Cost-Efficient Cooperative Video Caching Over Edge Networks

Bingjie Zhu, Liqiang Zhao, *Member, IEEE*, Wenqiang Yi, *Member, IEEE*, Zhixiong Chen, *Graduate Student Member, IEEE*, and Arumugam Nallanathan, *Fellow, IEEE*

***Abstract*—Cooperative caching has emerged as an efficient way to alleviate backhaul traffic and enhance user experience by proactively prefetching popular videos at the network edge. However, it is challenging to achieve the optimal design of video caching, sharing, and delivery within storage-limited edge networks due to the growing diversity of videos, unpredictable video requirements, and dynamic user preferences. To address this challenge, this work explores cost-efficient cooperative video caching via video compression techniques while considering unknown video popularity. First, we formulate the joint video caching, sharing, and delivery problem to capture a balance between user delay and system operative cost under unknown time-varying video popularity. To solve this problem, we develop a two-layer decentralized reinforcement learning algorithm, which effectively reduces the action space and tackles the coupling among video caching, sharing, and delivery decisions compared to the conventional algorithms. Specifically, the outer layer produces the optimal decisions for video caching and communication resource allocation by employing a multiagent deep deterministic policy gradient algorithm. Meanwhile, the optimal video sharing and computation resource allocation are determined in each agent's inner layer using the alternating optimization algorithm. Numerical results show that the proposed algorithm outperforms benchmarks in terms of the cache hit rate, delay of users and system operative cost, and effectively strikes a tradeoff between system operative cost and users' delay.**

***Index Terms*—Cooperative video caching, multiagent reinforcement learning, performance-cost tradeoff.**

This work was supported in part by the Key Research and Development Program of Shaanxi under Grant 2022KWZ-09; in part by the Postdoctoral Fellowship Program of CPSF under Grant GZC20232058; in part by the Key-Area Research and Development Program of Guangdong Province under Grant 2020B0101120003; in part by the Fundamental Research Funds for the Central Universities under Grant ZYTS24110; in part by the Postdoctoral Research Program of Shaanxi Province under Grant 2023BSHYDZZ100; and in part by the National Key Research and Development Program of China under Grant 2020YFB1807700. *(Corresponding author: Liqiang Zhao.)*

Bingjie Zhu is with the State Key Laboratory of Integrated Service Networks, Xidian University, Xi'an 710071, China (e-mail: bjzhu1@stu.xidian.edu.cn).

Liqiang Zhao is with the State Key Laboratory of Integrated Service Networks, Xidian University, Xi'an 710071, China, and also with the Guangzhou Institute of Technology, Xidian University, Guangzhou 510100, China (e-mail: lqzhao@mail.xidian.edu.cn).

Wenqiang Yi is with the School of Computer Science and Electronic Engineering, University of Essex, CO4 3SQ Colchester, U.K. (e-mail: wy23627@essex.ac.uk).

Zhixiong Chen and Arumugam Nallanathan are with the School of Electronic Engineering and Computer Science, Queen Mary University of London, E1 4NS London, U.K. (e-mail: zhixiong.chen@qmul.ac.uk;

I. Introduction

With the advancements in communication technologies, the proliferation of mobile terminals and diversified applications have led to the explosive growth of mobile data traffic [1]. According to the forecast report of Ericsson, the video traffic (e.g., short online videos, high-definition films, and live streaming), will account for 79 percent of the mobile network traffic that will reach 370 exabytes per month in 2027 [2].

Edge caching arises as a promising technology to address the challenge of mobile data traffic [3]. Specifically, edge caching enables the edge servers to store popular videos, thereby reducing latency and alleviating the transmission pressure [4], [5]. However, the capacity of the edge servers may not meet the requirement of data storage due to the rapid surge in data traffic. Fortunately, the introduction of cooperative caching has enabled the edge to accommodate a larger number of videos. This is attributed to its capability to facilitate collaboration among multiple edge servers, enabling them to share the cached videos [6], [7], [8]. Concurrently, video compression serves as another effective method to mitigate the limitations of storage space at edge servers. By compressing videos, edge servers can cache the smaller sized videos along with their corresponding transcoding parameters, thereby consuming less storage space compared to storing the original videos [9], [10]. Therefore, the combination of cooperative caching and video compression allows the edge to store a greater variety of videos within the limited storage capacities of edge servers.

Additionally, cooperative video sharing and video delivery are two pivotal strategies within the cooperative video caching framework, influencing the Quality of Experience (QoE) for users. On the one hand, along with the cooperative video caching strategy, video sharing strategies enable edge to provide users with more video services by video migration among edge servers [11]. On the other hand, video delivery strategies can directly impact the transmission delay that edge servers send the requested videos to users, thereby influencing the delay which users acquire videos [12]. Therefore, it is essential to jointly optimize the strategies of cooperative caching, sharing and delivery. Ren et al. [13], Zhang et al. [14], and Kuo et al. [15] investigated the joint optimization of the three above strategies in a cooperative caching system, aiming to minimize the delay of users [13], [14] or maximize user satisfaction [15]. However,

the operational cost was not considered in [13], [14], and [15], which is important for mobile network operator (MNO) [16]. MNO expects the development of cost-efficient cooperative caching systems, aiming to enhance users' QoE while minimizing operational costs, in response to practical constraints and cost considerations [17]. It is worth mentioning that there is an inherent conflict between high performance and low-operative cost in the cooperative caching system [18]. Therefore, achieving a favorable performance-cost tradeoff is essential when considering cooperative video sharing and video delivery in this system.

The integration of cooperative video caching, sharing, and delivery offers significant benefits, but achieving joint optimization of these elements is a complex challenge. This complexity is primarily due to the varying popularity of videos across different edge servers over time and space, influenced by the diverse and changing preferences of individual users. The spatio-temporal variations in video popularity are unknown, which poses significant challenges for traditional optimization methods like the greedy algorithm [19], [20], convex optimization [21], and Lyapunov optimization [22], [23]. These methods typically rely on the assumption that user video preferences are either known or can be accurately predicted to solve the joint optimization problem in cooperative caching systems, which is impractical for real-world scenarios.

Deep reinforcement learning (DRL) emerges as a promising solution to overcome the limitations of traditional optimization methods [24], [25], [26]. To be specific, the DRL algorithms do not rely on the assumption because the agent can directly interact with the environment, acquiring the ability to generate a sequence of actions that adapt to the spatio-temporal variations in video popularity [27], [28]. Existing works focused on utilizing the DRL algorithms to optimize service caching or content caching. Ren et al. [13] developed a proximal policy optimization algorithm to explore request dynamics, thereby enabling the joint optimization of service caching and request scheduling in the multiaccess edge computing-assisted networks. Luo et al. [29] proposed a Q-learning algorithm to directly learn content placement instead of predicting content popularity in cache-enabled networks.

However, the considered joint optimization problem in cooperative video caching system is still nontrivial even resorting to the powerful DRL algorithm. Specifically, the cooperative video caching system, when factoring in video sharing and delivery, involves numerous optimization variables. These variables encompass decisions related to video caching, sharing, and resource allocations during the video delivery phase, resulting in an extensive action space for DRL. One reason for the huge action space is attributed to the fact that the dimension of actions in the DRL algorithm exponentially increases with the number of optimization variables. Directly employing the DRL algorithm to address the optimization problem with an extensive action space requirement may slow down the convergence rate or even result in the failure of the DRL algorithm's convergence, which is called the curse of dimensionality [30], [31]. Hence, a tailored DRL-based algorithm that can reduce action space is required for the joint optimization problem of video caching, sharing and delivery.

To this end, we investigate the distributed cooperative caching over the edge networks from the performance-cost tradeoff perspective in this article. This is realized through the joint optimization of video caching, sharing and delivery while considering unknown time-varying video popularity and limited storage capacity. To address this joint optimization problem, we propose an innovative two-layer DRL algorithm based on alternative optimization and the multiagent deep deterministic policy gradient-based (MADDPG) algorithm. The simulation results prove the feasibility of the proposed scheme, which strikes the tradeoff between the operative cost and performance. The main contributions of this article are summarized as follows:

1) We consider the video sharing, video delivery and video compression in the distributed cooperative caching system with unknown video popularity and heterogeneous user needs. To investigate the performance-cost tradeoff in the cache-enabled edge network, we analyze and derive user delay and system operative cost.
2) We formulate the joint optimization problem of cooperative video caching, sharing and delivery to minimize the user delay and operative cost while satisfying users' delay requirements. The joint optimization involves five decision variables, including two variables, related to video caching and sharing as well as three resource allocation variables relevant to the video delivery, i.e., computation resource, subcarrier, and power allocation. These variables are tightly coupled in the objective function and constraints, which makes the problem difficult to solve.
3) To solve the problem with numerous optimization variables, we first decouple the joint optimization problem into two subproblems: a) joint optimization of video sharing and computation resource allocation and b) joint optimization of video caching and communication resource allocation. To deal with the two subproblems, we propose a two-layer DRL algorithm. Specifically, the outer layer of the proposed algorithm makes the video caching and communication resource allocation decision via the MADDPG algorithm, where each edge base station (BS) is regarded as a learning agent. Then, the inner layer of each agent generates a joint decision of video sharing and computation resource allocation via an alternating optimization algorithm, thereby effectively reducing the action space.
4) The simulation results show that our proposed scheme outperforms other benchmark methods in terms of average cache hit rate, the delay of users, and system operative cost, which indicates that the proposed scheme can better adapt to spatial-temporal variations of video popularity and time-varying wireless channel quality. In addition, under different tradeoff factor settings, the proposed scheme reduces over 11% user delay and 14.5% system operative cost compared to benchmark schemes, which verifies that the proposed scheme achieves a compelling

