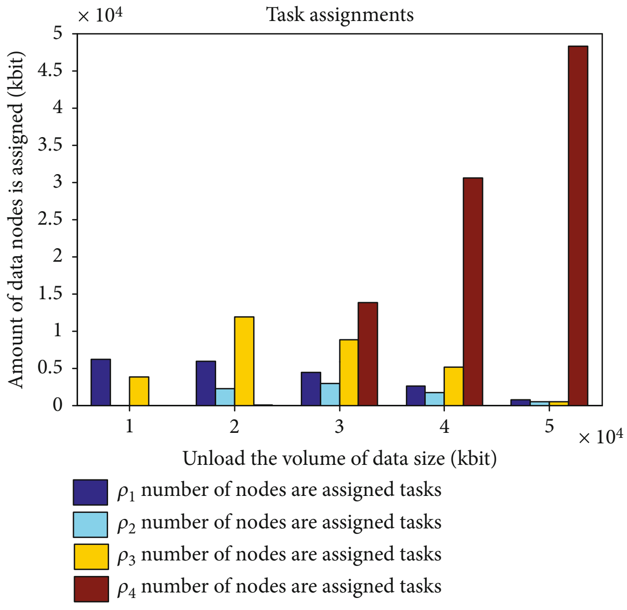

FIGURE 5: Comparison of offload schemes under a large task data size.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

We declare that there is no conflict of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (No. 61771195), the Natural Science Foundation of Hebei Province (No. F2018502047), and the Fundamental Research Funds for the Central Universities (No. 2020MS098).

References

[1] L. Atzori, A. Iera, and G. Morabito, "The Internet of things: a survey," *Computer Networks*, vol. 54, no. 15, pp. 2787–2805, 2010.

[2] H. Hui, C. Zhou, S. Xu, and F. Lin, "A novel secure data transmission scheme in industrial Internet of things," *China Communications*, vol. 17, no. 1, pp. 73–88, 2020.

[3] J. Y. Kim, H. Lee, J. Son, and J. Park, "Smart home web of objects-based IoT management model and methods for home data mining," in *2015 17th Asia-Pacific Network Operations and Management Symposium (APNOMS)*, pp. 327–331, Busan, 2015.

[4] Y. C. Hu, M. Patel, D. Sabella, N. Sprecher, and V. Young, *Mobile edge computing—a key technology towards 5G*, Eur. Telecommun. Stand, Inst, White Paper, Sophia Antipolis, France, 2015.

[5] Z. Wang, "An adaptive deep learning-based UAV receiver design for coded MIMO with correlated noise," *Physical Communication*, vol. 99, pp. 1–10, 2020.

[6] J. Liu, Y. Mao, J. Zhang, and K. B. Letaief, "Latency-optimal computation task scheduling for mobile-edge computing systems," in *Proc. IEEE Int.Symp. Inf. Theory (ISIT)*, pp. 1451–1455, Barcelona, Spain, 2016.

[7] Z. Ning, P. Dong, X. Kong, and F. Xia, "A cooperative partial computation offloading scheme for mobile edge computing enabled Internet of things," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4804–4814, 2019.

[8] L. Yang, J. Cao, H. Cheng, and Y. Ji, "Multi-user computation partitioning for latency sensitive mobile cloud applications," *IEEE Transactions on Computers*, vol. 64, no. 8, pp. 2253–2266, 2015.

[9] Y. Zhao, S. Zhou, T. Zhao, and Z. Niu, "Energy-efficient task offloading for multiuser mobile cloud computing," in *Proc. IEEE/CIC Int. Conf.Commun. China (ICCC)*, pp. 1–5, Shenzhen, China, November 2015.

[10] H. Zhao, *Research on computational unloading in resource-limited mobile edge computing system*, Beijing university of posts and telecommunications, 2019.

[11] O. Muñoz, A. Pascual-Iserte, and J. Vidal, "Optimization of radio and computational resources for energy efficiency in latency-constrained application offloading," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 10, pp. 4738–4755, 2015.

[12] C. You and K. Huang, "Multiuser resource allocation for mobileedge computation offloading," in *Proc. IEEE Glob. Commun. Conf.(GLOBECOM)*, pp. 1–6, Washington, DC, USA, December 2016.

[13] J. Chen, S. Chen, Q. Wang, B. Cao, G. Feng, and J. Hu, "iRAF: a deep reinforcement learning approach for collaborative mobile edge computing IoT networks," *IEEE Internet of Things Journal*, vol. 6, no. 4, pp. 7011–7024, 2019.

[14] W. Fan, Y. Liu, B. Tang, F. Wu, and Z. Wang, "Computation offloading based on cooperations of mobile edge computing-enabled base stations," *IEEE Access*, vol. 6, pp. 22622–22633, 2018.

[15] K. Habak, M. Ammar, K. A. Harras, and E. Zegura, "Femto clouds: leveraging mobile devices to provide cloud service at the edge," in *Proc.IEEE 8th Int. Conf. Cloud Comput.*, pp. 9–16, New York, NY, USA, June 2015.

[16] D. Deng, J. Xia, L. Fan, and X. Li, "Link selection in buffer-aided cooperative networks for green IoT," *IEEE Access*, vol. 8, pp. 30763–30771, 2020.

[17] Y. C. Hu, M. Patel, D. Sabella, N. Sprecher, and V. Young, "Mobile edge computing—a key technology towards 5G," in *Eur. Telecommun. Standards Inst.*, vol. 11, pp. 1–16, ETSI White Paper, Sophia Antipolis, France, 2015.

[18] W. Zhang, Y. Wen, K. Guan, D. Kilper, H. Luo, and D. O. Wu, "Energy-optimal mobile cloud computing under stochastic wireless channel," *IEEE Transactions on Wireless Communications*, vol. 12, no. 9, pp. 4569–4581, 2013.

[19] X. Chen, L. Jiao, W. Li, and X. Fu, "Efficient multi-user computation offloading for mobile-edge cloud computing," *IEEE/ACM Transactions on Networking*, vol. 24, no. 5, pp. 2795–2808, 2016.

[20] Z. Zhao, W. Zhou, D. Deng, J. Xia, and L. Fan, "Intelligent mobile edge computing with pricing in Internet of things," *IEEE Access*, vol. 8, pp. 37727–37735, 2020.

[21] D. Han, S. Li, Y. Peng, and Z. Chen, "Energy sharing-based energy and user joint allocation method in heterogeneous network," *IEEE Access*, vol. 8, pp. 37077–37086, 2020.

[22] D. Guo, L. Tang, X. Zhang, and Y. Liang, "Joint optimization of handover control and power allocation based on multi-agent deep reinforcement learning," *IEEE Transactions on Vehicular Technology*, 2020.

[23] D. W. K. Ng, E. S. Lo, and R. Schober, "Energy-efficient resource allocation in multi-cell OFDMA systems with limited backhaul capacity," *IEEE Transactions on Wireless Communications*, vol. 11, no. 10, pp. 3618–3631, 2012.

Hindawi
Wireless Communications and Mobile Computing
Volume 2020, Article ID 8899314, 10 pages
https://doi.org/10.1155/2020/8899314

Research Article

AN-Aided Secure Beamforming in SWIPT-Aware Mobile Edge Computing Systems with Cognitive Radio

Zhe Wang,[1,2] Taoshen Li,[1] Jin Ye,[3] Xi Yang,[4,5] and Ke Xiong[6]

[1] *School of Computer, Electronics and Information, Guangxi University, Nanning, Guangxi 530004, China*
[2] *Institute of Artificial Intelligence, Guangxi University for Nationalities, Nanning, Guangxi 530006, China*
[3] *Guangxi Key Laboratory of Multimedia Communication & Network Technology, Nanning, Guangxi 530004, China*
[4] *School of Information, Beijing Wuzi University, Beijing 101149, China*
[5] *Beijing Intelligent Logistics System Collaborative Innovation Center, Beijing 101149, China*
[6] *School of Computer and Information Technology, Beijing Jiaotong University, Beijing 100044, China*

Correspondence should be addressed to Jin Ye; yejin@gxu.edu.cn and Xi Yang; yxyoung@hotmail.com

Received 8 March 2020; Revised 15 May 2020; Accepted 1 July 2020; Published 4 November 2020

Academic Editor: Lisheng Fan

Simultaneous wireless information and power transfer (SWIPT) becomes more and more popular in cognitive radio (CR) networks, as it can increase the resource reuse rate of the system and extend the user's lifetime. Due to the deployment of energy harvesting nodes, traditional secure beamforming designs are not suitable for SWIPT-enabled CR networks as the power control and energy allocation should be considered. To address this problem, a dedicated green edge power grid is built to realize energy sharing between the primary base stations (PBSs) and cognitive base stations (CBSs) in SWIPT-enabled mobile edge computing (MEC) systems with CR. The energy and computing resource optimal allocation problem is formulated under the constraints of security, energy harvesting, power transfer, and tolerable interference. As the problem is nonconvex with probabilistic constraints, approximations based on generalized Bernstein-type inequalities are adopted to transform the problem into solvable forms. Then, a robust and secure artificial noise- (AN-) aided beamforming algorithm is presented to minimize the total transmit power of the CBS. Simulation results demonstrate that the algorithm achieves a close-to-optimal performance. In addition, the robust and secure AN-aided CR based on SWIPT with green energy sharing is shown to require a lower transmit power compared with traditional systems.

1. Introduction

In future mobile edge computing (MEC) networks, a great number of energy-limited internet of things (IoT) devices will be deployed to collect the data of system status and environment [1, 2]. Providing wireless power transfer capability is of great importance to relieve energy unbalance and prolong the system lifetime. In practice, radio frequency (RF) signals surrounding the communication systems are more controllable and stable to realize the simultaneous wireless information and power transfer (SWIPT) [3–5] in networks [6], which is a significant way to power the energy-limited edge nodes by transferring wireless energy and information simultaneously with the same waveform.

Cognitive radio (CR) is another meaningful way to utilize a spectrum efficiently for resource reusing in MEC networks. With the spectrum sharing in CR, secondary users (SUs) can coexist with the primary base station (PBS) and primary users (PUs) under the condition that the interference caused by the cognitive base station (CBS) is tolerable to PUs. Recently, many researches are considering about applying SWIPT into the CR network with energy-constrained devices to improve network efficiency and stabilization [7–11]. In [7], a SWIPT-enabled CR network was designed, in which the CBS provided SWIPT services to the secondary users, and the worst-case of SU for energy harvesting and tolerable interference power for PU were balanced based on the binary search method. This research can alleviate the double near-far problem in the energy harvesting network to a certain extent.

In light of the relay needs for network, [9] proposed an SWIPT-enabled CR framework in which the CBSs were integrated with relay capabilities to provide relay services opportunistically for the primary users. The novel framework in [9] was proposed to integrate the SWIPT-enabled CR technology into new scenarios and applications. [10] and [11], respectively, studied the wideband sensing problem under worst-case and max–min fairness case with a SWIPT-based CR framework, in order to achieve higher spectrum and energy utilization in wireless sensor networks.

Inspired by the SWIPT-enabled CR framework, we propose a novel mobile edge computing with SWIPT-enabled CR to realize the optimal allocation of energy and computing resources in this paper. CBSs and PBSs are selected with abundant energy to join the edge network, and a dedicated green edge power grid is built to achieve the energy flow and balance between those edge nodes. At the same time, we should also consider the security issues [12, 13] in the SWIPT-based network.

The characteristics of the wireless channel determine that the security rate of its physical layer is limited by channel state information (CSI) [14]. In CR networks, the transmission power of CBS in the secondary network is limited to prevent the primary users from interfering, which further affects the security rate [15]. At the same time, the formulation of a network operation strategy should take the limited energy of energy harvesting nodes (EHNs) into account. In fact, there is no cooperation among SUs, PUs, and EHNs in the CR network, so there are errors in channel estimation and quantification on SWIPT-enabled CR links that cannot be ignored. These errors seriously affect the efficiency of beamforming. Therefore, it is necessary to design a robust secure beamforming scheme for SWIPT-enabled CR MEC.

Aiming at the physical layer security of SWIPT, [16–18], respectively, studied the total transmission power minimization, safety rate maximization, and maximum minimum fairness optimization problem. On the other hand, in the SWIPT-based CR network, since the physical location of EHNs is closer to the transmitter to meet the demand of wireless power density for energy harvesting, it is possible for EHNs act as malicious eavesdroppers disturbing the reception of information receivers. Therefore, in the SWIPT design, a secure rate becomes an essential factor. In [16], a secure beamforming design in MISO SWIPT was proposed where there are multiple information receivers, energy receivers, and eavesdroppers coexisting. Similar to [16, 19], investigated a worst-case optimal problem under constraints of transmit power and harvested power in MISO SWIPT. Specifically, the secure and robust transmission strategy in the worst-case energy allocation problem for the MISO channel and multiple input and multiple output (MIMO) channel was also studied in [20, 21]. [22, 23] extended the strategy for multiple eavesdroppers with multiple antennas. For instance, an artificial noise- (AN-) aided transmit beamforming was designed in [24], which is an efficient way to improve the system secrecy rate efficiently. In [24–26], AN-aided transmission strategies for secret SWIPT were proposed in MISO, single input and single output (SISO), and MIMO channels. In [24], both perfect and imperfect CSI were considered for AN-aided SWIPT design. And with the channel uncertainties, [27] investigates the problem based on the bounded CSI and probabilistic CSI error model, respectively. However, the proposed schemes in [24–27] are difficult to be applied into the SWIPT CR network, because the energy harvesting requirement and the interference of a secondary network have not been taken into account.

This paper designs a SWIPT-enabled CR framework for MEC. By considering the energy allocation, power control, and interference constraints, an efficient and robust AN-aided secure beamforming strategy is designed to achieve better system gains. The main contributions of this paper are as follows.

First, a robust secure beamforming strategy for a CR network powered with SWIPT is proposed. The interference caused by the secondary network and the energy harvesting requirement for EHNs are both considered in our framework. By formulating as an inequality constraint in the system models, the security of the SWIPT-enabled CR network is also taken into account to minimize the transmit power.

Second, since the PBSs in the primary network and the CBSs in the secondary network are powered by green energy, we build a dedicated edge power grid made up of those PBSs and CBSs to share their redundant resources with the others. Therefore, energy balancing and computing offloading can be operated efficiently during these edge base stations in the MEC.

Third, a probabilistic CSI model is designed to characterize the CSI error, which is more suitable for delay-sensitive applications than the bounded CSI model. Subsequently, the optimal beamforming solution is solved by software CVX [28], and a 1-D algorithm is designed to approximate the optimal solution of the system with reasonable computational complexity.

Notations. For a square matrix $\mathbf{A}$, $\mathbf{A}^H$, $\mathrm{Tr}(\mathbf{A})$, and $\mathrm{Rank}(\mathbf{A})$ denote its Hermitian (conjugate) transpose, trace, and rank, respectively. $\mathbb{C}^{N\times M}$, $\mathbb{R}^{N\times M}$, and $\mathbb{H}^N$ denote the set of all the $N\times M$ complex matrices, $N\times M$ real matrices, and $N\times N$ Hermitian matrices. $\mathbf{A}\succ 0$ ($\mathbf{A}\underline{\succ}0$) indicates that matrix $\mathbf{A}$ is positive definite or semidefinite. In particular, the distribution of complex Gaussian random vectors with mean μ and variance σ^2 is denoted by (μ, σ^2).