TABLE I
NOTATION SUMMARY

Notation	Definition
$\mathcal{B}$, B	Set of physical BSs; size of $\mathcal{B}$
$\mathcal{K}$, K, $\mathcal{K}_b$	Set of users; size of $\mathcal{K}$, set of users associated with physical BS b
$\mathcal{V}$, V	Set of videos; size of $\mathcal{V}$
$\mathcal{S}$, S	Set of sub-carriers; size of $\mathcal{S}$
i, z_{v_i}, ρ	Version of videos and index of virtual edge; the size of video v_i; video compressed ratio
τ, T	The duration of each time slot; total number of time slots
$q^b_{k,v}(t)$	Video request indicator of user k
$D^{b,l}_{v_i}(t)$	Transmission delay of video v_i from network entity l to physical BS b
$D^{b,re}_{v}(t)$	Retrieval delay of video v in physical BS b
$D^{b,com}_{k,v}(t)$	Computation delay of video v for user k
$D^{b,ac}_{k,v}(t)$	Transmission delay of user k over access link
$D^b_k(t)$	Total delay of user k
$E^b_{re}(t)$, $E^b_{com}(t)$, $E^b_{ac}(t)$, $E^b_{ca}(t)$, $E^b(t)$	Retrieval cost; computation cost; caching cost; transmission cost over access link; total cost of physical BS b
$D(t)$, $E(t)$	Total delay of all users; total system operative cost
ζ, ξ	Refractive index of fiber; speed of light in the vacuum
$d_{b,l}$, $\eta_{b,l}$	Total optic fiber length; hop counts along the shortest transmission path from physical BS b to network entity l
R_o, R_b	Transmission rate of backhaul link; transmission rate of wired links among physical BSs
$\lambda^b_{k,v}$	Computing density required by user k to transcode the compressed video v_2
B_w	Total bandwidth of the downlink
$\gamma^b_{k,s}(t)$, $r^b_{k,s}(t)$, $R^b_k(t)$	SINR and achievable data rate of user k on sub-carrier s; transmission rate of user k over the access link
$w^i_{m,l}$, w_f, w_{bw}, w_p, w_c	Unit prices of video migration, computation resource, bandwidth, power and video updating
ε, δ	Normalized factor and trade-off factor between user delay and system operative cost
$r_p(t)$, o_p	Penalty function and penalty coefficient
$y^{b,l}_{v_i}(t)$, $\mathbf{Y}^b_t$, $\mathbf{Y}$	Indicator of video sharing decisions; set of $y^{b,l}_{v_i}(t)$; set of $\mathbf{Y}^b_t$
$c^b_{v_i}(t)$, $\mathbf{C}^b_t$, $\mathbf{C}$	Indicator of video caching decisions; set of $c^b_{v_i}(t)$; set of $\mathbf{C}^b_t$;
$f^b_k(t)$, $\mathbf{F}^b_t$, $\mathbf{F}$	Variables of computation resource allocation, set of $f^b_k(t)$; set of $\mathbf{F}^b_t$
$x^b_{k,s}(t)$, $\mathbf{X}^b_t$, $\mathbf{X}$	Indicator of sub-carrier allocation; set of $x^b_{k,s}(t)$; set of $\mathbf{X}^b_t$
$p^b_{k,s}(t)$, $\mathbf{P}^b_t$, $\mathbf{P}$	Transmit powers allocation variables; set of $p^b_{k,s}(t)$; set of $\mathbf{P}^b_t$

tradeoff between the system operative cost and user delay.

The remainder of this article is organized as follows: Section II presents the main components of the cooperative caching model and the formulated optimization problem minimizing delay and cost. In Section III, we propose a decentralized two-layer DRL algorithm to solve the large-scale mixed integer optimization problem. Simulation results are presented in Section IV to evaluate the performances of our proposed algorithm. Section V concludes this article. The main notations of this paper are summarized in Table I.

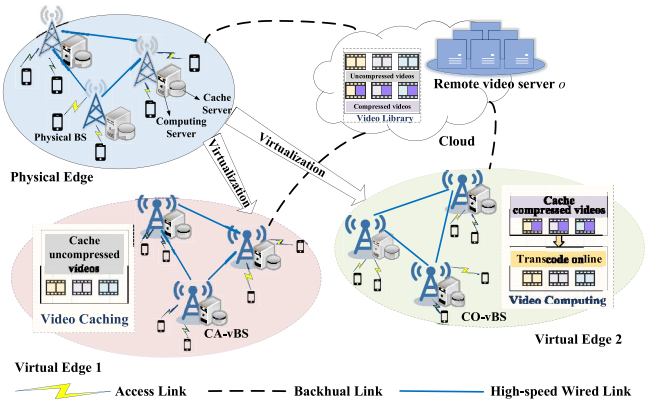


Fig. 1. System model of cooperative caching.

II. SYSTEM MODEL

A. Network Model

As shown in Fig. 1, B cache-enabled physical BSs connected via fibers provide video services to K users. The physical BSs and users are indexed by $\mathcal{B} = \{1, \ldots, b, \ldots, B\}$ and $\mathcal{K} = \{1, \ldots, k, \ldots, K\}$, respectively. For physical BS b, its associated users are denoted by $\mathcal{K}_b \subset \mathcal{K}$. It is assumed each physical BS has the same caching capacity of z bits and limited computing capacity of F cycle/s. The remote video server communicates with all physical BSs through a backhaul network.

By employing the network function virtualization (NFV), each physical BS can be virtualized as one caching virtual BS (CA-vBS) and one computation virtual BS (CO-vBS). Naturally, the physical edge, formed by all physical BSs, is divided into two virtual edges, namely, virtual edge 1 comprising CA-vBSs and virtual edge 2, consisting of CO-vBSs. In virtual edge 1, each CA-vBS is endowed with large storage space to cache uncompressed videos. Meanwhile, each CO-vBS in virtual edge 2 has small storage space but sufficient computing resources for caching and transcoding compressed videos. Suppose that there are V videos in the remote video server o, indexed by $\mathcal{V} = \{1, \ldots, v, \ldots, V\}$. Each video v has both uncompressed version v_1 and compressed version v_2, whose sizes are z_{v_1} and z_{v_2}, respectively. Moreover, $z_{v_2} = \rho z_{v_1}$ and $0 < \rho < 1$ is video compressed ratio. For the convenience of representation, let $i \in \{1, 2\}$ denote not only the version of videos but also the index of the virtual edge.

B. Video Request Strategy

The time dimension is divided into time slots of duration τ, indexed by $t \in \{1, 2, \ldots, T\}$. Assuming that each user sends its video request at the beginning of each time slot. Let the binary variable $q^b_{k,v}(t)$ indicate the request of user k associated with physical BS b for video v at time slot t. Specifically, if user k requests video v at time slot t, $q^b_{k,v}(t) = 1$; otherwise, $q^b_{k,v}(t) = 0$. In this cooperative caching system, we assume that each user request must be satisfied within the time slot requested by the user. This assumption can be eliminated by setting that the duration of each time slot τ exceeds the maximum delay constraint for all users [32]. Fig. 2 shows

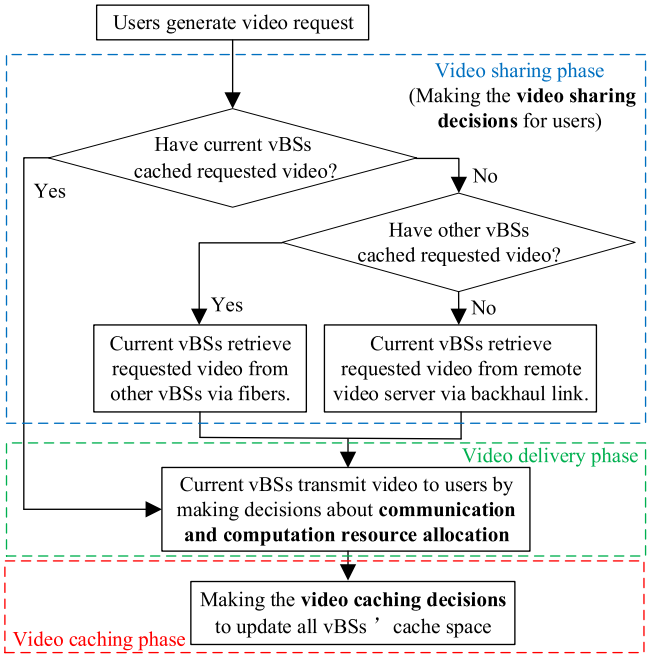


Fig. 2. Operation mechanism in one time slot.

the operation mechanism that the cooperative caching system provide users with requested videos in one time slot. The operation mechanism comprises a video sharing phase, a video delivery phase, and a video caching phase [29]:

1) *Video Sharing Phase:* After receiving a user's video request, the physical BS associated with the user first checks whether its corresponding CA-vBS and CO-vBS, i.e., current vBSs, have cached the requested video. If the current vBSs have cached the requested video, current vBSs deliver the requested video to users. When the requested video has been not cached by current vBSs but is available in other vBSs, other vBSs share the compressed or uncompressed requested video with current vBSs via the high-speed wired link. In the worst-case, none of the vBSs at the edge have cached the requested video. In such instances, the remote video server transmits the uncompressed requested videos to the current CA-vBS through the backhaul link. Consequently, the requested video can be fetched from the network entities, including current vBSs, other vBSs, and remote video server o. We denote the video sharing decision of cooperative edge caching system as $\mathbf{Y} = \{y_{v_i}^{b,l}(t) | b \in \mathcal{B}, t \in T, v \in \mathcal{V}, l \in \mathcal{B} \cup o, i \in \{1,2\}\}$, where $y_{v_i}^{b,l}(t) \in \{0,1\}$ is a binary variable to represent whether the current vBSs of physical BS b retrieve the video v_i from the network entity l or not.
2) *Video Delivery Phase:* Upon fetching the requested video shared by network entities, the current vBSs provide the requested video services to users over the access link by making the decisions on communication and computation resource allocation. If the current CA-vBS has fetched the requested video without compression, the current CA-vBS delivers the required video directly to the user over the access link. Alternatively, if the current CO-vBS has fetched the compressed version of the requested video, it conducts transcoding procedures before transmitting the required video to the user. Note that choosing which current vBS to finish video sharing and delivery process depends on current cache status, system operative cost and user delay, which will be introduced in detail later.
3) *Video Caching Phase:* At the end of each time slot, each vBS refreshes cached uncompressed or compressed videos from the cloud server under the video caching decision. In each time slot, the video caching decision determines which videos should have been cached, i.e., cache status, during the next time slot. The caching decision of the cooperative edge caching system is denoted as $\mathbf{C} = \{\mathbf{C}_t^b | b \in \mathcal{B}, t \in T\}$, where $\mathbf{C}_t^b = \{c_{v_i}^b(t) | v \in \mathcal{V}, i \in \{1,2\}\}$ is the caching decision variable of physical BS b at time slot t. $c_{v_i}^b(t) \in \{0,1\}$ is a binary variable to show whether physical BS b caches the corresponding video v_i at the end of time slot t or not.

C. Delay and Cost Model

We first analyze of the delay and cost within the video sharing and delivery phase, which can be divided into three components.

1) Retrieval Delay and Cost: In the video sharing phase, the retrieval delay of video v in physical BS b is given by

$$D_v^{b,re}(t) = \sum_{l \in \mathcal{B} \cup o} \sum_{i \in \{1,2\}} y_{v_i}^{b,l}(t) D_{v_i}^{b,l} \tag{1}$$

where $D_{v_i}^{b,l}$ is the transmission delay that network entity l delivers the requested video v_i to physical BS b. From physical BS b to network entity $l \in \mathcal{B} \cup o$, the total optic fiber length and hop counts along the shortest transmission path are denoted as $d_{b,l}$ and $\eta_{b,l}$, respectively. The transmission delay $D_{v_i}^{b,l}$ can be expressed by [13]

$$D_{v_i}^{b,l} = \begin{cases} \eta_{b,l} \frac{z_{v_i}}{R_b} + d_{b,l} \frac{\zeta}{\xi}, \text{if } \ l \in \mathcal{B} \\ \eta_{b,l} \frac{z_{v_i}}{R_o} + d_{b,l} \frac{\zeta}{\xi}, \text{if } \ l = o \end{cases} \tag{2}$$

where ζ are refractive index of fiber, ξ is speed of light in the vacuum, R_o is the transmission rate of backhaul link, and R_b is transmission rate of wired links among physical BSs.

The retrieval cost of each physical BS is defined as the migration cost of the requested videos from network entities to current vBSs [33]. Hence, the retrieval cost of physical BS b at time slot t can be given as follows:

$$E_{re}^b(t) = \sum_{v \in V} \sum_{i \in \{1,2\}} \sum_{l \in \mathcal{B} \cup o} w_{m,l}^i \eta_{b,l} y_{v_i}^{b,l}(t) \tag{3}$$

where $w_{m,l}^i$ is the unit migration price of transmitting videos between corresponding vBS in virtual edge i of physical BS b and network entity $l \in \mathcal{B} \cup o$. Note that $w_{m,o}^i > w_{m,b}^i \ \ \forall b \in \mathcal{B}, i \in \{1,2\}$. This is because the transmission overhead that the remote video server transmits the requested videos to vBSs is larger than the transmission overhead among vBSs. Besides, considering that the size of compressed videos is less than that of uncompressed videos, the unit migration price of

compressed video is smaller than uncompressed videos' unit migration price, that is, $w^1_{m,l} > w^2_{m,l}$.

2) Computation Delay and Cost: The computation delay and cost are introduced when the retrieval process is performed in virtual edge 2. The computation delay of user k who requests the video v at physical b is given by [34]

$$D^{b,\text{com}}_{k,v}(t) = \sum_{l\in\mathcal{B}\cup o} y^{b,l}_{v_2}(t)\frac{z_{v_2}\lambda^b_{k,v}}{f^b_k(t)} \tag{4}$$

where $\lambda^b_{k,v}$ is the computing density (in CPU cycle/bit) to transcode the compressed video v_2 requested by user k associated with physical BS b, $f^b_k(t)$ is computation resource of user k, which is allocated by the corresponding CO-vBS of physical BS b at time slot t. The computation cost of physical BS b depends on the amount of computation resource that the corresponding CO-vBS consumes to transcode the compressed videos at time slot t, which is calculated by

$$E^b_{\text{com}}(t) = \sum_{k\in\mathcal{K}_b} w_f f^b_k(t) \tag{5}$$

where w_f is the unit price of the computation resource. We denote the computation resource allocation decision of cooperative system as $\mathbf{F} = \{f^b_k(t)|b \in \mathcal{B}, k \in \mathcal{K}_b, t \in T\}$.

3) Transmission Delay and Cost of Access Link: The last stage is video transmission from the current vBSs of physical BSs to their associated users via the downlink of wireless access network. We use the orthogonal frequency division multiple access method for each cell's downlink communication without intracell interference [35]. The total bandwidth of the downlink is B_w Hz, which is divided into S subcarriers, each occupying a bandwidth of B_w/S Hz. Let $\mathcal{S} = \{1, \ldots, S\}$ denote the set of subcarriers. The binary variable $x^b_{k,s}(t) = 1$ indicates that the subcarrier s is allocated to the user k associated physical BS b at time slot t, otherwise $x^b_{k,s}(t) = 0$. In addition, we define the set of subcarrier and transmission power allocation of downlink at slot t as $\mathbf{X} = \{x^b_{k,s}(t)|b \in \mathcal{B}, k \in \mathcal{K}_b, s \in \mathcal{S}, t \in T\}$ and $\mathbf{P} = \{p^b_{k,s}(t)|b \in \mathcal{B}, k \in \mathcal{K}_b, s \in \mathcal{S}, t \in T\}$, where $p^b_{k,s}(t)$ is the transmission power of the user k associated with physical BS b and subcarrier s. The channel between a physical BS and a user is assumed to be a Rayleigh fading channel, which is independent and identical distributed over time. Then, the signal to interference plus noise ratio (SINR) of user k associated with physical BS b and subcarrier s at time slot t is given by

$$\gamma^b_{k,s}(t) = \frac{p^b_{k,s}(t)h^b_{k,s}(t)\left(d^b_k(t)\right)^{-\alpha}}{I^b_{k,s}(t) + \frac{B_w}{S}N_0} \tag{6}$$

where $h^b_{k,s}(t)$ is the channel gain of the desired transmission path between user k and its associated physical BS b on subcarrier s at slot t, which follows a unit-mean exponential distribution. The path-loss between the physical BS b and user k is modeled as $\left(d^b_k(t)\right)^{-\alpha}$, where $d^b_k(t)$ denotes the reference distance between them in the path loss model, and α is the path-loss exponent. $I^b_{k,s}(t) = \sum_{b'\in\mathcal{B}\setminus\{b\}} p^{b'}_{k,s}(t)h^{b'}_{k,s}(t)(d^{b'}_k(t))^{-\alpha}$ is the intercell interference of user k associated with physical BS b and subcarrier s at time slot t. N_0 is the power spectral density of the additive white Gaussian noise.

Hence, the achievable data rate of user k on subcarrier s associated with physical BS b at time slot t is given by

$$r^b_{k,s}(t) = \frac{B_w}{S}\log_2\left(1 + \frac{p^b_{k,s}(t)h^b_{k,s}(t)}{I^b_{k,s}(t) + \frac{B_w}{S}N_0}\right). \tag{7}$$

Based on (7), the total transmission rate of user k associated with physical BS b over the access link at time slot t is expressed as

$$R^b_k(t) = \sum_{s\in\mathcal{S}} x^b_{k,s}(t)r^b_{k,s}(t). \tag{8}$$

The transmission delay that physical BS b delivers video v to the associated user k depends on the total transmission rate of user k over the access link and the size of the uncompressed requested video v_1, which is given by

$$D^{b,ac}_{k,v}(t) = \frac{z_{v_1}}{R^b_k(t)}. \tag{9}$$

Meanwhile, the transmission cost of physical BS b over the access link at time slot t can be measured by the usage of bandwidth and transmission power, which is given by

$$E^b_{ac}(t) = \sum_{k\in\mathcal{K}_b}\left[w_{bw}\sum_{s\in S} x^b_{k,s}(t) + w_p p^b_k(t)\right] \tag{10}$$

where w_{bw} and w_p are the prices of unit bandwidth and unit power, respectively. Moreover, the transmission power of user k associated with physical BS b is expressed as $p^b_k(t) = \sum_{s\in S} x^b_{k,s}(t)p^b_{k,s}(t) + p^c_k$, where p^c_k is the circuit power consumption of user k associated with physical BS b.

4) Total Delay and Cost: Based on the above analysis, the user delay and system operative cost can be obtained. Specifically, the delay that users obtain the requested videos is composed of the retrieval delay, the computation delay, and the transmission delay over the access link. Based on (1), (4), and (9), the delay of user k associated with physical BS b at time slot t is expressed by

$$D^b_k(t) = \sum_{v\in\mathcal{V}} q^b_{k,v}(t)\left(D^{b,re}_{k,v}(t) + D^{b,\text{com}}_{k,v}(t) + D^{b,ac}_{k,v}(t)\right). \tag{11}$$

Thus, the total delay of all users in the proposed cooperative caching system is denoted as $D(t) = \sum_{b\in\mathcal{B}}\sum_{k\in\mathcal{K}_b} D^b_k(t)$. Correspondingly, we define the total system operative cost as the operative cost of all physical BSs. The total system operative cost at time slot t can be expressed as $E(t) = \sum_{b\in\mathcal{B}} E^b(t)$, where $E^b(t)$ is operative cost of physical BS b. The operative cost of physical BS b depends on not only retrieval cost, computation cost and access cost but video caching cost in the video caching phase. The operative cost of physical BS b at time slot t is modeled as

$$E^b(t) = E^b_{re}(t) + E^b_{\text{com}}(t) + E^b_{ac}(t) + E^b_{ca}(t) \tag{12}$$

where $E^b_{ca}(t)$ represents the caching cost of physical BS b at time slot t. The caching cost of physical BS b is related to the operations that physical BS b refreshes the videos cached in its current two vBSs' storage space. The caching cost of physical BS b at time slot t is given by

$$E^b_{ca}(t) = \omega_c\sum_{v\in\mathcal{V}}\sum_{i\in\{1,2\}} c^b_{v_i}(t)\left(1 - c^b_{v_i}(t-1)\right) \tag{13}$$

where w_c is the unit price of video updating, and the sum term is the number of uncompressed and compressed videos to be fetched and cached at time slot t, which were not cached at the time slot $(t-1)$ [36].

D. Problem Formulation

In general, the users tend to obtain the required videos with minimal delay while the MNO's objective is to provide video services at the least operative cost. From both the perspectives of users and MNO, this work aims to strike a balance between the system operation cost and user delay under the delay constraints of users, which is achieved by optimizing the video caching, video sharing, communication and computation resource allocation under the limited storage space, computation capacity, and communication resource. The optimization problem is formulated as follows:

$$
\begin{aligned}
\min_{\mathbf{C,Y,F,X,P}} \quad & \sum_{t=0}^{T-1} [\delta \varepsilon D(t) + (1-\delta) E(t)] \\
\text{s.t.} \quad & (\text{C1}): D_k^b(t) \le D_{k,th}^b(t) \quad \forall b, k, t \\
& (\text{C2}): \sum_{v \in \mathcal{V}} \sum_{i \in \{1,2\}} z_{v_i} c_{v_i}^b(t) \le z \quad \forall b, m \\
& (\text{C3}): \sum_{i \in \{1,2\}} c_{v_i}^b(t) \le 1 \quad \forall v, b, t \\
& (\text{C4}): y_{v_i}^{b,l}(t) \le c_{v_i}^l(t-1) \quad \forall l, v, b, i, t \\
& (\text{C5}): \sum_{l \in \mathcal{B} \cup o} \sum_{i \in \{1,2\}} y_{v_i}^{b,l}(t) \le 1 \quad \forall v, b, t \\
& (\text{C6}): \sum_{l \in \mathcal{B} \cup o} \sum_{i \in \{1,2\}} y_{v_i}^{b,l}(t) \ge \Gamma\left(\sum_{k \in \mathcal{K}_b} q_{k,v}^b(t) \right) \\
& \qquad \forall v, b, t, \\
& (\text{C7}): \sum_{k \in \mathcal{K}_b} x_{k,s}^b(t) \le 1 \quad \forall b, s \\
& (\text{C8}): \sum_{k \in \mathcal{K}_b} p_k^b(t) \le p_{\max}^b \quad \forall b \\
& (\text{C9}): \sum_{k \in \mathcal{K}_b} f_k^b(t) \le F \quad \forall b \\
& (\text{C10}): p_{k,s}^b(t) \ge 0 \quad \forall k, s, b \\
& (\text{C11}): f_k^b(t) \ge 0 \quad \forall k, b \\
& (\text{C12}): x_{k,s}^b(t), c_{v_i}^b(t), y_{v_i}^{b,l}(t) \in \{0, 1\} \\
& \qquad \forall b, k, l, v, i
\end{aligned} \tag{14}
$$

where ε is the normalized factor to make user delay and system operative cost in a similar scale, tradeoff factor $\delta \in [0, 1]$ is used to balance user delay and system operative cost, where a large δ emphasizes the reduction of user delay by sacrificing the system operation cost. (C1) represents that the delay of each user can not exceed its allowed delay threshold denoted by $D_{k,th}^b(t)$. (C2) is the cache capacity limitation of each physical BS. (C3) implies the corresponding CA-vBS and CO-vBS of each physical BS can not cache the same video, which can improve the video diversity at the edge. (C4) specifies the range of network entities from which the current vBSs can retrieve the requested videos at time slot t. The range of network entities refers to the network entities that have cached the requested videos during the video caching phase of time slot $(t-1)$. In (C6), we define a function $\Gamma(x)$ that its function value is 1 if $x > 0$ and 0 if $x \le 0$. (C5) and (C6) ensure that each physical BS selects the most suitable network entity to retrieve each requested video. (C7) guarantees that each physical BS could only allocate each subcarrier to at most one associated user. (C8) limits the maximal transmission power of physical BS $b \in \mathcal{B}$ to $p_{\max}^b$. (C9) is to ensure that the computation resource that each CO-vBS consumes to transcode the compressed videos should not surpass its corresponding physical BS's computation capacity. (C10) and (C11) indicate that the value of the transmission power and computation resource used by each user is nonnegative. (C12) ensures binary-valued $x_{k,s}^b(t)$, $c_{v_i}^b(t)$, $y_{v_i}^{b,l}(t)$.

The long-term optimization problem involves multiple optimization variables, i.e., video caching, video sharing, computation resource, subcarrier, and power allocation. Those variables are deeply coupled in the objective function and constraints, which makes this problem nonconvex. Furthermore, the presence of unknown future information, such as video popularity and wireless channel state, further complicates obtaining a long-term global optimal solution. To approach a long-term optimal solution, employing DRL algorithms is a viable approach. However, directly using DRL algorithms to solve the problem may face difficulties in convergence due to the extensive action space involving five variables. To address this complicated problem, an efficient two-layer DRL algorithm is proposed in the following section.

III. Solution Based on Two-Layer DRL Framework

Due to the interdependence of multiple variables, we propose to decouple problem (14) into two subproblems and employ a two-layer DRL framework to solve the subproblems sequentially. The details of each layer are given in the following two sections.

A. Inner Layer—Joint Video Sharing and Computation Resource Allocation Subproblem

Under given video caching and communication resource allocation policy {**C, X, P**}, the inner subproblem is to minimize the weighted sum of system operative cost and user delay via optimizing video sharing decision and computation resource allocation variables, i.e.,

$$
\begin{aligned}
\min_{\mathbf{Y,F}} \quad & \sum_{t=0}^{T-1} [\delta \varepsilon D(t) + (1-\delta) E(t)] \\
\text{s.t.} \quad & (\text{C1}), (\text{C4}) - (\text{C6}), (\text{C9}), (\text{C11}), (\text{C12}).
\end{aligned} \tag{15}
$$

By analyzing the objective function of problem (15), video sharing and computation resource allocation decisions at any time slot t only impact retrieval delay and cost as well as computation cost and delay in t. In addition, the constraints of problem (15) are independent in each time slot. Hence, optimizing long-term optimization problem (15) can be recast as optimizing multiple one-shot problems, whose objectives

are to minimize instantaneous retrieval delay and cost as well as computation cost and delay.

Furthermore, video sharing and resource allocation decisions of each physical BS b in any time slot t (i.e., $\{\mathbf{Y}_b^t, \mathbf{F}_b^t\}$) only affect its own retrieval and computation cost as well as the retrieval and computation delay of its associated users in t, i.e., $E_{re}^b(t)$, $E_{\text{com}}^b(t)$, $D_{k,v}^{b,re}(t)$ and $D_{k,v}^{b,\text{com}}(t)$ $\forall k \in \mathcal{K}_b$. Meanwhile, each physical BS makes video sharing and resource allocation decisions in each time slot independently. Therefore, the video sharing and computation resource allocation variables of each physical BS can be optimized independently. For physical BS b, the video sharing and computation variables $\{\mathbf{Y}_b^t, \mathbf{F}_b^t\}$ are optimized to minimize its retrieval delay and cost as well as computation cost and delay under caching state and communication resource allocation decision at time slot t. The corresponding problem can be reformulated as

$$\begin{aligned} \min_{\mathbf{Y_b^t},\ \mathbf{F_b^t}} \quad & f_{Y,F}^b(t) \\ \text{s.t.} \quad & \text{(C1), (C4)}-\text{(C6), (C9), (C11), (C12)} \end{aligned} \tag{16}$$

where

$$\begin{aligned} f_{Y,F}^b(t) = \sum_{v\in\mathcal{V}}\sum_{l\in\mathcal{B}\cup o}\Bigg[\sum_{i\in\{1,2\}} y_{v_i}^{b,l}(t) \\ \left(\sum_{k\in\mathcal{K}_b} q_{k,v}^b(t)\left(\eta_{b,l}\frac{z_{v_i}}{R_l} + d_{b,l}\frac{\zeta}{\xi}\right)\varepsilon_1^{'} + w_{m,l}^i \eta_{b,l}\varepsilon_2^{'}\right) \\ + y_{v_2}^{b,l}(t)\sum_{k\in\mathcal{K}_b} q_{k,v}^b(t)\frac{z_{v_2}\varepsilon_1^{'}\lambda_{k,v}^b}{f_k^b(t)}\Bigg] + \sum_{k\in\mathcal{K}_b} w_f\varepsilon_2^{'} f_k^b(t) \end{aligned} \tag{17}$$

and $\varepsilon_1^{'} = \delta\varepsilon$, $\varepsilon_2^{'} = 1-\delta$. Base on (4), (11), and (17), it can be observed that the constraint (C1) and the objective function $f_{Y,F}^b(t)$ is related to a fractional term $y_{v_2}^{b,l}(t)\cdot\sum_{k\in\mathcal{K}_b} q_{k,v}^b(t)([z_{v_2}\varepsilon_1^{'}\lambda_{k,v}^b]/f_k^b(t))$. Due to the coupling between $\mathbf{Y}_b^t$ and $\mathbf{F}_b^t$ in the fractional term, $f_{Y,F}^b(t)$ and the constraint (C1) are nonconvex, resulting that problem (16) is difficult to solve. Hence, we develop an alternating optimization algorithm to optimize video sharing and computation resource allocation alternatively. The details of the alternating optimization algorithm are given as follows.