2. System Model

The SWIPT-enabled CR MEC network is shown in Figure 1, in which the PBS and CBS act as MEC servers for users as well as the CBS providing SWIPT services for the cognitive network. There are K SUs, I PUs, and M ERs coexisting in the network. The CBS is equipped with N_t $(N_t > 1)$ antennas as the PBS has only one. The SUs and ERs are also equipped with single antenna nodes to harvest RF energy from the

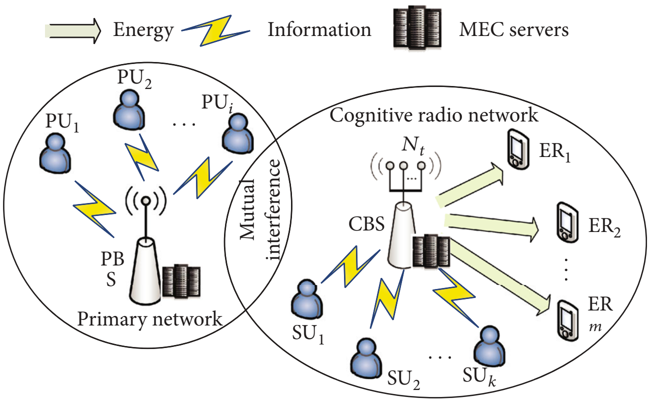

Figure 1: System model.

ambient environment. In practice, the idle users in the cognitive radio network may act as energy receivers (ERs) harvesting energy for further use, which makes the information receivers to may be eavesdrop or intercept during the communication durations. Thus, there are two problems in this system that need to be solved, which are energy sharing during edge nodes and power control for the cognitive network. We first assume that all the I PUs are friendly nodes [29–31] and all the channels are characterized by quasistatic fading models. Then, system models are performed in the next sections.

2.1. Channel Model. The signals received by SU$_k$, $k \in \{1, 2, \cdots, K\}$, and ER$_m$, $m \in \{1, 2, \cdots, M\}$, denoted by y_k^{SU} and y_m^{ER} , respectively, can be given as

$$y_k^{\mathrm{SU}} = \mathbf{h}_k^H \mathbf{x} + n_k^{\mathrm{SU}}, \tag{1}$$

$$y_m^{\mathrm{ER}} = \mathbf{g}_m^H \mathbf{x} + n_m^{\mathrm{ER}}, \tag{2}$$

where $h_k^H \in \mathbb{C}^{N_t \times 1}$ and $g_m^H \in \mathbb{C}^{N_t \times 1}$ are the channel vectors between the CBS and the kth SU and the mth ER, respectively. In (1), $x \in \mathbb{C}^{N_t \times 1}$ is the signal vector transmitted by the CBS to SUs and ERs. n_k^{SU} and n_m^{ER} are the complex Gaussian noise at SUs and ERs with zero mean and variances $\sigma_{\mathrm{SU}_k}^2$ and $\sigma_{\mathrm{ER}_m}^2$, which include terminal noise, signal processing noise, interference from PBS, and joint effects of received multicell interference.

In addition, the PU$_i$, $i \in \{1, 2, \cdots, I\}$, receiver is interfered by CBS, and the interference signal is

$$y_i^{\mathrm{PU}} = \mathbf{p}_i^H \mathbf{x}, \tag{3}$$

where $p_i^H \in \mathbb{C}^{N_t \times 1}$ is channel vector of the CBS to the ith PU receiver. The interference from the PBS to the secondary network is negligible as other researches [30–32].

2.2. CSI. Based on channel reciprocities [7, 33, and 34], it is assumed that $\mathbf{h}_k^H$, $\forall k \in \{1, 2, \cdots, K\}$, is accusable for both CBS and SUs. However, channel vectors $\mathbf{g}_m^H$ and $\mathbf{p}_i^H$ may not be known accurately, as the PUs, CBS, and ERs have no cooperation existing. As a result, we introduce probabilistic CSI models [21, 33, and 35] to characterize the outdated channel vector errors. The channel vectors follow stochastic distributions, which are

$$\begin{gathered}
\mathbf{g}_m = \widehat{\mathbf{g}}_m + \Delta \mathbf{g}_m, \\
\Delta \mathbf{g}_m \sim \mathscr{CN}(0, \mathbf{G}_m), \\
m \in \{1, 2, \cdots, M\}; \\
\mathbf{q}_i = \widehat{\mathbf{q}}_i + \Delta \mathbf{q}_i, \\
\Delta \mathbf{q}_i \sim \mathscr{CN}(0, \mathbf{Q}_i), \\
i \in \{1, 2, \cdots, I\},
\end{gathered} \tag{4}$$

where $\mathbf{g}_m$ and $\mathbf{q}_i$ are the channel estimated CSI available at the CBS and $\Delta \mathbf{g}_m$ and $\Delta \mathbf{q}_i$ denote the channel error vectors, respectively, with zero mean and covariance matrices $\mathbf{G}_m$ and $\mathbf{Q}_i$, correspondingly. In addition, $\Delta \mathbf{g}_m$ is independent of others for different m, and $\Delta \mathbf{q}_i$ is independent of others for different i.

2.3. Signal and Secure Rate Models. Actually, the ERs in the cognitive network may be illegal eavesdroppers as they have shorter transmit distance from the CBS. Thus, the secrecy of secondary users should be considered. In this paper, we propose a robust AN-aided communication strategy for CBS to efficiently transmit wireless information to end users. The transmitted signal vector $x \in \mathbb{C}^{N_t \times 1}$ is added with an artificial noise vector, which is

$$\mathbf{x} = \sum_{k=1}^{K} \mathbf{w}_k d_k + \mathbf{v}, \tag{5}$$

where $w_k \in \mathbb{C}^{N_t \times 1}$ is the dedicated beamforming vector allocated to SU$_k$ and $d_k \in \mathbb{C}$ denotes the data symbol for SU$_k$. Without loss of generality, we assumed that $\varepsilon\{|d_k|^2\} = 1$, $\forall k \in \{1, 2, \cdots, K\}$. $v \in \mathbb{C}^{N_t \times 1}$ is the AN vector transmitted by the CBS with $\mathbf{v} \sim \mathscr{CN}(0, \mathbf{V})$, in which $\mathbf{V}$ is the covariance matrix of AN and $\mathbf{V} \in H^{N_t}$, $\mathbf{V} \succeq 0$. Thus, the achievable data rate (ADR) of the transmission link between the CBS and SU$_k$ is

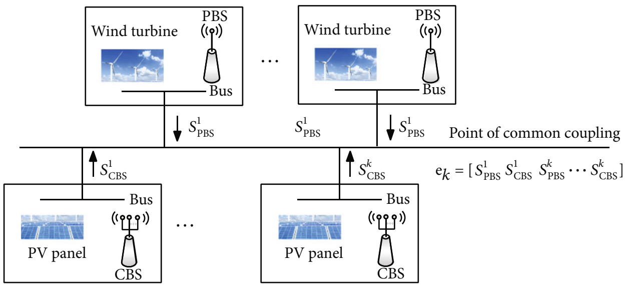

Figure 2: System power supply model.

$$C_{\mathrm{SU}_k} = \log_2\left(1 + \mathrm{SINR}_{\mathrm{SU}_k}\right), \tag{6}$$

and the signal to interference plus noise ratio (SINR) $\mathrm{SINR}_{\mathrm{SU}_k}$ can be denoted as

$$\mathrm{SINR}_{\mathrm{SU}_k} = \frac{\left|\mathbf{h}_k^H \mathbf{w}_k\right|^2}{\mathrm{Tr}\left(\mathbf{V}\mathbf{h}_k\mathbf{h}_k^H\right) + \sigma^2_{\mathrm{SU}_k}}. \tag{7}$$

On the other hand, the ADR C_{ER_m} between the CBS and ER_i and SINR of ER_m are given by

$$\begin{gathered} C_{\mathrm{ER}_m} = \log_2\left(1 + \mathrm{SINR}_{\mathrm{ER}_m}\right), \\ \mathrm{SINR}_{\mathrm{ER}_m} = \frac{\left|\mathbf{g}_m^H \mathbf{w}_k\right|^2}{\mathrm{Tr}(\mathbf{V}\mathbf{g}_m\mathbf{g}_m^H) + \sigma^2_{\mathrm{ER}_m}}. \end{gathered} \tag{8}$$

Since the ERs are potential eavesdroppers, the secrecy rate of SU_k is $R_k = \min_{m\in\{1,2,\cdots,M\}} [C_{\mathrm{SU}_k} - C_{\mathrm{ER}_m}]^+$.

2.4. System Power Supply Model. As previously described, a dedicated edge power grid is built to solve the energy sharing problem. CBSs and PBSs are selected to form a MEC network with energy sharing abilities. As shown in Figure 2, wind turbines and photovoltaic panels are appropriately prepared for MEC nodes as green energy harvesting devices. Furthermore, redundant energy of those MEC nodes can be shared with each other through the power grid. Theoretically, the SWIPT-enabled CR MEC framework with energy sharing can be easily expended for unlimited scale and infinite computing abilities. Here, we denote that the energy shared by the PBS and CBS is S_{PBS} and S_{CBS}, respectively. Then, the power loss of the grid caused by lines is given by

$$P_L = \mathbf{e}_s\mathbf{B}{\mathbf{e}_s}^T, \quad \mathbf{e}_s = \left[S^1_{\mathrm{PBS}}S^1_{\mathrm{CBS}} \cdots S^k_{\mathrm{PBS}}S^k_{\mathrm{CBS}}\right], \tag{9}$$

where $B \in \mathbb{R}^{2\times 2}$, $\mathbf{B} \succ 0$, is a B-coefficient matrix [34] which represents the energy consumption characters during its conversion process. It is noted that, for a fixed power grid topology with fixed numbers of sources and loads, the B-coefficient matrix is constant.

3. Problem Formulation

The energy balance can be realized by the sharing abilities of the MEC node based on the CR framework. In particular, the secure AN-aided SWIPT beamforming strategy is designed in each single CR network to minimize the MEC node transmit power under constraints of secrecy rate outage probability, energy harvesting, interferences, etc.:

$$\begin{aligned} P1: \quad & \min_{V,e_s,w_k} \sum_{k=1}^{K} \|w_k\|_2^2 + Tr(V) \\ \text{s.t.} \quad & C_1 : \Pr\left\{\left[C_{SU_k} - C_{ER_m}\right]^+ \geq R_{\min}\right\} \geq 1-\theta, \\ & C_2 : \Pr\left\{\mu\left(Tr(Vg_mg_m^H) + \sum_{k=1}^{K}\left|g_m^Hw_k\right|^2 + \sigma^2_{ER}\right) \geq \Omega_{ER_m}\right\} \geq 1-\theta_{ER_m}, \\ & C_3 : \Pr\left\{\sum_{k=1}^{K} q_i^H(w_kw_k^H)q_i \leq P_{\max}\right\} \geq 1-\theta_P, \\ & C_4 : P_{PBS} + P_{CBS} + \rho\left(\sum_{k=1}^{K}\|w_k\|_2^2 + Tr(V)\right) \leq e_s1 - e_sB{e_s}^T, \\ & C_5 : Tr(V) + \sum_{k=1}^{K}\|w_k\|_2^2 \leq P^{\max}_{CBS}, \\ & C_6 : 0 \leq e_s \leq e_s^{\max}, \\ & C_7 : V \underline{\succ} 0, \end{aligned} \tag{10}$$

where $R_{\min}$ denotes the CBS minimum secrecy requirement; $\theta \in (0,1]$, $\theta_{\mathrm{ER}_m} \in (0,1]$, and $\theta_P \in (0,1]$ are the maximum outage probabilities of secrecy rate, the maximum tolerable interference power at PUs, and the minimum energy harvesting requirement of kth ER, respectively; Ω_{ER_m} denotes the minimum required power transfer to ER_m; and $\mu \in (0,1]$ represents the RF-to-DC conversion efficiency of the ERs. $P^{\max}_{\mathrm{CBS}}$ in constraint C_4 indicates the allowance max-power for transmission at the CBS. In C_5, P_{PBS} and P_{CBS} represent the localized power expenditure of the PBS and the CBS; the term $\sum_{k=1}^{K}\|\mathbf{w}_k\|^2 + \mathrm{Tr}(\mathbf{V})$ is the output power of CBS, and $\rho \in (0,1]$ denotes the power efficiency of the power amplifier. The maximum power transmitting to the power grid from energy sources is constrained by C_6.

4. Robust Secure SWIPT Beamforming Design

P1 is an optimization problem with nonconvex forms and probabilistic constraints, which make P1 difficult to solve. Approximations based on Bernstein-type inequalities are first facilitated to transfer the outage probabilistic constraints C_1, C_2, and C_3 into clear and closed forms separately and safely. Then, a 1-D algorithm is proposed for computers to calculate the optimal solutions.

Lemma 1 (Bernstein-type inequalities [34, 35]). *Let $f(z) = \mathbf{z}^H\mathbf{A}\mathbf{z} + 2\,\mathrm{Re}\,\{\mathbf{z}^H\mathbf{b}\} + c$, where $\mathbf{A} \in H^{N_t}$, $\mathbf{b} \in \mathbf{C}^{N_t\times 1}$, and $c \in \mathbb{R}$. For any $\varpi \in (0,1]$, the approximate and convex forms transferred by $Pr\{f(z) \geq 0\} \geq 1-\varpi$ and $Pr\{f(z) \leq 0\} \geq 1-\varpi$ can be separately written by*

$$\begin{aligned}
&Tr(\mathbf{A})-\sqrt{-2\ln(\varpi)}v_1+\ln(\varpi)v_2+c\geq 0,\\
&\left\|\begin{bmatrix} vec(\mathbf{A})\\ \sqrt{2}\mathbf{b}\end{bmatrix}\right\|\leq v_1,\\
&v_2\mathbf{I}+\mathbf{A}\succeq 0,\\
&v_2\geq 0;\\
&Tr(\mathbf{A})+\sqrt{-2\ln(\varpi)}v_1-\ln(\varpi)v_2+c\leq 0,\\
&\left\|\begin{bmatrix} vec(\mathbf{A})\\ \sqrt{2}\mathbf{b}\end{bmatrix}\right\|\leq v_1,\\
&v_2\mathbf{I}-\mathbf{A}\succeq 0,\\
&v_2\geq 0,
\end{aligned}\tag{11}$$

where v_1 and v_2 are slack variables.

Let $\mathbf{W}_k=\mathbf{w}_k\mathbf{w}_k^H$,

$$\Psi=\frac{1}{1+\left(\left|\mathbf{h}_k^H\mathbf{w}_k\right|^2/\mathrm{Tr}\left(\mathbf{V}\mathbf{h}_k\mathbf{h}_k^H\right)+\sigma_{\mathrm{SU}_k}^2\right)}=\frac{\mathbf{h}_k^H\mathbf{V}\mathbf{h}_k+\sigma_{\mathrm{SU}_k}^2}{\mathbf{h}_k^H(\mathbf{V}+\mathbf{W})\mathbf{h}_k+\sigma_{\mathrm{SU}_k}^2},\tag{12}$$

$\Delta\mathbf{g}_m=\mathbf{G}_m^{1/2}\tilde{\mathbf{g}}_m$, $\tilde{\mathbf{g}}_m\sim\mathscr{CN}(0,\mathbf{I})$. By applying Lemma 1, C_1 can be transferred to

$$\begin{aligned}
&\mathrm{Tr}\left(\mathbf{G}_m^{1/2}\left[\mathbf{V}-2^{R_{\min}}\Psi(\mathbf{V}+\mathbf{W}_k)\right]\mathbf{G}_m^{1/2}\right)-\sqrt{-2\ln\theta}v_{s1}+\ln\theta v_{s2}\\
&\quad+\widehat{\mathbf{g}}_m^H\left[\mathbf{V}-2^{R_{\min}}\Psi(\mathbf{V}+\mathbf{W}_k)\right]\widehat{\mathbf{g}}_m+\sigma_{\mathrm{ER}_m}^2\left(1-2^{R_{\min}}\Psi\right)\geq 0,
\end{aligned}\tag{13}$$

$$\left\|\begin{bmatrix}\mathrm{vec}\left(\mathbf{G}_m^{1/2}\left[\mathbf{V}-2^{R_{\min}}\Psi(\mathbf{V}+\mathbf{W}_k)\right]\mathbf{G}_m^{1/2}\right)\\ \sqrt{2}\mathbf{G}_m^{1/2}\left[\mathbf{V}-2^{R_{\min}}\Psi(\mathbf{V}+\mathbf{W}_k)\right]\widehat{\mathbf{g}}_m\end{bmatrix}\right\|\leq v_{s1},\tag{14}$$

$$v_{s2}\mathbf{I}+\mathbf{G}_m^{1/2}\left[\mathbf{V}-2^{R_{\min}}\Psi(\mathbf{V}+\mathbf{W}_k)\right]\mathbf{G}_m^{1/2}\succeq 0,\tag{15}$$

$$v_{s2}\geq 0,\tag{16}$$

where v_{s1} and v_{s2} are slack variables.

Note. In the equation $f(z)=\mathbf{z}^H\mathbf{A}\mathbf{z}+2\,\mathrm{Re}\,\{\mathbf{z}^H\mathbf{b}\}+c$, the conversion process from C_1 to (13) is realized by

$$\begin{aligned}
&\mathbf{z}\text{ equivalents to }\tilde{\mathbf{g}}_m,\\
&A\text{ equivalents to }\mathbf{G}_m^{1/2}\left[\mathbf{V}-2^{R_{\min}}\Psi(\mathbf{V}+\mathbf{W}_k)\right]\mathbf{G}_m^{1/2},\\
&b\text{ equivalents to }\mathbf{G}_m^{1/2}\left[\mathbf{V}-2^{R_{\min}}\Psi(\mathbf{V}+\mathbf{W}_k)\right]\widehat{\mathbf{g}}_m,\\
&c\text{ equivalents to }\widehat{\mathbf{g}}_m^H\left[\mathbf{V}-2^{R_{\min}}\Psi(\mathbf{V}+\mathbf{W}_k)\right]\widehat{\mathbf{g}}_m+\sigma_{\mathrm{ER}_m}^2\left(1-2^{R_{\min}}\Psi\right).
\end{aligned}\tag{17}$$

Proof. (see the appendix).

Similar to the constraint C_1, based on Lemma 1, C_2 can be approximated as

$$\begin{aligned}
&\mathrm{Tr}\left(\mathbf{G}_m^{1/2}(\mathbf{V}+\mathbf{W}_k)\mathbf{G}_m^{1/2}\right)-\sqrt{-2\ln\theta_{\mathrm{ER}_m}}v_{e1}+\ln\theta_{\mathrm{ER}_m}v_{e2}\\
&\quad+\widehat{\mathbf{g}}_m^H(\mathbf{V}+\mathbf{W}_k)\widehat{\mathbf{g}}_m+\sigma_{\mathrm{ER}_m}^2-\frac{\Omega_{\mathrm{ER}_m}}{\mu}\geq 0,
\end{aligned}\tag{18}$$

$$\left\|\begin{bmatrix}\mathrm{vec}\left(\mathbf{G}_m^{1/2}(\mathbf{V}+\mathbf{W}_k)\mathbf{G}_m^{1/2}\right)\\ \sqrt{2}\mathbf{G}_m^{1/2}(\mathbf{V}+\mathbf{W}_k)\widehat{\mathbf{g}}_m\end{bmatrix}\right\|\leq v_{e1},\tag{19}$$

$$v_{e2}\mathbf{I}+\mathbf{G}_m^{1/2}(\mathbf{V}+\mathbf{W}_k)\mathbf{G}_m^{1/2}\succeq 0,\tag{20}$$

where v_{e1} and v_{e2} are slack variables. Let $\Delta\mathbf{q}_i=\mathbf{Q}_i^{1/2}\tilde{\mathbf{q}}_i$, $\tilde{\mathbf{q}}_i\sim\mathscr{CN}(0,\mathbf{I})$, using Lemma 1, constraint C_3 is approximated by

$$\begin{aligned}
&\mathrm{Tr}\left(\mathbf{Q}_i^{1/2}(\mathbf{V}+\mathbf{W}_k)\mathbf{Q}_i^{1/2}\right)+\sqrt{-2\ln\theta_P}v_{p1}-\ln\theta_P v_{p2}\\
&\quad+\widehat{\mathbf{q}}_i^H(\mathbf{V}+\mathbf{W}_k)\widehat{\mathbf{q}}_i-P_{\max}\geq 0,
\end{aligned}\tag{21}$$

$$\left\|\begin{bmatrix}\mathrm{vec}\left(\mathbf{Q}_i^{1/2}(\mathbf{V}+\mathbf{W}_k)\mathbf{Q}_i^{1/2}\right)\\ \sqrt{2}\mathbf{Q}_i^{1/2}(\mathbf{V}+\mathbf{W}_k)\widehat{\mathbf{q}}_i\end{bmatrix}\right\|\leq v_{p1},\tag{22}$$

$$v_{p2}\mathbf{I}-\mathbf{Q}_i^{1/2}(\mathbf{V}+\mathbf{W}_k)\mathbf{Q}_i^{1/2}\succeq 0,\tag{23}$$

where v_{p1} and v_{p2} are slack variables.

Although the approximated forms of (13), (18), and (21) are deterministic forms, the problem P1 is still difficult to solve due to the nonconvex caused by fractional and coupling constraints. In particular, we take advantage of the characteristics of fractional parameter Ψ, which can be seen that $0<\Psi\leq 1$. Thus, for a given Ψ, suboptimal computed solutions can be obtained by a proposed 1-D search method. The relaxation of P1 is

$$\begin{aligned}
\mathrm{P2}:\quad&\min_{\mathbf{V},\mathbf{e}_s,\mathbf{W}_k,\{v_{s1}\},\{v_{s2}\},\{v_{e1}\},\{v_{e2}\},\{v_{p1}\},\{v_{p2}\}}\sum_{k=1}^{K}\mathrm{Tr}(\mathbf{W}_k)+\mathrm{Tr}(\mathbf{V})\\
\text{s.t.}\quad&\mathbf{h}_k^H((\Psi-1)\mathbf{V}+\Psi\mathbf{W}_k)\mathbf{h}_k+(\Psi-1)\sigma_{\mathrm{SU}_k}^2=0\\
&(1),\ (2),\ (3),\ C4-C7\\
&C_8: R(\mathbf{W}_k)\leq 1,\ \mathbf{W}_k\succeq 0,\ \mathbf{W}_k\in\mathbb{H}^{N_t},
\end{aligned}\tag{24}$$

in which C_8 guarantees that $\mathbf{W}_k=\mathbf{w}_k\mathbf{w}_k^H$. As C_8 is a rank-one, a safe relaxation is introduced to make P2 convex by removing C_8 from P2's constraints. Then, the optimal solutions are obtained by convex programming software CVX [28].

Algorithm 1 is the detail of the 1-D algorithm. In this case, parameters θ, θ_{ER_m}, and θ_P have to be initialized and the search interval of the uniform search variable Ψ is (0,1]. It is noted that the problem P2 cannot make the matrices $\mathbf{W}_k$ rank-one. If the solutions $\mathbf{W}_k$ are rank-one for all SUs, the optimal secure beamforming vector can be obtained through eigenvalue decomposing of $\mathbf{W}_k$. If the solutions $\mathbf{W}_k$

```
1: Initialization:$R_{\min}, \theta, \mu, \Omega_{ER_m}, \theta_{ER_m}, P_{\max}, \theta_P, P_{CBS}^{\max}, P_{PBS}, P_{CBS}, \rho$.
2: Input:$\mathbf{h}_k^H, k = 1, 2, \cdots, K, \widehat{\mathbf{g}}_m, m = 1, 2, \cdots, M, \widehat{\mathbf{q}}_i, i = 1, 2, \cdots, I$.
3: Set: The iteration index $n$=1, the step length factor of the
1-D search $\tau$.
4: Optimization:
5: for$\Psi$=$\tau$:$\tau$:1
6: solve the P3,
    $\text{P3}: \min_{\mathbf{V}, \mathbf{e}_s, \mathbf{w}_k, \{v_{s1}\}, \{v_{s2}\}, \{v_{e1}\}, \{v_{e2}\}, \{v_{p1}\}, \{v_{p2}\}} \sum_{k=1}^{K} \text{Tr}(\mathbf{W}_k) + \text{Tr}(\mathbf{V})$
        $s.t.\ \mathbf{h}_k^H((\Psi - 1)\mathbf{V} + \Psi\mathbf{W}_k)\mathbf{h}_k + (\Psi - 1)\sigma_{SU_k}^2 = 0$
                (1), (2), (3), C4-C7
      obtain $\{\mathbf{W}_1, \mathbf{W}_2, \cdots, \mathbf{W}_K\}^n, \mathbf{V}^n, \mathbf{e}_s{}^n, (\sum_{k=1}^{K} \text{Tr}(\mathbf{W}_k) + \text{Tr}(\mathbf{V}))^n$;
      set $n$=$n$+1.
7: end
8: Comparison: Find the optimal $\{\mathbf{W}_1, \mathbf{W}_2, \cdots, \mathbf{W}_K\}^{opt}, \mathbf{V}^{opt}, \mathbf{e}_s{}^{opt}, (\sum_{k=1}^{K} \text{Tr}(\mathbf{W}_k) + \text{Tr}(\mathbf{V}))^{opt}$from the $n$ $\{\mathbf{W}_1, \mathbf{W}_2, \cdots, \mathbf{W}_K\}^n, \mathbf{V}^n, \mathbf{e}_s{}^n$,
$(\sum_{k=1}^{K} \text{Tr}(\mathbf{W}_k) + \text{Tr}(\mathbf{V}))^n$
```

Algorithm 1: The 1-D search algorithm.

Table 1: Parameters for simulation.

Parameters	Values
Minimum secrecy rate requirement	$R_{\min} = 1.5\text{bits/s/Hz}$
Energy conversion efficiency	$\mu = 0.9$
Minimum required power of ER_m	$\Omega_{\text{ER}_m} = 5\text{ W}$
Maximum tolerable interference at PU	$P_{\max} = 5\text{ W}$
Outage probabilities	$\theta = 0.05, \theta_{\text{ER}_m} = 0.05, \theta_P = 0.05$
Maximum transmit power allowance at the CBS	$P_{\text{CBS}}^{\max} = 10\text{ W}$
Fixed power expenditure	$P_{\text{PBS}} = 50\text{ W}, P_{\text{CBS}} = 50\text{ W}$
Channel power gains	$\mathbf{h}_k \sim \mathscr{CN}(0, \mathbf{I}), \widehat{\mathbf{g}}_m \sim CN(0, \mathbf{I}), \widehat{\mathbf{q}}_i \sim CN(0, \mathbf{I})$
Variances of noise	$\sigma_{\text{SU}_k}^2 = \sigma_{\text{ER}_m}^2 = 0.05$

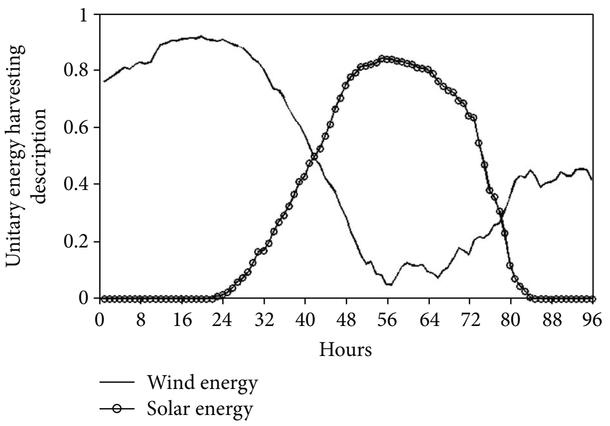

Figure 3: System power supply model.

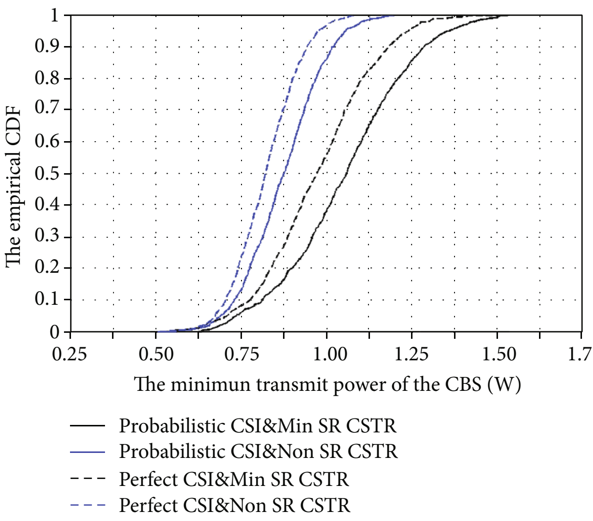

Figure 4: The CDF of the minimum power for transmission of the CBS.

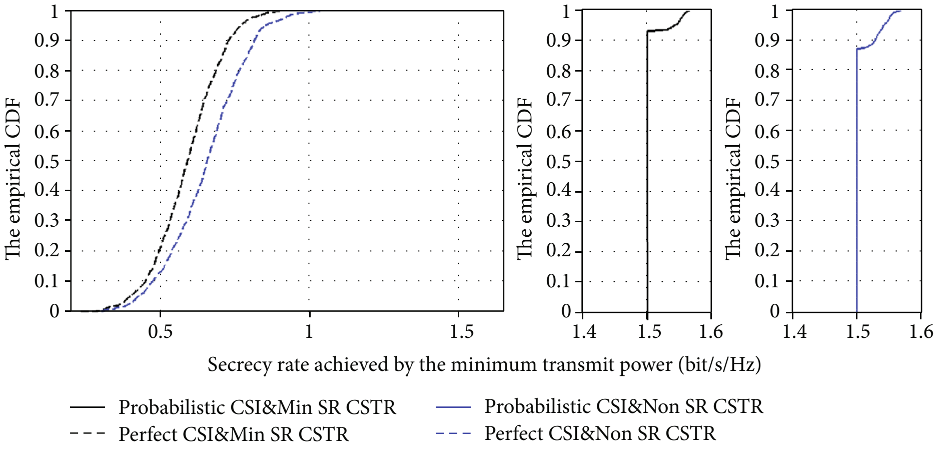

FIGURE 5: The CDF of the secrecy rate achieved by the minimum transmit power of the CBS.

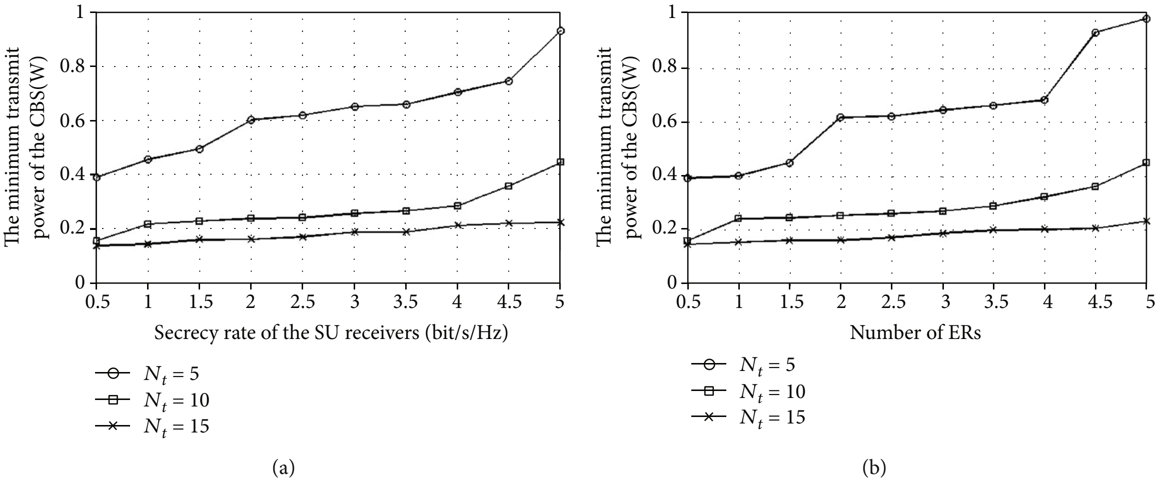

FIGURE 6: The CBS minimum transmit power under different secrecy rate requirements and different numbers of ERs. (a) The relationship between the CBS minimum transmit power and the number of ERs. (b) The relationship between the CBS minimum transmit power and the secrecy rate requirements.

are not rank-one, Gaussian randomization procedure can be used to find the suboptimal solutions. Meanwhile, the corresponding optimal objective value provides the lower bound of (24) [35].