1) Optimization of $\mathbf{F}_b^t$: The computation resource allocation problem for a given $\{\mathbf{Y}_b^t\}$ from (16) becomes

$$\begin{aligned} \min_{\mathbf{F}_b^t} \quad & \sum_{k\in\mathcal{K}_b}\left[\sum_{v\in\mathcal{V}}\sum_{l\in\mathcal{B}\cup o} q_{k,v}^b(t) y_{v_2}^{b,l}(t)\frac{z_{v_2}\varepsilon_1^{'}\lambda_{k,v}^b}{f_k^b(t)} + w_f\varepsilon_2^{'} f_k^b(t)\right] \\ \text{s.t.} \quad & \text{(C1)}: f_k^b(t) \geq \frac{\sum_{v\in\mathcal{V}}\sum_{l\in\mathcal{B}\cup o} q_{k,v}^b(t) y_{v_2}^{b,l}(t) z_{v_2}\lambda_{k,v}^b}{D_{k,th}^b(t) - \sum_{v\in\mathcal{V}} q_{k,v}^b(t)\left(D_{k,v}^{b,re}(t) + D_{k,v}^{b,ac}(t)\right)} \\ & \forall k, b, t \\ & \text{(C9)}: \sum_{k\in\mathcal{K}_b} f_k^b(t) \leq F \quad \forall k, b, t \end{aligned} \tag{18}$$

we can prove that problem (18) is a convex optimization problem. Specifically, the constraint (C1) indicates that the computation resource allocation variable $f_k^b(t)$ is nonnegative. Hence, $(1/f_k^b(t))$ and $f_k^b(t)$ are convex functions with respect to $f_k^b(t)$. Accordingly, the objective function of problem (18) remains convex due to the convexity of convex functions' linear combination. In addition, (C1) and (C9) are all linear. Therefore, problem (18) is convex with respect to $\mathbf{F}_b^t$. Consequently, the optimal solution of computation resource allocation at time slot t is obtained by using Lemma 1.

Algorithm 1: Binary Search-Based Method for the Lagrange Multiplier

Input: Give a big enough $\theta_{\text{up}}^b(t)$, initialize $\theta_{\text{low}}^b(t) = 0$, $\theta^b(t) = \frac{\theta_{\text{low}}^b(t)+\theta_{\text{up}}^b(t)}{2}$, set success = False;
Output: Optimal value of the Lagrange multiplier $\theta^{b*}(t)$;

```
while NOT success do
    Calculate H^b(θ^b(t)) according to (20);
    if 0 ≤ H^b(θ^b(t)) ≤ Λ_1 then
        Obtain the optimal computation resource
        allocation with given Y_b^t and accuracy level Λ_1,
        set success = True;
    else if H^b(θ^b(t)) < 0 then
        Halve the searching region according to
        θ_low^b(t) = θ^b(t), θ^b(t) = (θ_low^b(t)+θ_up^b(t))/2
    else
        Halve the searching region according to
        θ_up^b(t) = θ^b(t), θ^b(t) = (θ_low^b(t)+θ_up^b(t))/2
    end
end
```

Lemma 1: The optimal computation resource allocation solution for problem (18) meets the following condition:

$$f_k^b(t) = \max\left\{\phi_k^b(t), \varphi_k^b(t)\right\} \tag{19}$$

where

$$\phi_k^b(t) = \sqrt{\frac{\sum_{v\in\mathcal{V}}\sum_{l\in\mathcal{B}\cup o} y_{v_2}^{b,l}(t) q_{k,v}^b(t) z_{v_2}\varepsilon_1^{'}\lambda_{k,v}^b}{w_f\varepsilon_2^{'} + \theta^b(t)}}$$

$$\varphi_k^b(t) = \frac{\sum_{v\in\mathcal{V}}\sum_{l\in\mathcal{B}\cup o} q_{k,v}^b(t) y_{v_2}^{b,l}(t) z_{v_2}\lambda_{k,v}^b}{D_{k,th}^b(t) - \sum_{v\in\mathcal{V}} q_{k,v}^b(t)\left(D_{k,v}^{b,re}(t) + D_{k,v}^{b,ac}(t)\right)}$$

and $\theta^b(t) \geq 0$ is a Lagrange multiplier, which is determined by the equation $\sum_{k\in\mathcal{K}_b} f_k^b(t) = F$.

Proof: Please see Appendix A. ∎

By substituting (19) into $\sum_{k\in\mathcal{K}_b} f_k^b(t) = F$, it is difficult to derive analytically $\theta^b(t)$. To get the optimal value of $\theta^b(t)$ numerically, we define a function with respect to $\theta^b(t)$, which is given by

$$H^b\left(\theta^b(t)\right) = F - \sum_{k\in\mathcal{K}_b}\left[\beta_k^b(t)\phi_k^b(t) + \left(1-\beta_k^b(t)\right)\varphi_k^b(t)\right] \tag{20}$$

where $\beta_k^b(t) = 1$ if $\phi_k^b(t) \geq \varphi_k^b(t)$ and $\beta_k^b(t) = 0$ otherwise. Since $\phi_k^b(t)$ monotonically decreases with $\theta^b(t)$, $H^b(\theta^b(t))$ is monotonically decreasing function with respect to $\theta^b(t)$. It is obtained that the low bound of $\theta^b(t)$ is 0 from Lemma 1. A binary search method (Algorithm 1) is proposed to obtain the optimal value of $\theta^b(t)$ numerically within initial searching region $[0, \theta_{\text{up}}^b(t)]$. Begin with the value $\theta^b(t) =$

$([\theta^b_{\text{low}}(t) + \theta^b_{\text{up}}(t)]/2)$, we iteratively calculate the $H^b(\theta^b(t))$ and the computation resource $f^b_k(t)$ for current value $\theta^b(t)$. The searching region is divided in half, with the larger half preserved if $H^b(\theta^b(t)) < 0$, and the smaller half retained if $H^b(\theta^b(t)) > \Lambda_1$. Once the given precision requirement (i.e., Λ_1) is satisfied, the searching will be terminated. By substituting the optimal value of the Lagrange multiplier $\theta^{b*}(t)$ into (19), the optimal computation resource allocation $\mathbf{F^t_b}$ is obtained with given $\mathbf{Y}^t_b$.

2) Optimization of $\mathbf{Y}^t_b$: Given $\mathbf{F}^t_b$, problem (16) is nonconvex with respect to $y^{b,l}_{v_i}(t)$. To this end, we relax the value of $y^{b,l}_{v_i}(t)$ to the interval of [0, 1]. Accordingly, problem (16) is recast as

$$\begin{aligned} \min_{\mathbf{Y^t_b}} \quad & f^b_Y(t) \\ \text{s.t.} \quad & \text{(C1), (C4)} - \text{(C6)}, \\ & \text{(C12)}: 0 \le y^{b,l}_{v_i}(t) \le 1 \quad \forall b, l, v, i \end{aligned} \tag{21}$$

where

$$\begin{aligned} f^b_Y(t) = \sum_{v\in\mathcal{V}} \sum_{l\in\mathcal{B}\cup o} \Bigg[& \sum_{i\in\{1,2\}} y^{b,l}_{v_i}(t) \\ & \left(\sum_{k\in\mathcal{K}_b} q^b_{k,v}(t) \left(\eta_{b,l} \frac{z_{v_i}}{R_l} + d_{b,l} \frac{\zeta}{\xi} \right) \varepsilon_1^{'} + w^i_{m,l} \eta_{b,l} \varepsilon_2^{'} \right) \\ & + y^{b,l}_{v_2}(t) \sum_{k\in\mathcal{K}_b} q^b_{k,v}(t) \frac{z_{v_2} \varepsilon_1^{'} \lambda^b_{k,v}}{f^b_k(t)} \Bigg]. \end{aligned} \tag{22}$$

It is observed from (21) that the objective function and constraints are all linear. Therefore, the optimal video sharing decision for each physical BS can be directly obtained via convex optimization toolkit, such as CVXPY [37]. Note that the relaxation can be regarded as a cooperative transmission among network entities, where the set of network entities is denoted by $\{l | y^{b,l}_{v_i}(t) \neq 0 \quad \forall l \in B \cup o\}$. To illustrate, a partial video sharing indicator $y^{b,l}_{v_i}(t) \neq 0$ means that network entity l transmits the portion $y^{b,l}_{v_i}(t)$ of the video v_i to current vBS b at time slot t. After the cooperative transmission among multiple network entities in the video sharing phase, the current vBS b receives the complete requested video v_i, and subsequently transmits the requested video v_i to its associated users in the video delivery phase.