5. Simulation Results

Based on the proposed 1-D algorithm, this section illustrates the system gains by simulations. The SWIPT-enabled CR MEC network is considered with 2 SUs, 3 ERs, and 4 PUs. The antennas of the CBS are $N_t = 5$. Table 1 details the parameters for simulations. We discuss the system gains realized by the secure beamforming compared with ideal cases.

The real energy harvesting data are descripted in Figure 3, which are collected by Guangxi Power Grid Co., Ltd with the intervals of fifteen minutes. In practice, the information transmission coherence times are small enough, which can be ignored compared with the power transfer intervals. Thus, $\mathbf{e}_s^{\max}$ is supposed to be a known constant; that is to say, the PBS and CBS can use energy harvested in current intervals based on the knowledge of harvesting status. Those approaches contribute to the constraint C_5 in problem formulations; the energy cooperation is realized between PBS and CSB until the problem is solved and the average minimum energy sequences are obtained.

Figure 4 is 500 calculations for the CDF of the CBS optimized transmit power under the probabilistic CSI with minimum secrecy rate (SR) constraint, the probabilistic CSI without minimum SR constraint, the perfect CSI with minimum SR constraint, and the perfect CSI without minimum SR constraint, separately. It can be seen that the minimum power transmitted by the CBS under the probabilistic CSI error models is bigger than the other scenes as the channel uncertainty takes more energy consumptions to meet the expected secrecy rate. Figure 5 indicates the CDF of the secrecy rate of the 500 calculations, from which we can see that with the minimum secrecy rate constraint, the system can achieve higher security rates and the imperfect CSI significantly reduce the system preference. It is worth noting that the cumulative probabilities under probabilistic CSI and perfect CSI scenes are started from the position of the lowest safety rate, and the minimum secrecy rate is 1.5 bit/s/Hz. The vertical parts of the CDF curves are caused by the introduction of secrecy rate constraints.

The minimum power transmitted by CBS is shown in Figure 6 under different secrecy rate requirements and different numbers of ERs. N_t is set as $N_t = 5$, $N_t = 10$, or $N_t = 15$. Figure 6(a) concluded that a higher secure rate results in greater energy consumption. Furthermore, the transmit power of CBS decreases with N_t increases; this is because more antennas degrees the source allocation with higher degrees of freedom. Figure 6(b) describes the relationship between ERs and CBS optimal transmit power.

6. Conclusions

This paper proposes a SWIPT-enabled CR framework for MEC networks. A dedicated edge power grid is built for sharing energy among MEC nodes to balance the energy distribution. In particular, robust AN-aided secure beamforming is designed for CR networks to realize secure communications and optimal power control. Bernstein-type inequalities are introduced to transfer the problem into closed and safely relaxed forms. The solutions are obtained with the designed 1-D algorithm. Simulation results proved the effectiveness and performance of the proposed scheme. System gains are also achieved with the cost of implementation complexities.

Appendix

The Proof of Lemma 1

Let $\mathbf{W}_k = \mathbf{w}_k\mathbf{w}_k^H$ and $\Psi = (\mathbf{h}_k^H\mathbf{V}\mathbf{h}_k + \sigma_{\mathrm{SU}_k}^2)/(\mathbf{h}_k^H(\mathbf{V}+\mathbf{W}_k)\mathbf{h}_k + \sigma_{\mathrm{SU}_k}^2)$, we can rewrite the inequality $[C_{\mathrm{SU}_k} - C_{\mathrm{ER}_m}]^+ \geq R_{\min}$ as the following form:

$$\frac{\mathbf{g}_m^H\mathbf{V}\mathbf{g}_m + \sigma_{\mathrm{ER}_m}^2}{\Psi\left(\mathbf{g}_m^H(\mathbf{V}+\mathbf{W}_k)\mathbf{g}_m + \sigma_{\mathrm{ER}_m}^2\right)} - 2^{R_{\min}} \geq 0. \tag{A.1}$$

Due to the nonnegative of the denominator in the above inequality, we have

$$\mathbf{g}_m^H\mathbf{V}\mathbf{g}_m + \sigma_{\mathrm{ER}_m}^2 - 2^{R_{\min}}\left(\Psi\left(\mathbf{g}_m^H(\mathbf{V}+\mathbf{W}_k)\mathbf{g}_m + \sigma_{\mathrm{ER}_m}^2\right)\right) \geq 0. \tag{A.2}$$

After the equation $\mathbf{g}_m = \widehat{\mathbf{g}}_m + \Delta\mathbf{g}_m$, $\Delta\mathbf{g}_m \sim \mathscr{CN}(0, \mathbf{G}_m)$, $m \in \{1, 2, \cdots, M\}$, is substituted and performed some algebraic manipulations; (A.2) can be rewritten as

$$\begin{aligned}&\Delta\mathbf{g}_m^H\left[\mathbf{V} - 2^{R_{\min}}\Psi(\mathbf{V}+\mathbf{W}_k)\right]\Delta\mathbf{g}_m + \Delta\mathbf{g}_m^H\left[\mathbf{V} - 2^{R_{\min}}\Psi(\mathbf{V}+\mathbf{W}_k)\right]\widehat{\mathbf{g}}_m \\ &\quad + \widehat{\mathbf{g}}_m^H\left[\mathbf{V} - 2^{R_{\min}}\Psi(\mathbf{V}+\mathbf{W}_k)\right]\Delta\mathbf{g}_m + \widehat{\mathbf{g}}_m^H\left[\mathbf{V} - 2^{R_{\min}}\Psi(\mathbf{V}+\mathbf{W}_k)\right]\widehat{\mathbf{g}}_m \\ &\quad + \sigma_{\mathrm{ER}_m}^2\left(1 - 2^{R_{\min}}\Psi\right) \geq 0.\end{aligned} \tag{A.3}$$

Let $\Delta\mathbf{g}_m = \mathbf{G}_m^{1/2}\tilde{\mathbf{g}}_m$, $\tilde{\mathbf{g}}_m \sim \mathscr{CN}(0, \mathbf{I})$, then the first term of the polynomial of the left side in the inequality (A.3) is rewritten as

$$\begin{aligned}&\Delta\mathbf{g}_m^H\left[\mathbf{V} - 2^{R_{\min}}\Psi(\mathbf{V}+\mathbf{W}_k)\right]\Delta\mathbf{g}_m \\ &\quad = \tilde{\mathbf{g}}_m{}^H\mathbf{G}_m^{1/2H}\left[\mathbf{V} - 2^{R_{\min}}\Psi(\mathbf{V}+\mathbf{W}_k)\right]\mathbf{G}_m^{1/2}\tilde{\mathbf{g}}_m.\end{aligned} \tag{A.4}$$

Based on the expression $f(z) = \mathbf{z}^H\mathbf{A}\mathbf{z} + 2\ \mathrm{Re}\ \{\mathbf{z}^H\mathbf{b}\} + c$ in Lemma 1, $\mathbf{A}$ is equivalent to $\mathbf{G}_m^{1/2H}[\mathbf{V} - 2^{R_{\min}}\Psi(\mathbf{V}+\mathbf{W}_k)]\mathbf{G}_m^{1/2}$ and $\mathbf{z}$ is equivalent to $\tilde{\mathbf{g}}_m$. Then, the rest of polynomial is converted as

$$\begin{aligned}&\Delta\mathbf{g}_m^H\left[\mathbf{V} - 2^{R_{\min}}\Psi(\mathbf{V}+\mathbf{W}_k)\right]\widehat{\mathbf{g}}_m + \widehat{\mathbf{g}}_m^H\left[\mathbf{V} - 2^{R_{\min}}\Psi(\mathbf{V}+\mathbf{W}_k)\right]\Delta\mathbf{g}_m \\ &\quad = \tilde{\mathbf{g}}_m{}^H\mathbf{G}_m^{1/2H}\left[\mathbf{V} - 2^{R_{\min}}\Psi(\mathbf{V}+\mathbf{W}_k)\right]\widehat{\mathbf{g}}_m + \widehat{\mathbf{g}}_m^H\left[\mathbf{V} - 2^{R_{\min}}\Psi\right. \\ &\qquad \left.(\mathbf{V}+\mathbf{W}_k)\right]\mathbf{G}_m^{1/2}\tilde{\mathbf{g}}_m = \tilde{\mathbf{g}}_m{}^H\mathbf{G}_m^{1/2}\quad\left[\mathbf{V} - 2^{R_{\min}}\Psi(\mathbf{V}+\mathbf{W}_k)\right]\widehat{\mathbf{g}}_m \\ &\qquad\quad + \widehat{\mathbf{g}}_m^H\left[\mathbf{V} - 2^{R_{\min}}\Psi(\mathbf{V}+\mathbf{W}_k)\right]\mathbf{G}_m^{1/2}\tilde{\mathbf{g}}_m \\ &\quad = 2\ \mathrm{Re}\ \left\{\tilde{\mathbf{g}}_m{}^H \cdot \mathbf{G}_m^{1/2}\left[\mathbf{V} - 2^{R_{\min}}\Psi(\mathbf{V}+\mathbf{W}_k)\right]\widehat{\mathbf{g}}_m\right\}.\end{aligned} \tag{A.5}$$

Therefore, $\mathbf{b}$ in the expression $f(z) = \mathbf{z}^H\mathbf{A}\mathbf{z} + 2\ \mathrm{Re}\ \{\mathbf{z}^H\mathbf{b}\} + c$ is equivalent to $\mathbf{G}_m^{1/2}[\mathbf{V} - 2^{R_{\min}}\Psi(\mathbf{V}+\mathbf{W}_k)]\widehat{\mathbf{g}}_m$, and c is equivalent to $\widehat{g}_m^H[V - 2^{R_{\min}}\Psi(V + W_k)]\widehat{g}_m + \sigma_{ER_m}^2(1 - 2^{R_{\min}}\Psi)$.

Finally, the constraint C_1 can be equivalently expressed as (A.6), where υ_{e1} and υ_{e2} are slack variables:

$$\begin{aligned}\Psi = {}& \frac{\mathbf{h}_k^H\mathbf{V}\mathbf{h}_k + \sigma_{\mathrm{SU}_k}^2}{\mathbf{h}_k^H(\mathbf{V}+\mathbf{W}_k)\mathbf{h}_k + \sigma_{\mathrm{SU}_k}^2}\mathrm{Tr}\left(\mathbf{G}_m^{1/2}\left[\mathbf{V} - 2^{R_{\min}}\Psi(\mathbf{V}+\mathbf{W}_k)\right]\mathbf{G}_m^{1/2}\right) \\ & - \sqrt{-2\ln\theta}\upsilon_{s1} + \ln\theta\upsilon_{s2} + \widehat{\mathbf{g}}_m^H\left[\mathbf{V} - 2^{R_{\min}}\Psi(\mathbf{V}+\mathbf{W}_k)\right]\widehat{\mathbf{g}}_m \\ & + \sigma_{\mathrm{ER}_m}^2\left(1 - 2^{R_{\min}}\Psi\right) \geq 0,\end{aligned} \tag{A.6}$$

$$\left\|\begin{bmatrix}\mathrm{vec}\left(\mathbf{G}_m^{1/2}\left[\mathbf{V} - 2^{R_{\min}}\Psi(\mathbf{V}+\mathbf{W}_k)\right]\mathbf{G}_m^{1/2}\right) \\ \sqrt{2}\mathbf{G}_m^{1/2}\left[\mathbf{V} - 2^{R_{\min}}\Psi(\mathbf{V}+\mathbf{W}_k)\right]\widehat{\mathbf{g}}_m\end{bmatrix}\right\| \leq \upsilon_{s1}, \tag{A.7}$$

$$\upsilon_{s2}\mathbf{I} - \mathbf{G}_m^{1/2}\left[\mathbf{V} - 2^{R_{\min}}\Psi(\mathbf{V}+\mathbf{W}_k)\right]\mathbf{G}_m^{1/2}\succeq 0, \tag{A.8}$$

$$\upsilon_{s2} \geq 0. \tag{A.9}$$

This completes the proof.

Data Availability

This paper is based on mathematical models of wireless networks including the channel model, communication model, energy model, and computing models.

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant 61762010, in part by the Guangxi University for Nationalities Introduction of Talents Research Project Foundation under Grant 2019KJQD17, in part by the Natural Science Foundation of Guangxi under Grant 2018GXNSFAA294155, and also in part by the Beijing Intelligent Logistics System Collaborative Innovation Center under Grant BILSCIC-2019KF-07.

References

[1] C.-H. Hsu, S. Wang, Y. Zhang, and A. Kobusinska, "Mobile edge computing," *Wireless Communications and Mobile Computing*, vol. 2018, Article ID 7291954, 3 pages, 2018.

[2] J. Zhang, W. Xie, F. Yang, and Q. Bi, "Mobile edge computing and field trial results for 5G low latency scenario," *China Communications*, vol. 13, no. 2, pp. 174–182, 2016.

[3] R. L. Varshney, "Transporting information and energy simultaneously," in *2008 IEEE International Symposium on Information Theory*, Toronto, ON, Canada, July 2008.

[4] R. Zhang and C. K. Ho, "MIMO broadcasting for simultaneous wireless information and power transfer," *IEEE Transactions on Wireless Communications*, vol. 12, no. 5, pp. 1989–2001, 2013.

[5] K. Xiong, C. Chen, G. Qu, P. Fan, and K. B. Letaief, "Group cooperation with optimal resource allocation in wireless powered communication networks," *IEEE Transactions on Wireless Communications*, vol. 16, no. 6, pp. 3840–3853, 2017.

[6] K. Xiong, B. Wang, and K. J. R. Liu, "Rate-energy region of SWIPT for MIMO broadcasting under nonlinear energy harvesting model," *IEEE Transactions on Wireless Communications*, vol. 16, no. 8, pp. 5147–5161, 2017.

[7] F. Zhou, N. C. Beaulieu, J. Cheng, Z. Chu, and Y. Wang, "Robust max-min fairness resource allocation in sensing-based wideband cognitive radio with SWIPT: imperfect channel sensing," *IEEE Systems Journal*, vol. 12, no. 3, pp. 2361–2372, 2018.

[8] R. Liu and W. Trappe, *In Securing Wireless Communications at the Physical Layer*, Springer-Verlag, New York, NY, USA, 1st edition, 2010.

[9] Y. Pei, Y.-c. Liang, L. Zhang, K. Teh, and K. Li, "Secure communication over MISO cognitive radio channels," *IEEE Transactions on Wireless Communications*, vol. 9, no. 4, pp. 1494–1502, 2010.

[10] D. W. K. Ng, E. S. Lo, and R. Schober, "Robust beamforming for secure communication in systems with wireless information and power transfer," *IEEE Transactions on Wireless Communications*, vol. 13, no. 8, pp. 4599–4615, 2014.

[11] Y. Lu, K. Xiong, P. Fan, Z. Ding, Z. Zhong, and K. B. Letaief, "Global energy efficiency in secure MISO SWIPT systems with non-linear power-splitting EH model," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 1, pp. 216–232, 2019.

[12] J. Liu, Y. Gu, L. Zha, Y. Liu, and J. Cao, "Event-Triggered H_∞ Load Frequency Control for Multiarea Power Systems Under Hybrid Cyber Attacks," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 49, no. 8, pp. 1665–1678, 2019.

[13] L. Zha, E. Tian, X. Xie, Z. Gu, and J. Cao, "Decentralized event-triggered H_∞ control for neural networks subject to cyber-attacks," *Information Sciences*, vol. 457-458, no. 457, pp. 141–155, 2018.

[14] D. W. K. Ng and R. Schober, "Max-min in fair wireless energy transfer for secure multiuser communication systems," in *IEEE Information Theory Workshop (ITW 2014)*, pp. 326–330, Hobart, TAS, Australia, 2014.

[15] R. Feng, Q. Li, Q. Zhang, and J. Qin, "Robust Secure Transmission in MISO Simultaneous Wireless Information and Power Transfer System," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 1, pp. 400–405, 2015.

[16] J. Guang and A. L. Swindlehurst, "Robust secure transmission in MISO channels based on worst-case optimization," *IEEE Transactions on Signal Processing*, vol. 60, pp. 1696–1702, 2012.

[17] W. Wu, X. J. Yin, P. Deng, T. W. Guo, and B. Y. Wang, "Transceiver Design for Downlink SWIPT NOMA Systems With Cooperative Full-Duplex Relaying," *IEEE Access*, vol. 7, pp. 33464–33472, 2019.

[18] W. Wu, F. H. Zhou, P. Li, P. Deng, B. Y. Wang, and V. C. M. Leung, "Energy-efficient resource allocation for secure NOMA-enabled mobile edge computing networks," *IEEE Transactions on Communications*, vol. 68, no. 1, pp. 493–505, 2020.

[19] Q. Li and W. K. Ma, "Spatially selective artificial-noise aided transmit optimization for MISO multi-eves secrecy rate maximization," *IEEE Transactions on Signal Processing*, vol. 61, no. 10, pp. 2704–2717, 2013.

[20] W. C. Liao, T. H. Chang, W. K. Ma, and C. Chi, "QoS-based transmit beamforming in the presence of eavesdroppers: an optimized artificial noise-aided approach," *IEEE Transactions on Signal Processing*, vol. 59, no. 3, pp. 1202–1216, 2011.

[21] D. W. K. Ng, E. S. Lo, and R. Schober, "Energy-efficient resource allocation for secure OFDMA systems," *IEEE Transactions on Vehicular Technology*, vol. 61, no. 6, pp. 2572–2585, 2012.

[22] Y. Lu, K. Xiong, P. Fan, Z. Zhong, and K. B. Letaief, "Robust transmit beamforming with artificial redundant signals for secure SWIPT system under non-linear EH model," *IEEE Transactions on Wireless Communications*, vol. 17, no. 4, pp. 2218–2232, 2018.

[23] F. Zhou, Z. Li, J. Cheng, Q. Li, and J. Si, "Robust AN-aided beamforming and power splitting design for secure MISO cognitive radio with SWIPT," *IEEE Transactions on Wireless Communications*, vol. 16, no. 4, pp. 2450–2464, 2017.

[24] M. Grant and S. Boyd, "CVX: Matlab software for disciplined convex programming, version 2.0 beta," 2013, http-s://http://cvxr.com/cvx.

[25] X. Lu, P. Wang, D. Niyato, D. I. Kim, and Z. Han, "Wireless networks with RF energy harvesting: a contemporary survey," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 2, pp. 757–789, 2015.

[26] R. K. Sharma and D. B. Rawat, "Advances on security threats and countermeasures for cognitive radio networks: a survey," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 2, pp. 1023–1043, 2015.

[27] N. Mokari, S. Parsaeefard, H. Saeedi, P. Azmi, and E. Hossain, "Secure robust ergodic uplink resource allocation in relay-assisted cognitive radio networks," *IEEE Transactions on Signal Processing*, vol. 63, no. 2, pp. 291–304, 2015.

[28] B. Fang, Z. Qian, W. Zhang, and W. Shao, "An-aided secrecy precoding for SWIPT in cognitive MIMO broadcast channels,"

IEEE Communications Letters, vol. 19, no. 9, pp. 1632–1635, 2015.

[29] Z. Chu, H. Xing, M. Johnston, and S. Le Goff, “Secrecy Rate Optimizations for a MISO Secrecy Channel with Multiple Multiantenna Eavesdroppers,” *IEEE Transactions on Wireless Communications*, vol. 15, no. 1, pp. 283–297, 2016.

[30] Z. Chu, Z. Zhu, M. Johnston, and S. Y. Le Goff, “Simultaneous wireless information power transfer for MISO secrecy channel,” *IEEE Transactions on Vehicular Technology*, vol. 65, no. 9, pp. 6913–6925, 2016.

[31] S. Ma and D. Sun, “Chance constrained robust beamforming in cognitive radio networks,” *IEEE Communications Letters*, vol. 17, no. 1, pp. 67–70, 2013.

[32] A. J. Wood and B. F. Wollenberg, “Power generation operation and control — 2nd edition,” *Fuel and Energy Abstracts*, vol. 37, no. 3, 1996.

[33] K.-Y. Wang, A. M.-C. So, T.-H. Chang, W.-K. Ma, and C.-Y. Chi, “Outage constrained robust transmit optimization for multiuser MISO downlinks: tractable approximations by conic optimization,” *IEEE Transactions on Signal Processing*, vol. 62, no. 21, pp. 5690–5705, 2014.

[34] I. Bechar, “A Bernstein-type inequality for stochastic processes of quadratic forms of Gaussian variables,” *Mathematics*, 2009, http://arxiv.org/abs/ 0909.3595v1.

[35] Z. Luo, W. Ma, A. M. So, Y. Ye, and S. Zhang, “Semidefinite relaxation of quadratic optimization problems,” *IEEE Signal Processing Magazine*, vol. 27, no. 3, pp. 20–34, 2010.

Hindawi
Wireless Communications and Mobile Computing
Volume 2020, Article ID 8891793, 12 pages
https://doi.org/10.1155/2020/8891793

Research Article

Impact of Hardware Impairments with Imperfect Channel Estimation for Cache-Enabled UAV Relaying Networks

Dan Deng,[1] Yanyi Rao ,[2] and Fusheng Zhu [3]

[1] *School of Information Engineering, Guangzhou Panyu Polytechnic, Guangzhou 511483, China*
[2] *School of Computer Science and Cyber Engineering, Guangzhou University, Guangzhou 510006, China*
[3] *Guangdong New Generation Communication and Network Innovative Institute (GDCNi), Guangzhou, China*

Correspondence should be addressed to Yanyi Rao; raoyy@gzhu.edu.cn and Fusheng Zhu; zhufushengcom@126.com

Received 27 April 2020; Revised 10 June 2020; Accepted 26 June 2020; Published 10 August 2020

Academic Editor: Lisheng Fan

The effects of hardware impairments with imperfect channel estimation for cache-enabled UAV networks are studied in this paper. The effects of the setup parameters, such as the number of cached or relaying nodes, the hardware impairment factor, the channel estimation error, and the transmission SNR, on the outage probability are present by deriving the exact closed-form expressions on outage probability. Also, the asymptotic results are present when the transmission SNR is large enough. From the asymptotic analysis, we can see that there is an error floor introduced by the hardware impairments and imperfect channel estimation. Specifically, the error floor is dependent on the channel estimation error factor, the hardware impairment distortion factor, and the QoS requirements. And the diversity order of the error floor for the cached/relaying links is equal to the number of cached/relaying UAV nodes.

1. Introduction

In recent years, wireless throughput is growing up exponentially [1–3] and wireless communication network is facing more and more challenges introduced by wireless big data [4–7]. Cache technology, which can both improve the quality of service (QoS) of wireless link and relieve the pressure of wireless throughput, has demonstrated as a remarkable enabling solution for wireless networks [8–10]. By exploiting interference neutralization, the authors in [11] proposed an optimal content placement method to improve the system capacity. Furthermore, the maximum distance separable coding is adopted in wireless cache networks in [12], which can improve the system performance in terms of successful retrieval probability.

On the other hand, hardware impairments, which are introduced by phase noise of the RF components, show great effects on the system performance. Hardware impairments have been extensively studied and modeled as additive distortion noise [13] or nonlinear polynomial multiplicative factor [14, 15]. Considering amplify-and-forward (AF) relaying networks, the authors in [16] investigated the effects of hardware impairments on channel capacity by exact closed-form expression as well as the tight bounds, while Ref. [17] proved that larger hardware impairments can be used in massive MIMO due to the huge degrees of space freedom. By using iterative optimization algorithm, Ding et al. [18] proposed a solution to maximize the detection probability.

Meanwhile, due to the high mobility, there exists imperfect channel estimation since the linear estimation methods are applied on pilot signals [19]. Considering correlated Rayleigh fading with imperfect channel status information, Al-Hussaibi and Ali [20] derived expressions on the ergodic capacity with antenna selection. The authors in [21] analyzed the impact of imperfect CSI estimation for Alamouti-OSTBC Wireless Cooperative Networks in terms of symbol error probability.

However, the impacts of hardware impairments with imperfect channel estimation on relaying networks are still open questions. Considering unmanned aerial vehicle (UAV) networks, the UAV nodes work as a cache enable access point or an AF relaying node. To improve the QoS of the wireless channels, the best cached UAV nodes or AF relaying nodes are selected to help the message transmission from the macro BS to the user. The effects of the system parameters, such as the number of cached/relaying nodes, the hardware impairment factor, the channel estimation error, and the transmission SNR, on the outage probability are present by deriving the exact closed-form expressions on outage probability. Also, the asymptotic results are present when the transmission SNR is large enough. From the asymptotic analysis, we can see that there is an error floor introduced by the hardware impairments and imperfect channel estimation. Specifically, the error floor is dependent on the channel estimation error factor, the hardware impairment distortion factor, and the QoS requirements. And the diversity order of the error floor for the cached/relaying links is equal to the number of cached/relaying UAV nodes.

The main contributions of this paper are as follows:

(i) Cache-enabled UAV relaying protocol is adopted to improve the QoS of the wireless links, where the UAV nodes work as a cache enable access point or an AF relaying node

(ii) We provide the deep insight on the effects of the system parameters, such as the number of cached/relaying nodes, the hardware impairment factor, the channel estimation error, and the transmission SNR, on the outage probability by deriving the exact closed-form expressions on outage probability

The organization of this paper is as follows. Section 2 presents the system model of cache-enabled amplify-and-forward UAV networks, while the UAV selection algorithm and its performance analysis are given in Section 3. Furthermore, the asymptotical analysis is present in Section 4 with large transmission power. In Section 5, simulation results are present to validate the theoretical analysis. Finally, conclusion is given in Section 6.

2. System Model

The system model of cache-enabled amplify-and-forward unmanned aerial vehicle (UAV) networks is present in Figure 1, where there is one destination user, one macro base station (BS) and N cache-enabled UAV relaying nodes. We assumed that all nodes are equipped with only one antenna. Furthermore, because of the shadow fading, there is no direct link from the macro BS to the destination user. In order to improve the QoS of the links, one of the UAV nodes is selected to assist the message transmission. Since the UAV nodes hold limited cache memory, the backhaul load from the macro BS to the UAV nodes can be relieved if the requested files from the user are hit in the UAV nodes. In this case, one of the UAV nodes is selected and directly transmits message to the user. On the other hand, if the requested file is not cached, the selected UAV node will work as an amplify-and-forward relaying node to help the data transmission. It is assumed that all links are block Rayleigh fading channel and independent with each other.

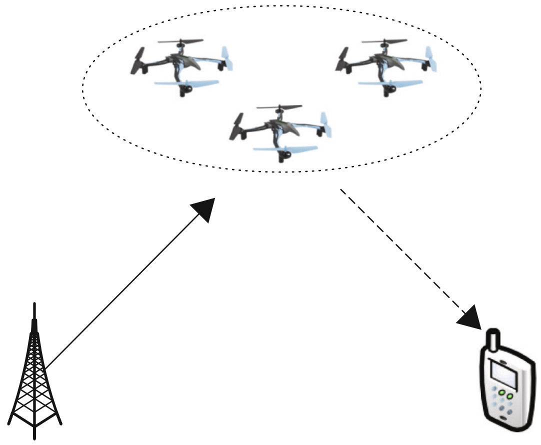

Figure 1: System model of cache-enabled amplify-and-forward UAV networks.

3. Selection Algorithm and Performance Analysis

3.1. Cached Links. If the requested file is cached by the UAV nodes, the power normalized transmission signal from the UAV node can be modeled as [13]

$$s_i = \sqrt{\frac{1}{1+\kappa_\eta}}\left(\widehat{s}_i + \sqrt{\kappa_\eta}\eta_i\right), \tag{1}$$

where $\widehat{s}_i$ is the information bearing signal with unit power, $\eta_i \in \mathscr{CN}(0,1)$ is the distortion noise introduced by hardware impairments [22–25], and κ_η is the hardware impairment distortion factor.

Since the classical MMSE channel estimation algorithm is adopted, there exists some estimation error between the real channel fading coefficient and the estimated version [26–29]. It is assumed that the estimation error is independent with the imperfect CSI [30–35]. Thus, the relationship between the imperfect channel status information and the ideal version is given as

$$h_{u,i} = \sqrt{\frac{1}{1+\kappa_e}}\left(\widehat{h}_{u,i} + \sqrt{\kappa_e e_i}\right), \tag{2}$$

where $h_{u,i} \in \mathscr{CN}(0,1)$ is the CSI from the ith UAV node to the user, while $\widehat{h}_{u,i} \in \mathscr{CN}(0,1)$ denotes the imperfect estimation of h_i, $e_i \in \mathscr{CN}(0,1)$ is the difference between $\widehat{h}_{u,i}$ and $h_{u,i}$, and κ_e is the channel estimation error factor. Specifically, $\kappa_e = 0$ means the perfect channel estimation.

Given that the ith UAV relaying node is selected, the requested file is cached by the selected node. Then, the received signal of the user can be given as

$$\begin{aligned} y_i &= \sqrt{P_R} h_{u,i} s_i + n_i \\ &= \sqrt{\frac{P_R}{(1+\kappa_e)\left(1+\kappa_\eta\right)}} \left(\widehat{h}_{u,i} + \sqrt{\kappa_e} e_i\right)\left(\widehat{s}_i + \sqrt{\kappa_\eta}\eta_i\right) + n_i, \end{aligned} \tag{3}$$

where P_R is the transmission power of the UAV node, and $n_i \in \mathscr{CN}(0, \sigma^2)$ is the received white Gauss noise.

Then, the signal to interference plus noise ratio (SINR) can be given as

$$\gamma_{c,i} = \frac{\rho_{u,i}}{\rho_{u,i}\kappa_\eta + d_1}, \tag{4}$$

where

$$\begin{cases} \rho_{u,i} = \left|\widehat{h}_{u,i}\right|^2, \\ \lambda_R = \dfrac{P_R}{\sigma^2}, \\ d_1 = \kappa_e\left(1+\kappa_\eta\right) + (1+\kappa_e)\dfrac{\left(1+\kappa_\eta\right)}{\lambda_R}. \end{cases} \tag{5}$$

Obviously, $\gamma_{c,i}$ is a monotone increasing function of $\rho_{u,i}$. In this case, the optimal UAV node can be given as follows:

$$n_1^* = \arg \max_{i\in\Omega_C} \left(\rho_{u,i}\right). \tag{6}$$

Due to the independence between different $\rho_{u,i}$'s, according to the ordered statistics, we have

$$F_{\rho_{u,n_1^*}}(x) = \left[F_{\rho_{u,i}}(x)\right]^{N_R} = \left[1 - e^{-x}\right]^{N_R}, \tag{7}$$

where N_R is the number of relaying nodes.

Then, the outage probability of the cached links can be given as

$$P_{O,C} = \Pr[C_C < C_0] = Pr\left[\gamma_{r,n_1^*} < \gamma_0\right] = Pr\left[\rho_{u,n_1^*} < \frac{d_1\gamma_0}{1-\kappa_\eta\gamma_0}\right], \tag{8}$$

where $\gamma_0 = 2^{C_0} - 1$ and C_0 are the required capacity.

Substituting (7) into (8), we obtain

$$\begin{aligned} P_{O,C} &= Pr\left[\rho_{u,n_1^*} < \frac{d_1\gamma_0}{1-\kappa_\eta\gamma_0}\right] \\ &= \left[1 - \exp\left(\frac{d_1\gamma_0}{1-\kappa_\eta\gamma_0}\right)\right]^{N_R}, \quad \gamma_0 < \frac{1}{\kappa_\eta}. \end{aligned} \tag{9}$$

3.2. Relaying Links. On the other hand, if the requested file is not cached in the selected UAV node, the original message will be transmitted from the macro BS to the user with the help of the UAV node by using amplify-and-forward relaying protocol.