Combining the solutions for $\mathbf{F}^b_t$ and $\mathbf{Y}^b_t$, problem (16) can be addressed via an alternative optimization method. Specifically, by updating the solutions for $\mathbf{F}^b_t$ and $\mathbf{Y}^b_t$ at each iteration d, we can obtain the optimal video sharing and computation resource allocation until the decrease in objective of problem (16) falls below a threshold Λ_2. The detailed procedure for solving problem (16) are summarized in Algorithm 2. It is noted that the inner optimization problem (16) can be infeasible if the user delay constraint (C1) is not satisfied by performing the joint decision of video caching and communication resource allocation. The infeasibility issue can be solved by introducing a penalty mechanism in the outer layer algorithm, which is detailed in the next section.

Algorithm 2: Alternating Optimization Algorithm for Video Sharing and Computation Resource Allocation

Input: Initialize the video sharing decision $\mathbf{Y^{t,1}_b}$, set the iteration number $d = 1$ and accuracy level Λ_2,.
Output: Video sharing variables $\mathbf{Y^t_b}$ and computation resource allocation decisions $\mathbf{F^t_b}$;

```
while |f^{b,(d+1)}_{Y,F}(t) − f^{b,(d)}_{Y,F}(t)| < Λ2 do
    Solve problem (18) for given Y^{t,(d)}_b, and obtain the optimal computation resource allocation F^{t,(d)}_b by using Lemma 1 and Algortihtm 1;
    Solve problem (21) for given F^{t,(d)}_b, and get the optimal video sharing strategy Y^{t,(d+1)}_b via convex optimization solvers;
    Update d = d + 1;
end
```

B. Outer Layer—Video Caching and Communication Resource Allocation Subproblem

By fixing $\mathbf{Y}$ and $\mathbf{F}$, problem (14) can be recast as outer subproblem (23). The outer subproblem is to select which videos are cached at vBSs and allocate communication resource to users with the objective of minimizing the long-term weighted sum of user delay and system operative cost, i.e.,

$$\begin{aligned} \min_{\mathbf{C},\mathbf{X},\mathbf{P}} \quad & \sum_{t=0}^{T-1} [\delta \varepsilon_1 D(t) + (1-\delta)\varepsilon_2 E(t)] \\ \text{s.t.} \quad & \text{(C1), (C4)}-\text{(C6), (C9), (C11), (C12).} \end{aligned} \tag{23}$$

By analyzing the objective of problem (23), the current decisions on video caching and communication resource allocation $\{\mathbf{C}^t, \mathbf{X}^t, \mathbf{P}^t\}$ impact not only the current user delay $D(t)$ and system operative cost $E(t)$ but also affect the future state, i.e., $D(t+1)$ and $E(t+1)$, forming a sequence decision problem. In addition, the video caching and communication resource allocation decisions of each physical BS b $\{\mathbf{C}^t_b, \mathbf{X}^t_b, \mathbf{P}^t_b\}$ impact user delay and operative cost of other physical BSs, i.e., $D^{b'}_k(t)$ and $E^{b'}(t)$, $b' \in \mathcal{B}\backslash\{b\}$, which is caused by video sharing and spectrum resource interference among physical BSs. The above analysis suggests that the cooperative caching and dynamic communication resource allocation in multiple cells resemble a stochastic game. Consequently, we reformulate problem (23) as a stochastic game and use the MADDPG algorithm to solve it [38], [39].

Hence, a tuple of $G =< \mathcal{B}, \mathcal{S}, \{\mathcal{A}_b\}_{b\in\mathcal{B}}, \mathcal{O}, \mathcal{P}, \mathcal{R}, \gamma >$ is defined for the stochastic game, where $\mathcal{B} = \{1, 2, \ldots, B\}$ is the set of B agents. Each physical BS b is regarded as agent b. $\mathcal{S}$ is the state space of the entire cooperative caching system. $\mathcal{A}_b$ is the action space of physical BS b, which is a set of all possible actions of physical BS b. $\mathcal{A} = \mathcal{A}_1 \times \mathcal{A}_2 \times \cdots \times \mathcal{A}_B$ is joint action space of all agents. $\mathcal{O}$ is the observation set of all agents. In each time slot, each agent $b \in \mathcal{B}$ makes its action decision $\mathbf{a}^t_b \in \mathcal{A}_b$ based on local observation $\mathbf{o}^t_b$ of system state $\mathbf{s}^t \in \mathcal{S}$, thereby forming a joint action of all agents, i.e., $\mathbf{a}^t = \{\mathbf{a}^t_1, \mathbf{a}^t_2, \ldots, \mathbf{a}^t_B\}$. After taking the joint action $\mathbf{a}^t$, system state changes from current state $\mathbf{s}^t = \{\mathbf{o}^t_1, \mathbf{o}^t_2, \ldots, \mathbf{o}^t_B\}$ to next

state $\mathbf{s}^{t+1} = \{\mathbf{o}_1^{t+1}, \mathbf{o}_2^{t+1}, \ldots, \mathbf{o}_B^{t+1}\}$. $\mathcal{P}$ denotes the transition probability function among different states. Considering that the objective of problem (23) is to minimize the weighted sum of total user delay and system operative cost, all agents have the same reward function, $\mathcal{R}$:$S \times A \rightarrow \mathbb{R}$, $\gamma \in [0, 1]$ denotes the discount factor. For each physical BS, the observation, action and reward are formulated as:

Observation: The observation space of BS b at slot t contains the channel states, users' requests and video caching decision in the edge at the last time slot, i.e.,

$$\mathbf{o}_b^t = \{\mathbf{h}^b(t), \mathbf{h}_{in}^b(t), \mathbf{q}^b(t), \mathbf{D}_{th}^b(t), \mathbf{C}^b(t-1)\} \tag{24}$$

where $\mathbf{h}^b(t)$ is the set of the channel gain $h_{k,s}^b(t)$ and $\mathbf{h}_{in}^b(t)$ is the set of interference channel gain of users, $\mathbf{q}^b(t)$ is the set of users' request for videos, $\mathbf{D}_{th}^b(t) = \{D_{k,th}^b(t) | \forall k \in K_b\}$ is the set of users' delay requirement at time slot t, and $\mathbf{C}^b(t-1)$ is the set of video caching decision at last time slot $t-1$ in of CA-vBS b and CO-vBS b.

Action: Consistent with the decision variables in the problem (23), the action set includes the caching decision variables, subcarrier and power resources allocation variables, i.e.,

$$\mathbf{a}_{b1}^t = \{\mathbf{C}_b^t, \mathbf{X}_b^t, \mathbf{P}_b^t\}. \tag{25}$$

To maintain continuous action space required by the MADDPG algorithm, the binary variables $\{\mathbf{C}_b^t, \mathbf{X}_b^t\}$ are relaxed to continuous variables ranging from 0 to 1. Note that Each agent checks the selected actions, and then modifies any actions violating the constraints of problem (23). Specifically, if some of actions $\mathbf{C}_b^t$ and $\mathbf{X}_b^t$ violate caching decision indicator and subcarrier allocation constraints (C3), (C7), agent b will reserve one caching decision indicator or one subcarrier allocation indicator and modify the conflicting actions as zero. For the convenience of analysis, the storage space constraint of each physical BS (C2) is transformed as the maximum number of videos that its corresponding CA-vBS and CO-vBS can cache [40]. Assume that each CA-vBS can cache up to V_1 uncompressed videos and the storage space of each CO-vBS is V_2 uncompressed videos. If the selected $\mathbf{C_b^t}$ does not meet the storage space constraint of each vBS, the value of some cache actions will be replaced by zero to ensure that constraint (C2) is not violated. Besides, some of the selected actions $\mathbf{P}_b^t$ will be modified to the low values to satisfy the transmission power constraints (C8), (C10) if they are not within the limitation range of transmission power.

Reward: The reward function should assess how the actions taken impact the performance of the system [41]. In this system, we minimize the long-term total user delay and system operative cost while satisfying users' delay requirements. If user delay exceeds the maximum tolerable delay, these actions are regarded as detrimental to the system's performance and the agent should face penalties. Therefore, the reward function is designed to include both the optimization objective of problem (23) and a penalty function $r_p(t)$, which is defined based on the constraint (C1), i.e.,

$$r_p(t) = \sum_{b \in \mathcal{B}} \sum_{k \in \mathcal{K}_b} o_p \left(D_{k,th}^b(t) - D_k^b(t) \right)$$

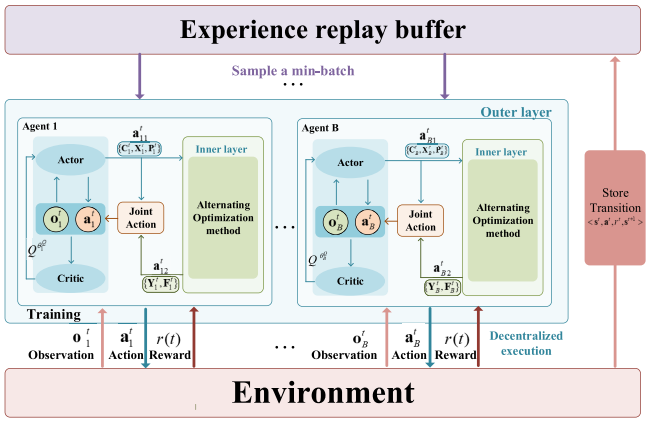


Fig. 3. Proposed algorithm framework. There are B agents in the outer layer, and the actor network of each agent makes decisions about video caching and communication resource allocation $\mathbf{a}_{b1}^t = \{\mathbf{C}_b^t, \mathbf{X}_b^t, \mathbf{P}_b^t\}$ according to local observation $\mathbf{o}_b^t$. Then, the alternating optimization method in the inner layer generates the video sharing and computational resource allocation decisions $\mathbf{a}_{b2}^t = \{\mathbf{Y}_b^t, \mathbf{F}_b^t\}$ based on the current state $\mathbf{o}_b^t$ and the action $\mathbf{a}_{b1}^t$.