In this case, the received signal at the ith UAV node can be given as

$$r_i = h_{r,i}\sqrt{\frac{P_S}{1+\kappa_\eta}}\left(\widehat{s}_i + \sqrt{\kappa_\eta}\eta_i\right) + n_{r,i}, \tag{10}$$

where P_S is the transmission power of the macro BS, $h_{r,i} \in \mathscr{CN}(0,1)$ denotes the channel fading coefficient from the macro BS to the ith UAV node, and $n_{r,i} \in \mathscr{CN}(0,\sigma^2)$ is the received Gauss noise at the UAV node.

Then, the UAV node will amplify the received signal with power P_R and retransmit to the user. Thus, the received signal at the user through AF protocol can be given as

$$y_i = h_{u,i}\zeta_i r_i + n_i, \tag{11}$$

where ζ_i is the power amplify factor as follows

$$\zeta_i = \sqrt{\frac{P_R}{\rho_{r,i}P_S + \sigma^2}}, \tag{12}$$

and $\rho_{r,i} = |h_{r,i}|^2$ is the channel fading power of the first hop.

By substituting (2) and (12) into (11), we can obtain the SINR at the user through AF protocol

$$\gamma_{r,i} = \frac{\rho_{r,i}\rho_{u,i}}{\rho_{r,i}\rho_{u,i}\kappa_\eta + \rho_{r,i}C_r + \rho_{u,i}C_u + C_n}, \tag{13}$$

where

$$\begin{cases} \rho_{r,i} = \left|\widehat{h}_{r,i}\right|^2, \\ \lambda_S = \dfrac{P_S}{\sigma^2}, \\ \lambda_R = \dfrac{P_R}{\sigma^2}, \\ c_r = \kappa_e\left(1+\kappa_\eta\right), \\ c_u = \dfrac{(1+\kappa_e)\left(1+\kappa_\eta\right)}{\lambda_R + \sigma^2}, \\ c_n = \kappa_e\sigma^2 + \dfrac{(1+\kappa_e)\left(1+\kappa_\eta\right)}{\lambda_S/\lambda_R}. \end{cases} \tag{14}$$

In order to obtain the best SINR, an UAV selection algorithm can be employed. Note that if the transmission power is large enough, i.e., $\lambda_S \to 0$, $\lambda_R \to 0$, the high order items can be ignored. Then, $\gamma_{r,i}$ in (13) can be simplified as

$$\gamma_{r,i} \simeq \left[\kappa_\eta + \frac{c_r}{\rho_{u,i}} + \frac{c_u}{\rho_{r,i}}\right]^{-1}. \tag{15}$$

To simply the complexity of theoretical analysis, we use the following approximation. Recall the following fact that [36]

$$\frac{1}{\min(x,y)} \le \frac{1}{x} + \frac{1}{y} \le \frac{2}{\min(x,y)}, \quad \forall x, y > 0. \tag{16}$$

To obtain the best SINR through AF protocol, the selection criterion can be given as

$$n_2^* = \arg\max_{i\in\Omega_R} \min\left(\rho_{u,i}, \xi\rho_{r,i}\right), \tag{17}$$

where $\xi = c_r/c_u$.

Theorem 1. *The cumulative distribution functions of ρ_{u,n_2^*} and ρ_{r,n_2^*} are given as*

$$\begin{cases} F_{\rho_{u,n_2^*}}(x) = N_R \sum_{k=1}^{N_R-1} b_k \left[\frac{1}{\xi_2} - \frac{e^{-x}}{\xi_3} + \frac{e^{-\xi_2 x}}{(\xi_3\xi_2)}\right], \\ F_{\rho_{r,n_2^*}}(x) = N_R \sum_{k=1}^{N_R-1} b_k \left[\frac{1}{\xi_2} - \frac{e^{-x}}{\xi_4} + \frac{e^{-\xi_2\xi x}}{(\xi\xi_2\xi_4)}\right], \end{cases} \tag{18}$$

for $N_R \ge 2$.

$$F_{\rho_{u,n_2^*}}(x) = F_{\rho_{r,n_2^*}}(x) = 1 - e^{-x}, \tag{19}$$

for $N_R = 1$, where

$$\begin{cases} \xi_1 = 1 + \frac{1}{\xi}, \\ \xi_2 = (k+1)\xi_1, \\ \xi_3 = k\xi_1 + \frac{1}{\xi} = \xi_2 - 1, \\ \xi_4 = k\xi_1 + 1 = \xi_2 - \frac{1}{\xi}, \\ b_k = C_{N_{R-1}}^k (-1)^{k-1} k\xi_1. \end{cases} \tag{20}$$

Taking the derivative of equations in Theorem 1, we can obtain the following corollary.

Corollary 2. *The probability distribution functions of ρ_{u,n_2^*} and ρ_{r,n_2^*} are given as*

$$\begin{cases} f_{\rho_{u,n_2^*}}(x) = N_R \sum_{k=1}^{N_R-1} \frac{b_k}{\xi_3\left(e^{-x} - e^{-\xi_2 x}\right)}, \\ f_{\rho_{r,n_2^*}}(x) = N_R \sum_{k=1}^{N_R-1} \frac{b_k}{\xi_4\left(e^{-x} - e^{-\xi_2\xi x}\right)}, \end{cases} \tag{21}$$

for $N_R \ge 2$, and

$$f_{\rho_{u,n_2^*}}(x) = f_{\rho_{r,n_2^*}}(x) = e^{-x}, \tag{22}$$

for $N_R = 1$.

Proof. See Appendix VI-A.

Considering the SINR expression in (13), the outage occurs when γ_{r,n_2^*} falls below the quality-of-service requirement, i.e.,

$$P_{O,R} = \Pr[C_R < C_0] = \Pr\left[\gamma_{r,n_2^*} < \gamma_0\right], \tag{23}$$

where $\gamma_0 = 2^{C_0} - 1$.

Substituting (13) into (24), we have

$$\begin{aligned} P_{O,R} &= Pr\left[\frac{\rho_{r,i}\rho_{u,i}}{\rho_{r,i}\rho_{u,i}\kappa_\eta + \rho_{r,i}C_r + \rho_{u,i}C_u + C_n} < \gamma_0\right] \\ &= Pr\left[\rho_{r,i}\rho_{u,i}\left(1 - \gamma_0\kappa_\eta\right) < \left(\rho_{r,i}C_r\gamma_0 + \rho_{u,i}C_u\gamma_0 + \gamma_0 C_n\right)\right] \\ &= Pr\left[\rho_{r,i}\left[\rho_{u,i}\left(1 - \gamma_0\kappa_\eta\right) - C_r\gamma_0\right] < \left(\rho_{u,i}C_u\gamma_0 + \gamma_0 C_n\right)\right]. \end{aligned} \tag{24}$$

Applying conditional probability formula, we have

$$\begin{aligned} P_{O,R} &= Pr\left[\rho_{r,i}\left[\rho_{u,i}\left(1 - \gamma_0\kappa_\eta\right) - c_r\gamma_0\right] < \left(\rho_{u,i}c_u\gamma_0 + \gamma_0 c_n\right)\right] \\ &= Pr\left[\rho_{u,i} < c_1\right] + Pr\left[\rho_{r,i} < c_2\frac{\rho_{u,i} + c_n/c_u}{\rho_{u,i} - c_1}, \rho_{u,i} \ge c_1\right] \\ &= \underbrace{Pr\left[\rho_{u,i} < c_1\right]}_{P_1} + \underbrace{Pr\left[\rho_{r,i} \le c_2, \rho_{u,i} \ge c_1\right]}_{P_2} \\ &\quad \underbrace{+ Pr\left[c_2 < \rho_{r,i} < c_2\frac{\rho_{u,i} + c_n/c_u}{\rho_{u,i} - c_1}, \rho_{u,i} \ge c_1\right]}_{P_3}, \end{aligned} \tag{25}$$

where

$$\begin{cases} c_1 = \dfrac{c_r \gamma_0}{\left(1-\gamma_0 \kappa_\eta\right)}, \\ c_2 = \dfrac{c_u \gamma_0}{\left(1-\gamma_0 \kappa_\eta\right)}, \\ c_3 = \dfrac{c_n}{\left(c_1 c_u\right)}. \end{cases} \tag{26}$$

Applying (18) on (25), we can obtain

$$P_1 = F_{\rho_{u,n_2^*}}(c_1), \tag{27}$$

$$P_2 = F_{\rho_{r,n_2^*}}(c_2)\left[1 - F_{\rho_{u,n_2^*}}(c_1)\right]. \tag{28}$$

To derive the exact expression of P_3, firstly we give the following definition

$$\begin{aligned} F_Z(z) &= Pr\left[\frac{\rho_{u,i} + c_n/c_u}{\rho_{u,i} - c_1} < z, \rho_{u,i} \geq c_1\right], \quad z > 1 \\ &= Pr\left[\rho_{u,i} > c_1 \frac{z + c_n/(c_1 c_u)}{z-1}, \rho_{u,i} \geq c_1\right], \quad z > 1 \\ &= Pr\left[\rho_{u,i} > c_1 \frac{z + c_n/(c_1 c_u)}{z-1}\right], \quad z > 1 \\ &= Pr\left[\rho_{u,i} > c_1 \frac{z + c_3}{z-1}\right], \quad z > 1. \end{aligned} \tag{29}$$

Applying (18) on (29), we can obtain

$$F_Z(z) = \Pr\left[\rho_{u,i} > c_1 \frac{z + c_3}{z-1}\right], \quad z > 1 = F_{\rho_{u,n_2^*}}\left(c_1 \frac{z + c_3}{z-1}\right). \tag{30}$$

Thus, the item P_3 in equation (25) can be given as

$$\begin{aligned} P_3 &= \Pr\left[c_2 < \rho_{r,i} < c_2 \frac{\rho_{u,i} + c_n/c_u}{\rho_{u,i} - c_1}, \rho_{u,i} \geq c_1\right] \\ &= \int_{x>c_2}^{+\infty} f_{\rho_{r,n_2^*}}(x)\left[1 - F_Z\left(\frac{x}{c_2}\right)\right] dx \\ &= \int_{x>c_2}^{+\infty} f_{\rho_{r,n_2^*}}(x)\left[1 - F_{\rho_{u,n_2^*}}\left(c_1 \frac{x + c_3 c_2}{x - c_2}\right)\right] dx. \end{aligned} \tag{31}$$

By using necessary mathematical derivation and applying equation (3.324-1) in [37], i.e.,

$$\int_0^{+\infty} \exp\left(\frac{\beta}{4x} - \gamma x\right) dx = \sqrt{\frac{\beta}{\gamma}} K_1\left(\sqrt{\beta\gamma}\right), \tag{32}$$

the closed-form expression of P_3 can be obtained.

By substituting (27), (28), and (31) into (25), we can obtain the exact closed-form expression of $P_{O,R}$.

4. Asymptotical Analysis

To get a deep insight on the impact of hardware impairments with imperfect channel estimation, we will conduct the asymptotical analysis of outage probability. Note that if the transmission power is large enough, i.e., $1/\lambda_S \to 0$, $1/\lambda_R \to 0$, and $\kappa_e \to 0$, $\kappa_\eta \to 0$.

Thus, we have

$$\begin{aligned} &\begin{cases} c_r \simeq \kappa_e\left(1 + \kappa_\eta\right), \\ c_u \simeq \sigma^2, \\ c_n \simeq \kappa_e \sigma^2 \\ d_1 \simeq \kappa_e\left(1 + \kappa_\eta\right), \\ \xi \simeq \dfrac{\kappa_e\left(1 + \kappa_\eta\right)}{\sigma^2}, \end{cases} \\ &\begin{cases} c_1 \simeq \dfrac{\kappa_e\left(1 + \kappa_\eta\right)\gamma_0}{\left(1 - \gamma_0 \kappa_\eta\right)}, \\ c_2 \simeq \dfrac{\sigma^2 \gamma_0}{\left(1 - \gamma_0 \kappa_\eta\right)}, \\ c_3 \simeq \dfrac{\left(1 - \gamma_0 \kappa_\eta\right)}{\left(1 + \kappa_\eta\right)\gamma_0}. \end{cases} \end{aligned} \tag{33}$$

Applying the approximation that $1 - e^{-x} \simeq x$, if $x \to 0$ on (9), we have

$$P_{O,C} \simeq \left(\frac{d_1 \gamma_0}{1 - \kappa_\eta \gamma_0}\right)^{N_C} \simeq \left[\gamma_0 \frac{\kappa_e\left(1 + \kappa_\eta\right)}{1 - \kappa_\eta \gamma_0}\right]^{N_C}. \tag{34}$$

According to the asymptotical expression on outage probability for cached links, we can conclude the following remarks:

Remark 3. There is an error floor for the cached links when the transmission power is large enough. Meanwhile, the diversity order with respect to the transmission power is zero.

Remark 4. The error floor for the cached links is dependent on the channel estimation error factor κ_e, the hardware impairment distortion factor κ_η, and the QoS requirements γ_0.

Remark 5. The diversity order of error floor for the cached links is equal to the number of cached UAV nodes.

Considering the SINR expression in (15), we have

$$\begin{aligned} P_{O,R} &\simeq Pr\left[\max_{i \in \Omega_R} \min\left(\rho_{u,i}, \xi \rho_{r,i}\right) < \gamma_0 \frac{\xi c_u}{\left(1 - \gamma_0 \kappa_\eta\right)}\right] \\ &= \left\{Pr\left[\min\left(\rho_{u,i}, \xi \rho_{r,i}\right) < c_4\right]\right\}^{N_R}. \end{aligned} \tag{35}$$

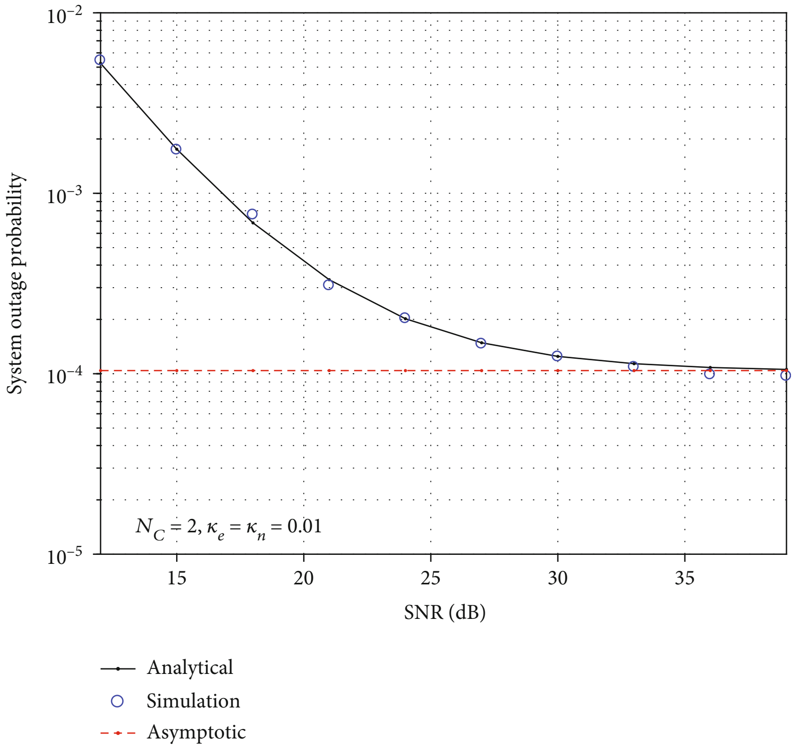

Figure 2: Outage probability for cached link with $N_C = 2$.

By using the result in (A.3), we have

$$P_{O,R} \simeq \left(1 - e^{-\xi_1 c_4}\right)^{N_R} \simeq (\xi_1 c_4)^{N_R} \simeq \left[\gamma_0 \frac{(1+\xi)\sigma^2}{(1-\gamma_0\kappa_\eta)}\right]^{N_R} \simeq \left[\gamma_0 \frac{\sigma^2 + \kappa_e(1+\kappa_\eta)}{(1-\gamma_0\kappa_\eta)}\right]^{N_R}, \quad (36)$$

where

$$c_4 = \gamma_0 \frac{\xi c_u}{(1-\gamma_0\kappa_\eta)} \simeq \gamma_0 \frac{\xi\sigma^2}{(1-\gamma_0\kappa_\eta)}. \quad (37)$$

Remark 6. There also exists an error floor for the relaying links when the transmission power is large enough.

Remark 7. The diversity order of error floor for the relaying links is equal to the number of relaying UAV nodes.

5. Simulation Results

In this section, simulation results are provided to verify the accuracy of the theoretical analysis. For the sake of simplicity, we set $\kappa_e = \kappa_\eta = \sigma^2$. The impacts of the system parameters, such as the number of UAV nodes N, the transmission SNR of the cached links and the relaying links λ_S, λ_R, and the capacity threshold C_0, are investigated.

Figure 2 shows the impacts of transmission SNR λ_S on outage performance for cached links. In this simulation, we have the following system parameters: $N_C = 2, \kappa_e = \kappa_\eta = 0.01$, $C_0 = 1$ bps/Hz, and the transmission SNR λ_S changes from 12 dB to 39 dB. The simulation results and the theoretical analysis as well as the asymptotic analysis are compared in this figure. As shown in this figure, in all SNR regions, the simulation results match well with the theoretical analysis. Furthermore, an error floor appears when the transmission SNR grows large, which coincides with the asymptotic performance analysis. The reason is that when SNR is large enough, the bottleneck of the system performance is the hardware impairments and the channel estimation error. The detailed relationship between the error floor and κ_η and κ_e is given as in equations (34) and (36). When the requested file is not hit by the UAV nodes, amplify-and-forward protocol is adopted and the simulation results are present in Figure 3, where similar conclusion can be obtained for relaying links with $N_R = 2$.

Figures 4 and 5 show the effects of the capacity threshold C_0 on outage probability for cached links and relaying links, respectively. The system parameters are set as follows: $N_C = N_R = 2, \kappa_e = \kappa_\eta = 0.01$, and the capacity threshold C_0 changed from 1 bps/Hz to 3 bps/Hz. From the two figures, we can observe that C_0 shows significant impact on outage probability for all of the SNR regions. Specifically, the diversity order of outage probability is zero with respect to the transmission SNR λ_S or λ_R.

Figures 6 and 7 show the effects of the hardware impairment factor κ_η on outage probability for cached links and relaying links, respectively. The system parameters are set as follows: $N_C = N_R = 2, \kappa_e = \kappa_\eta = 0.01, C_0 = 1$ bps/Hz, and

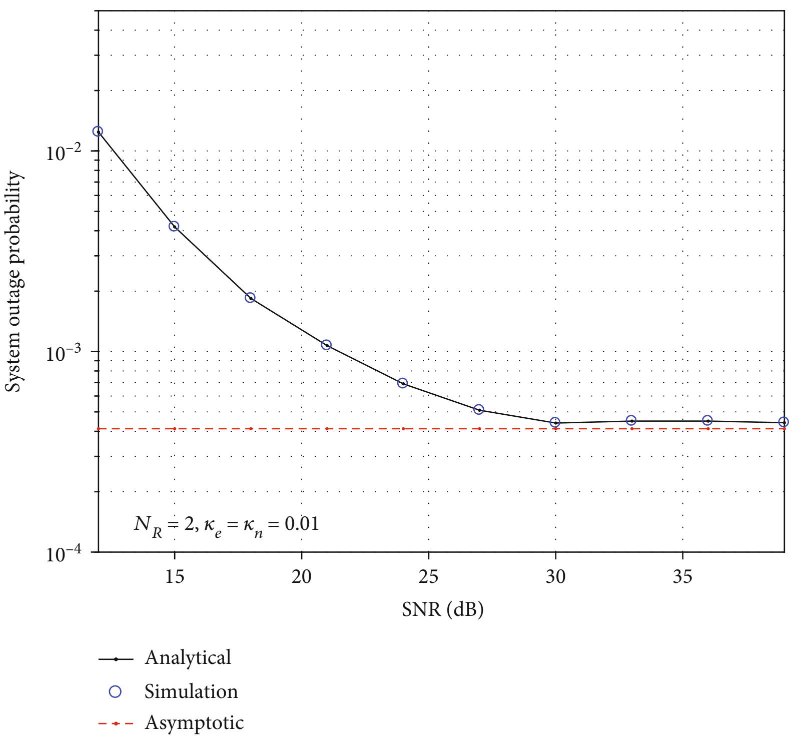

FIGURE 3: Outage probability for AF relaying links with $N_R = 2$.

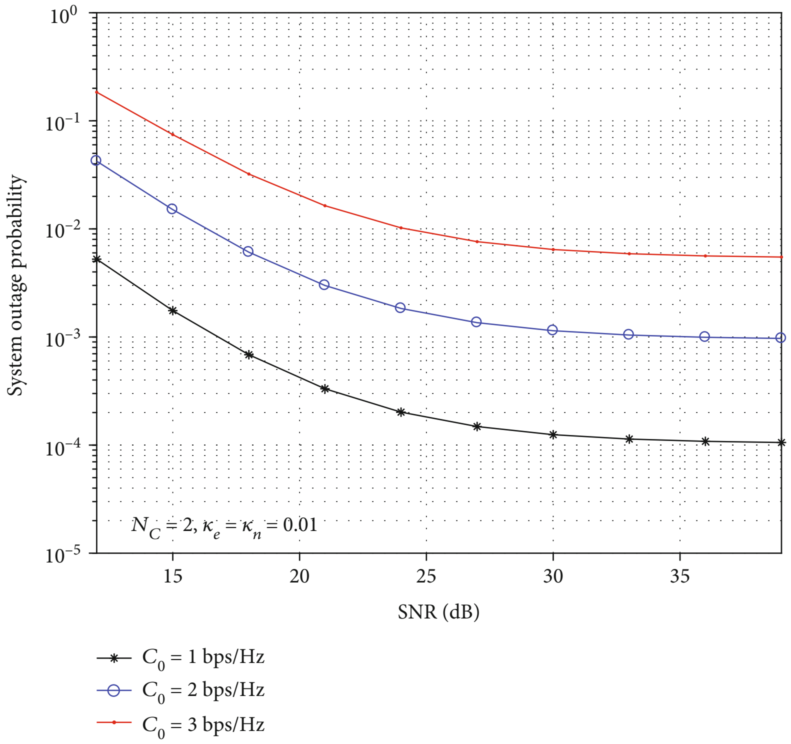

FIGURE 4: Outage probability versus capacity threshold for cached links with $N_C = 2$.

the hardware impairment factor κ_η changed from 0.01 to 0.03. From these figures, we can observe that C_0 shows considerable impact on outage probability especially in high SNR regions. Specifically, the larger κ_η will introduce higher error floor. The reason is that, in the considered scenario, the bottleneck of the system performance is the hardware

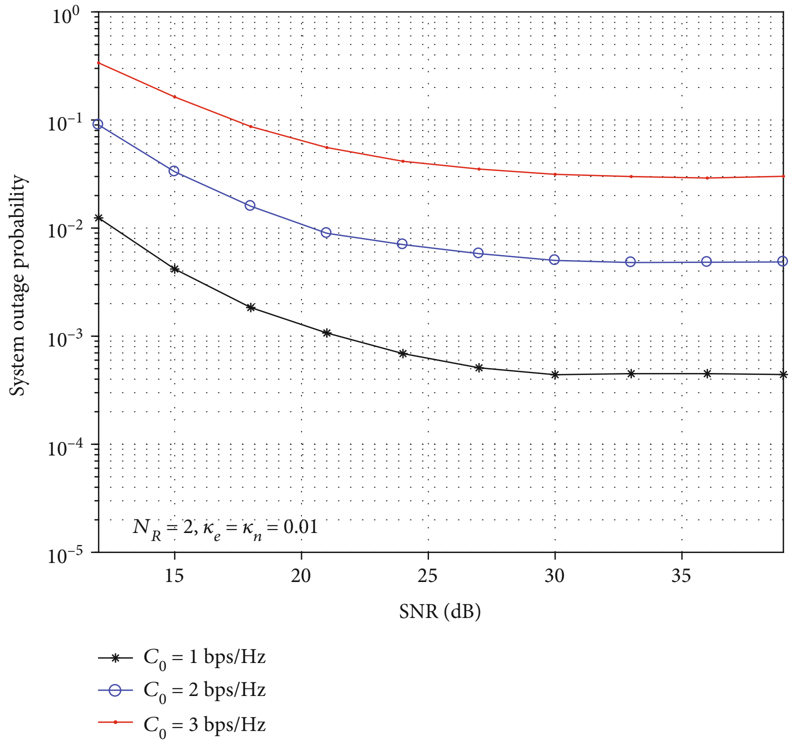

Figure 5: Outage probability versus capacity threshold for relaying links with $N_R = 2$.

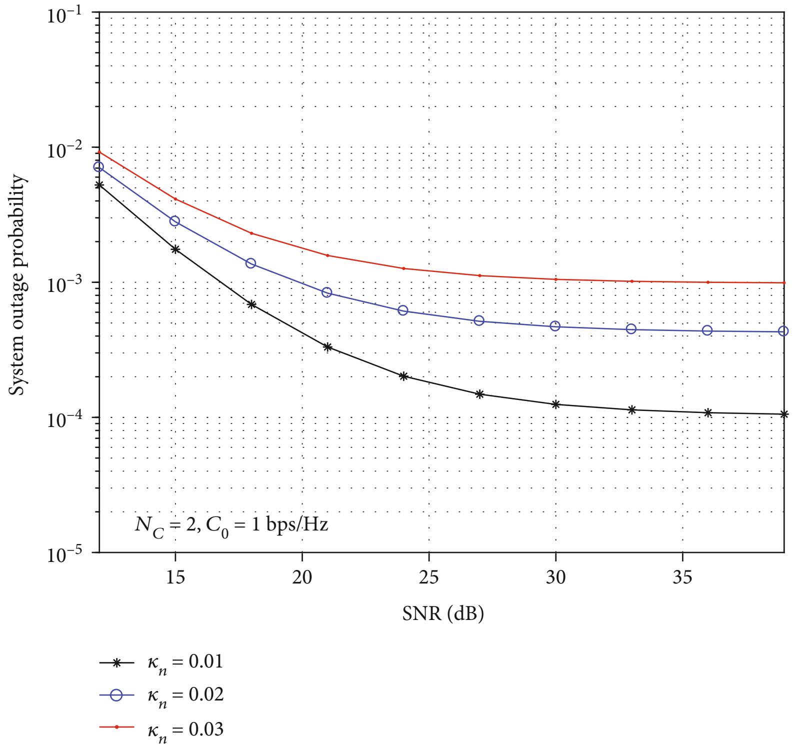

Figure 6: Outage probability versus κ_η for cached links with $N_C = 2$.

impairments. When the hardware impairment factor κ_η grows larger, the noise power will be higher, which will deteriorate the system performance.

The effect of the number of cached UAV nodes is present in Figure 8, where the system parameters are set as follows: $\kappa_e = \kappa_\eta = 0.01$, $C_0 = 1$ bps/Hz, and the number of cached

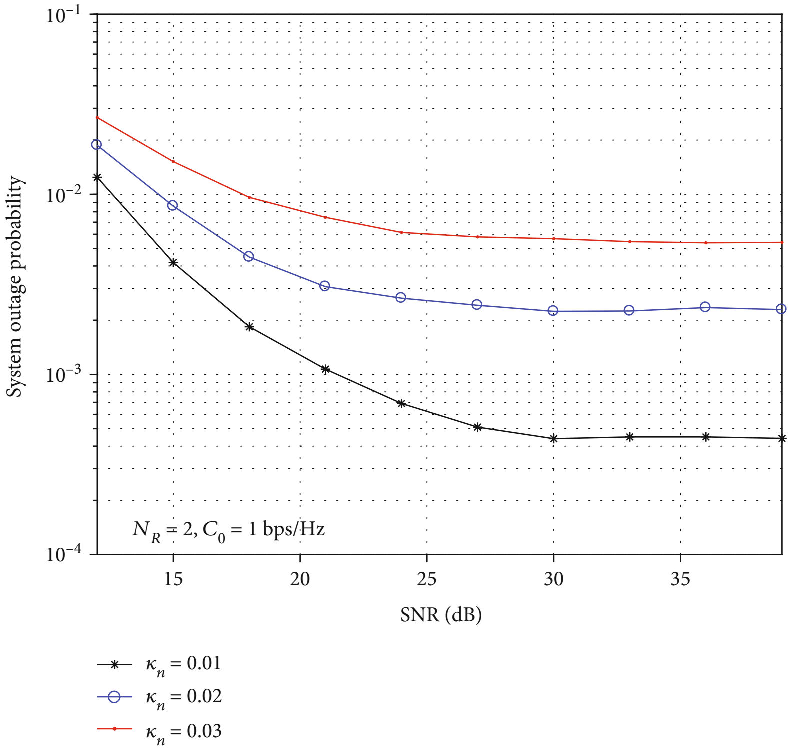

Figure 7: Outage probability versus κ_η for relaying links with $N_R = 2$.

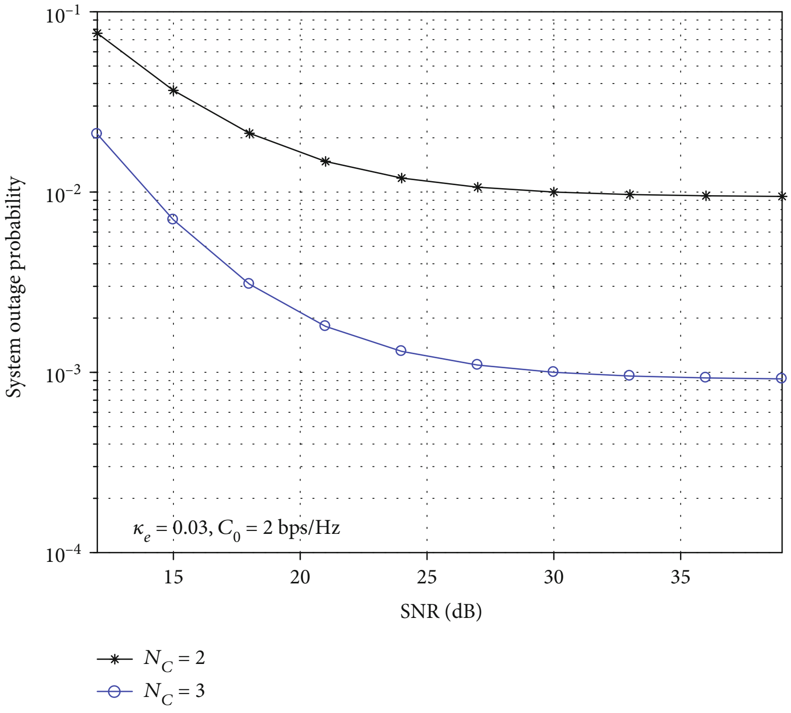

Figure 8: Outage probability versus N_C for cached links.

UAV nodes N_C changed from 2 to 3. We can see from this figure that N_C shows great impact on system performance. As mentioned by the remarks of asymptotic analysis, the diversity order of error floor for the cached/relaying links is equal to the number of cached/relaying UAV nodes.

6. Conclusions

The impacts of hardware impairments with imperfect channel estimation for cache-enabled UAV relaying networks are investigated in this paper. To improve the QoS of the wireless channels, the best cached UAV nodes or AF relaying nodes are selected to help the message transmission from the macro BS to the user. The impacts of the setup parameters, such as the number of cached/relaying nodes, the hardware impairment factor, the channel estimation error, and the transmission SNR, on the outage probability are present by deriving the exact closed-form expressions on outage probability. Also, the asymptotic results are present when the transmission SNR is large enough. From the asymptotic analysis, we can see that there is an error floor introduced by the hardware impairments and imperfect channel estimation. Specifically, the error floor is dependent on the channel estimation error factor, the hardware impairment distortion factor, and the QoS requirements. In future works, we will introduce deep learning-based [38–40] or Q-learning-based algorithms [41, 42] to further improve the system performance. Moreover, we will apply the considered wireless techniques into some practical IoT systems [43, 44] to achieve the green energy applications.