where o_p is the penalty coefficient for not satisfying the maximum tolerable delay for users. Note that the penalty mechanism not only guides the agents to take actions ensuring the existence of feasible solutions to the inner optimization problem but also encourages agents to take actions reducing user delay. At time slot t, the immediate reward for agent b is expressed as

$$r(t) = -\delta \varepsilon_1 D(t) - (1-\delta)\varepsilon_2 E(t) + r_p(t). \tag{26}$$

Therefore, the expectation of the long-term discounted cumulative reward of each agent is defined as

$$J(\mu) = E_\mu \Big[r(0) + \gamma r(1) + \gamma^2 r(2) + \cdots + \gamma^{T-1} r(T-1) \Big]$$

where μ is the policy of actor in agents. By utilizing the system-wide performance as the immediate reward function for each agent, cooperation is promoted among the agents.

C. Proposed Algorithm

Combining the above-given inner and outer layers, we design a two-layer MADDPG method to solve the cooperative caching problem (14). The two-layer architecture of the proposed algorithm is shown in Fig. 3. In the outer layer, the actor network of each agent makes decisions about communication resource allocation and video caching based on local observation in a decentralized manner. For agent b, we denote the neural network parameters of the online network and target network in the actor network as θ_b^μ and $\theta_b^{\mu'}$, and the corresponding parameters in the critic network are denoted as θ_b^Q and $\theta_b^{Q'}$. At time slot t, the actor network makes video caching and communication resource allocation decisions $\mathbf{a}_{b1}^t = \{\mathbf{C}_b^t, \mathbf{X}_b^t, \mathbf{P}_b^t\}$ based on current local observation $\mathbf{o}_b^t$, i.e.,

$$\mathbf{a}_{b1}^t = \mu\left(\mathbf{o}_b^t | \theta_b^\mu\right) + \mathbf{N}_b \tag{27}$$

where $\mathbf{N}_b$ is exploration noise.

Then, an alternating optimization method in the inner layer of agent b generates the video sharing and computation

resource allocation decisions $\mathbf{a}_{b2}^t = \{\mathbf{Y}_b^t, \mathbf{F}_b^t\}$ based on the current state and the action $\mathbf{a}_{b1}^t$. Note that each agent may make communication resource allocation decisions $\{\mathbf{X}_b^t, \mathbf{P}_b^t\}$ that enable the transmission delay of access link to be greater than users' maximum tolerate delay, i.e., $D_k^{b,ac}(t) > D_{k,th}^b(t)$. In this case, there is no feasible solutions to the inner optimization problem (16), and we can not calculate reward. To address this issue, we will set the variable $\tilde{D}_k^{b,ac}(t)$, and substitute $\tilde{D}_k^{b,ac}(t)$ for $D_k^{b,ac}(t)$ in the constraint (C1) of inner optimization problem (16), ensuring there are feasible solutions to the inner optimization problem (16). Accordingly, we obtain video sharing and computation resource allocation decisions. A joint decision set of agent b $\mathbf{a}_b^t = \{\mathbf{a}_{b1}^t, \mathbf{a}_{b2}^t\}$ can be obtained. The environment gives a reward and state transition as the feedback after taking joint actions of all agents. Afterward, the experience memory $\mathcal{D}$ stores all agents' current experience, which can be represented by $<\mathbf{s}^t, \mathbf{a}^t, r^t, \mathbf{s}^{t+1}>$. The storage size of the experience memory $\mathcal{D}$ is M_D, which is located at the cloud computing center. During the training phase, both the actor network and critic network of each agent are trained by utilizing a mini-batch of M_s samples from the experience memory. The parameter of each agent's actor network is updated by the following policy gradient:

$$\nabla_{\theta_b^\mu} J\left(\mu_{\theta_b^\mu}\right) = E_{\mathbf{s}^t,\mathbf{a}^t\sim\mathcal{D}}\Big[\nabla_{\theta_b^\mu}\mu\left(\mathbf{o}_b^t|\theta_b^\mu\right) \nabla_{\mathbf{a}_b^t} Q\left(\mathbf{s}^t, \mathbf{a}_1^t, \ldots, \mathbf{a}_B^t|\theta_b^Q\right)|_{\mathbf{a}_{b1}^t=\mu\left(\mathbf{o}_b^t|\theta_b^\mu\right)}\Big]. \quad (28)$$

The critic network of each agent in the outer layer is trained to assess the joint decision by estimating the Q-values based on global information. The critic online network of agent b is updated by minimizing its own loss function, i.e.,

$$L\left(\theta_b^Q\right) = E_{\mathbf{s}^t,\mathbf{a}^t,r^t,\mathbf{s}^{t+1}\sim\mathcal{D}}\Big[r_b^t + \gamma Q^{\theta_b^{Q'}}\left(\mathbf{s}^{t+1}, \mathbf{a}_1^{t+1}, \ldots \mathbf{a}_B^{t+1}\right)|_{\mathbf{a}_{b1}^{t+1}=\mu\left(\mathbf{o}_b^{t+1}|\theta_b^\mu\right)} - Q^{\theta_b^Q}\left(\mathbf{s}^t, \mathbf{a}_1^t, \ldots, \mathbf{a}_B^t\right)^2\Big] \quad (29)$$

where $\mathbf{a}_b^{t+1} = \{\mathbf{a}_{b1}^{t+1}, \mathbf{a}_{b2}^{t+1}\}$.

For agent b, based on the soft update rule, the parameters of the target network for both the actor network and critic network are updated by gradually tracking the corresponding online networks [42], i.e.,

$$\theta_b^{\mu'} \leftarrow \tau\theta_b^\mu + (1-\tau)\theta_b^{\mu'} \quad (30)$$

$$\theta_b^{Q'} \leftarrow \tau\theta_b^Q + (1-\tau)\theta_b^{Q'} \quad (31)$$

where τ is the update rate of target networks. The proposed algorithm is encapsulated in Algorithm 3. Considering that the cloud computing center has a significant computational advantage, the training process of all agents' neural networks, including the actor network and critic network, is completed at the cloud computing center in an offline model. After sufficient training, the cloud computing center transmits the training models to all agents (i.e., physical BSs) via high-speed backhaul links. Each physical BS is equipped with the trained actor network, which generates joint policies for video caching, sharing, and delivery based on local observation.

Algorithm 3: Proposed Algorithm

Input: Users' request for videos and corresponding delay requirement, channel state and the current video caching status at each BS, i.e., $\mathbf{o_b^t}$;

Output: The video caching, communication resource allocation, video sharing and computation resource allocation decisions $\{\mathbf{C}^t, \mathbf{X}^t, \mathbf{P}^t, \mathbf{Y}^t, \mathbf{F}^t\}$;

Initialization: Initialize actor and critic networks $(\theta_b^\mu, \theta_b^{\mu'}, \theta_b^Q, \theta_b^{Q'}, \forall b \in B)$ and the experience memory;
for *episode*$= \{1, 2, \ldots, E\}$ **do**
  Reset environment and obtain the initial observation $\mathbf{o}$ according to (24);
  for *slot* $t = \{1, 2, \ldots, T\}$ **do**
    ▷ Experience generation
    For each agent b, choose the action according to (27). Obtain the video caching and communication resource allocation decision $\mathbf{a}_{b1}^t = \{\mathbf{C}_b^t, \mathbf{X}_b^t, \mathbf{P}_b^t\}$;
    Obtain the video sharing and computation resource allocation decisions $\mathbf{a}_{b2}^t = \{\mathbf{Y}_b^t, \mathbf{F}_b^t\}$ by using Algorithm 1;
    Execute joint decision $\mathbf{a}^t = \{\mathbf{a}_{b1}^t, \mathbf{a}_{b2}^t, \forall b \in B\}$;
    Receive the reward r^t and obtain the new observation $\mathbf{o}'$;
    Store the transition $< \mathbf{o}^t, \mathbf{a}^t, r^t, \mathbf{o}^{t+1} >$ into experience memory;
    for *agent* $b = \{1, 2, \ldots, B\}$ **do**
      Sample randomly a mini-batch of M_s transitions from the experience memory;
      Update the critic online network by minimizing the loss function in (29);
      Update the actor online network by the policy gradient in (28);
    end
    Update target network actor and critic for each agent in (30) and (31).
  end
end

IV. Simulation Results and Discussion

This section evaluates the performance of our proposed algorithm based on the simulation results. The simulation experiments are conducted in a cellular network with four physical BSs, i.e., $B = 4$. There are different video popularity distributions within the coverage area of each physical BS. We assume that the video popularity follows a Zipf-like distribution. Similar to [32] and [43], the video popularity is modeled as a finite Markov state transition model, which includes four states $\{o_{v1}, o_{v2}, o_{v3}, o_{v4}\}$ with different parameters that can indicate the skewness of popularity, i.e., Δ_1 =0.8, Δ_2 =0.9, Δ_3 =1.0, Δ_4 =1.2. The popularity of video v with parameters Δ_j is $p_v^{(j)} = v^{-\Delta_j}/\sum_{i=1}^{V} i^{-\Delta_j}$. We denote the probability that video popularity transfer from state o_{vu} to state o_{vj} as P_{uj}, where $P_{uj} \in \{0.2, 0.4\}$ $\forall u, j \in \{1, 2, 3, 4\}$. To assess the performance of video caching decision, we define the cache