Appendix

Proof of Theorem 1

For the case $N_R = 1$, the theorem obviously holds.

We will focus on the case that $N_R \geq 2$. Firstly, we give the following definition

$$z_i = \min\left(\rho_{u,i}, \xi\rho_{r,i}\right). \tag{A.1}$$

Then, the PDF of z_i can be derived as

$$\begin{aligned} F_{zi}(x) &= Pr\left\{\min\left(\rho_{u,i}, \xi\rho_{r,i}\right) < z\right\} \\ &= 1 - Pr\left\{\min\left(\rho_{u,i}, \xi\rho_{r,i}\right) \geq z\right\} \\ &= 1 - Pr\left\{\rho_{u,i} \geq z, \rho_{r,i} \geq \frac{z}{\xi}\right\}. \end{aligned} \tag{A.2}$$

Due to the independence between $\rho_{u,i}$ and $\rho_{r,i}$, we obtain

$$F_{zi}(x) = 1 - \Pr\left\{\rho_{u,i} \geq z\right\}\Pr\left\{\rho_{r,i} \geq \frac{z}{\xi}\right\} = 1 - \frac{e^{-z}e^{-z}}{\xi = 1 - e^{-(1+1/\xi)z}}. \tag{A.3}$$

Taking the derivative of $F_{z_i}(x)$, we can get the PDF of z_i as

$$f_{zi}(x) = \xi_1 e^{-\xi_1 x}, \tag{A.4}$$

where

$$\xi_1 = 1 + \frac{1}{\xi}. \tag{A.5}$$

Secondly, we give another definition as

$$\theta_m = \max_{i\in\Omega_R/m} z_i. \tag{A.6}$$

Since z_i's are independent identical distribution, we can have

$$F_{\theta_m}(x) = \left[F_{z_i}(x)\right]^{N_R-1}. \tag{A.7}$$

By substituting (A.3) into (A.7) and applying the binomial theorem, we have

$$\begin{aligned} F_{\theta_m}(x) &= \left(1 - e^{-\xi_1 z}\right)^{N_R-1} \\ &= \sum_{k=0}^{N_R-1} C_{N_R-1}^{k}(-1)^k e^{-k\xi_1 z} \\ &= 1 - \sum_{k=1}^{N_R-1} C_{N_R-1}^{k}(-1)^{k-1} e^{-k\xi_1 z}, \quad N_R \geq 2. \end{aligned} \tag{A.8}$$

As such, the PDF of θ_m can be obtained as

$$f_{\theta_m}(x) = \sum_{k=1}^{N_R-1} b_k e^{-k\xi_1 x}, \tag{A.9}$$

where $b_k = C_{N_R-1}^{k}(-1)^{k-1}k\xi_1$.

According to the definition in (17), we can derive the CDF of ρ_{u,n_2^*},

$$\begin{aligned} F_{\rho_{u,n_2^*}}(x) &= \sum_{m=1}^{N_R} Pr\left[\rho_{u,m} < x, z_m > \theta_m\right] \\ &= N_R Pr\left[\rho_{u,1} < x, z_1 > \theta_1\right] \\ &= N_R Pr\left[\rho_{u,1} < x, \rho_{u,1} > \theta_1, \xi\rho_{r,1} > \theta_1\right] \\ &= N_R Pr\left[\theta_1 < \rho_{u,1} < x, \rho_{r,1} > \frac{\theta_1}{\xi}, \theta_1 < x\right]. \end{aligned} \tag{A.10}$$

By using the result in (A.9), we have

$$\begin{aligned} F_{\rho_{u,n_2^*}}(x) &= N_R Pr\left[\theta_1 < \rho_{u,1} < x, \rho_{r,1} > \frac{\theta_1}{\xi}, \theta_1 < x\right] \\ &= N_R \int_0^x f_{\theta_1}(\theta)\left(e^{-\theta} - e^{-x}\right)e^{-\theta/\xi}d\theta \\ &= N_R \sum_{k=1}^{N_R-1} b_k\left[\frac{1}{\xi_2} - \frac{e^{-x}}{\xi_3} + \frac{e^{-\xi_2 x}}{\xi_3\xi_2}\right], \end{aligned} \tag{A.11}$$

where

$$\begin{cases} \xi_2 = (k+1)\xi_1, \\ \xi_3 = k\xi_1 + \dfrac{1}{\xi} = \xi_2 - 1, \\ \xi_4 = k\xi_1 + 1 = \xi_2 - \dfrac{1}{\xi}. \end{cases} \tag{A.12}$$

Similarly, we can obtain the CDF of ρ_{r,n_2^*} as follows:

$$F_{\rho_{r,n_2^*}}(x) = N_R \sum_{k=1}^{N_R-1} b_k \left[\frac{1}{\xi_2} - \frac{e^{-x}}{\xi_4} + \frac{e^{-\xi_2 \xi x}}{\xi \xi_2 \xi_4} \right]. \tag{A.13}$$

Thus, Theorem 1 is proved.

Data Availability

The data of this work can be available through the request on the corresponding author by e-mail.

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Acknowledgments

This work was partly supported by the Natural Science Foundation of Guangdong Province with grant number 2018A030313736, the Scientific Research Project of Education Department of Guangdong with grant numbers 2017GKTSCX045 and 2019GZDXM002, the Project of Technology Development Foundation of Guangdong with grant number 706049150203, the Science and Technology Program of Guangzhou under grant 201807010103, the Application Technology Collaborative Innovation Center of GZPYP with grant number 2020ZX01, the Yangcheng Scholar, Scientific Research Project of Guangzhou Education Bureau with grant number 202032761, the Department of Education of Guangdong Province of China with grant number 2017GKTSCX047, the Education Department of Guangzhou City of China with grant number 201831785, the Personnel Training Project of Guangzhou University with grant number RP2020122, and the Yangcheng Scholars Innovative Academic Team Project with grant number 1201610010.

References

[1] J. Xia, C. Li, X. Lai et al., "Cache-aided mobile edge computing for B_5G wireless communication networks," *EURASIP Journal on Wireless Communications and Networking*, vol. 2020, no. 1, 2020.

[2] C. Li, Z. Gao, J. Xia, D. Deng, and L. Fan, "Cache-enabled physical-layer secure game against smart uavassisted attacks in b_5g NOMA networks," *EURASIP Journal on Wireless Communications and Networking*, vol. 2020, no. 1, 2020.

[3] J. Zhao, X. Guan, and X. P. Li, "Power allocation based on genetic simulated annealing algorithm in cognitive radio networks," *Chinese Journal of Electronics*, vol. 22, no. 1, pp. 177–180, 2013.

[4] Y. Guo, Z. Zhao, R. Zhao et al., "Intelligent offloading strategy design for relaying mobile edge computing networks," *IEEE Access*, vol. 8, pp. 35127–35135, 2020.

[5] D. Deng, J. Xia, L. Fan, and X. Li, "Link selection in buffer-aided cooperative networks for green iot," *IEEE Access*, vol. 8, pp. 30763–30771, 2020.

[6] J. Zhao, S. Ni, L. Yang, Z. Zhang, Y. Gong, and X. You, "Multiband cooperation for 5G HetNets: a promising network paradigm," *IEEE Vehicular Technology Magazine*, vol. 14, no. 4, pp. 85–93, 2019.

[7] J. Zhu, C. Gong, S. Zhang, M. Zhao, and W. Zhou, "Foundation study on wireless big data: concept, mining, learning and practices," *China Ccommunications*, vol. 15, no. 12, pp. 1–15, 2018.

[8] D. Deng and J. Xia, "Cache-enabled cooperative edge networks for intelligent connected vehicles," *IEEE Access*, vol. 7, pp. 166939–166949, 2019.

[9] Z. Ming, S. Zhou, W. Zhou, and J. Zhu, "An improved uplink sparse coded multiple access," *IEEE Communications Letters*, vol. 21, no. 1, pp. 176–179, 2017.

[10] Y. Wang, M. Zhao, D. Deng, S. Zhou, and W. Zhou, "Fractional sparse code multiple access and its optimization," *IEEE Wireless Communications Letters*, vol. 7, no. 6, pp. 990–993, 2018.

[11] J. Song, H. Song, and W. Choi, "Which one is better to cache: requested contents or interfering contents?," *IEEE Wireless Communications Letters*, vol. 8, no. 3, pp. 861–864, 2019.

[12] D. Ko, B. Hong, and W. Choi, "Probabilistic caching based on maximum distance separable code in a user-centric clustered cache-aided wireless network," *IEEE Transactions on Wireless Communications*, vol. 18, no. 3, pp. 1792–1804, 2019.

[13] X. Li, J. Li, Y. Liu, Z. Ding, and A. Nallanathan, "Residual transceiver hardware impairments on cooperative NOMA networks," *IEEE Transactions on Wireless Communications*, vol. 19, no. 1, pp. 680–695, 2020.

[14] O. T. Demir and E. Bjornson, "Channel estimation in massive MIMO under hardware non-linearities: Bayesian methods versus deep learning," *IEEE Open Journal of the Communications Society*, vol. 1, pp. 109–124, 2020.

[15] J. Zhu, D. W. K. Ng, N. Wang, R. Schober, and V. K. Bhargava, "Analysis and design of secure massive MIMO systems in the presence of hardware impairments," *IEEE Transactions on Wireless Communications*, vol. 16, no. 3, pp. 2001–2016, 2017.

[16] A. K. Papazafeiropoulos, S. K. Sharma, S. Chatzinotas, and B. Ottersten, "Ergodic capacity analysis of AF DH MIMO relay systems with residual transceiver hardware impairments: conventional and large system limits," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 8, pp. 7010–7025, 2017.

[17] E. Björnson, J. Hoydis, M. Kountouris, and M. Debbah, "Massive MIMO systems with non-ideal hardware: energy efficiency, estimation, and capacity limits," *IEEE Transactions on Information Theory*, vol. 60, no. 11, pp. 7112–7139, 2014.

[18] G. Ding, X. Gao, Z. Xue, Y. Wu, and Q. Shi, "Massive mimo for distributed detection with transceiver impairments," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 1, pp. 604–617, 2018.

[19] S. Guo and X. Zhou, "Robust resource allocation with imperfect channel estimation in NOMA-based heterogeneous vehicular networks," *IEEE Transactions on Communications*, vol. 67, no. 3, pp. 2321–2332, 2019.

[20] W. A. Al-Hussaibi and F. H. Ali, "A closed-form approximation of correlated multiuser mimo ergodic capacity with antenna selection and imperfect channel estimation," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 6, pp. 5515–5519, 2018.

[21] Y. M. Khattabi and M. M. Matalgah, "Alamouti-OSTBC wireless cooperative networks with mobile nodes and imperfect CSI estimation," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 4, pp. 3447–3456, 2018.

[22] Z. Zhao, W. Zhou, D. Deng, J. Xia, and L. Fan, "Intelligent mobile edge computing with pricing in internet of things," *IEEE Access*, vol. 8, pp. 37727–37735, 2020.

[23] L. Fan, J. Xia, N. Yang, T. Q. Duong, G. Karagiannidis, and A. Nallanathan, "Opportunistic access point selection for mobile edge computing networks," *IEEE Transactions on Wireless Communications*, no. 99, 2020.

[24] Z. Junhui, Y. Tao, G. Yi, W. Jiao, and F. Lei, "Power control algorithm of cognitive radio based on non-cooperative game theory," *China Communications*, vol. 10, no. 11, pp. 143–154, 2013.

[25] N. Zhao, X. Liu, F. R. Yu, M. Li, and V. C. M. Leung, "Communications, caching, and computing oriented small cell networks with interference alignment," *IEEE Communications Magazine*, vol. 54, no. 9, pp. 29–35, 2016.

[26] Z. Na, J. Wang, C. Liu, M. Guan, and Z. Gao, "Join trajectory optimization and communication design for UAV-enabled OFDM networks," *Ad Hoc Networks*, vol. 98, article 102031, 2020.

[27] Z. Na, J. Lv, F. Jiang, M. Xiong, and N. Zhao, "Joint subcarrier and subsymbol allocation-based simultaneous wireless information and power transfer for multiuser GFDM in iot," *IEEE Internet of Things Journal*, vol. 6, no. 4, pp. 5999–6006, 2019.

[28] Z. Na, J. Lv, M. Zhang, B. Peng, M. Xiong, and M. Guan, "GFDM based wireless powered communication for cooperative relay system," *IEEE Access*, vol. 7, pp. 50971–50979, 2019.

[29] Z. Na, Y. Wang, X. Li et al., "Subcarrier allocation based simultaneous wireless information and power transfer algorithm in 5G cooperative OFDM communication systems," *Physical Communication*, vol. 29, pp. 164–170, 2018.

[30] S. Lai, J. Xia, D. Zou, and L. Fan, "Intelligent secure communication for cognitive networks with multiple primary transmit power," *IEEE Access*, vol. 8, pp. 37343–37351, 2020.

[31] R. Zhao, "Deep reinforcement learning based mobile edge computing for intelligent internet of things," *IEEE Access*, no. 99, 2020.

[32] J. Xia, D. Deng, and D. Fan, "A note on implementation methodologies of deep learning-based signal detection for conventional MIMO transmitters," *IEEE Transactions on Broadcasting*, no. 99, pp. 1-2, 2020.

[33] K. He, "Ultra-reliable MU-MIMO detector based on deep learning for 5G/B5G-enabled IoT," *EURASIP Journal on Wireless Communications and Networking*, no. 99, 8 pages, 2020.

[34] J. Liao, J. Zhao, F. Gao, and G. Y. Li, "A model-driven deep learning method for massive MIMO detection," *IEEE Communications Letters*, no. 99, 2020.

[35] N. Zhao, Y. Cao, F. R. Yu, Y. Chen, M. Jin, and V. C. M. Leung, "Artificial noise assisted secure interference networks with wireless power transfer," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 2, pp. 1087–1098, 2018.

[36] D. Deng, L. Fan, R. Zhao, and R. Q. Hu, "Secure communications in multiple amplify-and-forward relay networks with outdated channel state information," *Transactions on Emerging Telecommunications Technologies*, vol. 27, no. 4, pp. 494–503, 2016.

[37] I. S. Gradshteyn and I. M. Ryzhik, *Table of Integrals, Series, and Products*, Academic Press, Elsevier Inc, San Diego, CA, USA, 7th edition, 2007.

[38] J. Xia, K. He, W. Xu, S. Zhang, L. Fan, and G. K. Karagiannidis, "A MIMO detector with deep learning in the presence of correlated interference," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 4, pp. 4492–4497, 2020.

[39] G. Liu, Y. Xu, Z. He, Y. Rao, J. Xia, and L. Fan, "Deep learning-based channel prediction for edge computing networks toward intelligent connected vehicles," *IEEE Access*, vol. 7, pp. 114487–114495, 2019.

[40] K. He, Z. Wang, W. Huang, D. Deng, J. Xia, and L. Fan, "Generic deep learning-based linear detectors for MIMO systems over correlated noise environments," *IEEE Access*, vol. 8, pp. 29922–29929, 2020.

[41] Z. Zhao, R. Zhao, J. Xia et al., "A novel framework of three-hierarchical offloading optimization for MEC in industrial IoT networks," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 8, pp. 5424–5434, 2020.

[42] J. Xia, Y. Xu, D. Deng, Q. Zhou, and L. Fan, "Intelligent secure communication for internet of things with statistical channel state information of attacker," *IEEE Access*, vol. 7, pp. 144481–144488, 2019.

[43] C. Fan and Y. Ding, "Cooling load prediction and optimal operation of HVAC systems using a multiple nonlinear regression model," *Energy and Buildings*, vol. 197, pp. 7–17, 2019.

[44] C. Fan, Y. Ding, and Y. Liao, "Analysis of hourly cooling load prediction accuracy with data-mining approaches on different training time scales," *Sustainable Cities and Society*, vol. 51, article 101717, 2019.