TABLE II
SYSTEM PARAMETERS

Parameter	Value	Parameter	Value
V	100	$p^b_{\max}$	40 W
S	40	p_c	0.1 W
K	16	R_o	150 Mbps
B_w	30 MHz	R_b	100 Mbps
N_0	-99 dBm	z_{v_1}	[1,2] Mbits
δ	0.6	$D^b_{k,th}$	[0,1] s
F	10 Gigacycle/s	ρ	0.33
w_{bw}	10^{-5}	λ^b_k	1000 cycle/bit
w_f	0.8×10^{-8}/cycles	$\eta_{b,o}$	12
w_p	20 $/W$	$\eta_{b,b'}$	{1,2,3}
ζ	1.5	$d_{b,o}$	13 km
ξ	3×10^8 m/s	$d_{b,b'}$	259.8m
ε	1000	$w^1_{m,o}$	100
o_p	300	$w^2_{m,o}$	50
V_1	3	$w^2_{m,b}$	1
V_2	7	$w^1_{m,b}$	2
E	500	T	100

TABLE III
HYPERPARAMETER OF PROPOSED ALGORITHM

Parameter	Value
Actor hidden layers	2
Critic hidden layers	2
Actor hidden units	64
Critic hidden units	64
Learning rate of actor	0.001
Learning rate of critic	0.001
Discount factor	0.9
Minibatch size	96
Optimizer	Adam

hit rate of physical BS b at time slot t as

$$\text{hit}_b(t) = \frac{\sum_{v\in\mathcal{V}}\sum_{k\in\mathcal{K}_b}\sum_{i\in\{1,2\}} q^b_{k,v}(t)c^b_{v_i}(t-1)}{\sum_{v\in\mathcal{V}}\sum_{i\in\{1,2\}} c^b_{v_i}(t-1)}.$$

The other simulation parameters are summarized in Table II, and the primary simulation environment settings of the proposed algorithm are concluded in Table III.

In addition, to reflect the advantages of our proposed joint optimization scheme, we compare it with different benchmark schemes, which are listed as follows,

1) *Proposed Scheme With Different Caching Methods:* We consider three caching strategies to replace the caching scheme in the proposed scheme, including the random-cache (RC) scheme, the least frequently used (LFU) scheme, and no video compression in proposed scheme (NVC). In the RC scheme, the caching decisions for each CA-vBS and CO-vBS are randomly selected. In the LFU scheme, the videos with the least requested times will be replaced in CA-vBSs and CO-vBSs. There are no CO-vBSs in the system and each CA-vBS only can cache uncompressed videos in the NVC scheme. In the above three schemes, the policies about video sharing and delivery keep the same as the proposed scheme, where the communication resource allocation policy is still determined by the MADDPG algorithm.

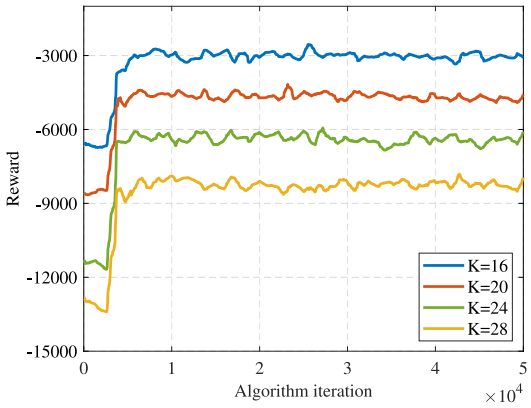


Fig. 4. Convergence performance for proposed algorithm under different numbers of users.

2) *Independent MADDPG With Proposed Video Sharing and Computation Resource Allocation Strategies (IMADDPG):* In this solution method, agents only have local observations from the environment and make caching and communication allocation decisions independently of each other, with the goal of maximizing the sum of the delay of local users and operative cost instead of the overall system reward.
3) *Random Communication Resource Allocation With Proposed Video Caching, Video Sharing and Computation Resource Allocation Strategies (R-Com):* Each vBS allocates randomly subcarriers and transmission power to its associated users. The caching decision and communication resource allocation policy remain consistent with the proposed scheme.
4) *Proposed Scheme With Noncooperative Caching (NCC) Mechanism:* In this scheme, the process of video sharing between different vBSs is not considered while video caching and delivery policies remain the same as the proposed scheme.

Fig. 4 illustrates the convergence performance with varying numbers of users. We employ a total of 50 000 algorithm iterations, calculated by multiplying 100 episodes and 500 time steps. From Fig. 4, we can see that the proposed algorithm can achieve a relatively stable reward value after about 7500 training iterations. It can also be observed that there are some minor fluctuations in reward values, which are caused by each agent's exploration in action space and the time-varying video popularity. In addition, the reward value is impacted by the number of users. With an increasing number of users, the reward value decreases. This phenomenon is explained by (26), where both delay and cost values, comprising the main components of the reward, grow as the number of users increases.

Fig. 5 presents a comparison of the reward under five different cache methods. It is observed that when the learning process becomes stable, our proposed scheme outperforms the other caching methods, by converging to a larger reward value. This is because our proposed scheme can adjust caching decisions along with video sharing, computation and communication resource allocation according to the time-varying user

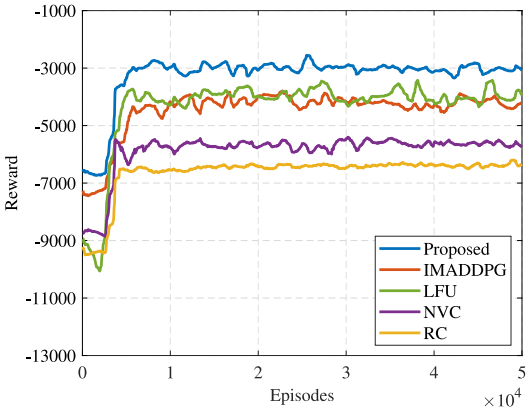


Fig. 5. Reward comparison of different caching methods.

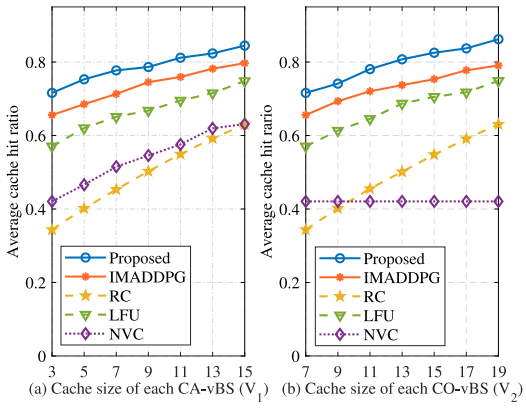


Fig. 6. Comparison of average cache hit rate versus the cache size of vBSs under different caching methods (a) Cache size of each CA-vBS (V_1) (b) Cache size of each CO-vBS (V_2).

preference. Meanwhile, RC neglects the cooperation cache among vBSs, resulting in a low-cache hit rate and high-video delivery delay, which are crucial components affecting the reward. Compared with the proposed scheme, IMADDPG converges to a lower reward because it relies on local observations, prioritizing local video popularity and neglecting cooperative caching between vBSs. Additionally, LFU demonstrates a relatively high-reward value as it leverages historical information about video popularity from the previous episode. Nevertheless, LFU exhibits a wider fluctuation range of reward due to its inability to learn the video popularity distribution and promptly adapt to time-varying user preferences like our proposed scheme. NVC exhibits low-video diversity and cache hit rate at the edge, attributed to caching fewer videos due to the absence of video compression. This can explain why NVC has the lowest reward value.

Fig. 6 depicts the performance of the proposed scheme in terms of the average cache hit rate under different cache size of vBSs. Here, the average cache hit rate is defined as the average value of the cache hit rate of all BSs $hit_b(t)$. From Fig. 6(a), we observe average cache hit rates of all caching schemes increase with the expansion of the cache space in CA-vBSs. This is because user requests are more likely to be satisfied by CA-vBSs that can cache more uncompressed videos with larger cache capacity. Furthermore, the proposed scheme consistently

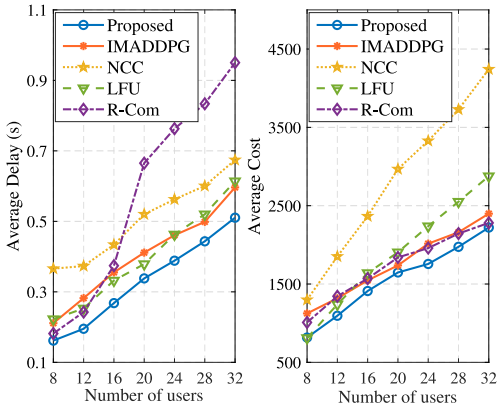


Fig. 7. Average delay and cost versus numbers of users under different schemes.

holds the advantage in average cache hit rate compared to all other caching methods. Compared with other caching methods, the high-average cache hit rate achieved by our proposed scheme is attributed not only to learning the changing mode of users' preference but facilitation of enhanced cooperation among all vBSs. There are some similar trends and characteristics regarding the average cache hit rate in Fig. 6(b) except for NVC. The average cache hit rate of NVC remains unchanged as the cache size of CO-vBSs increases due to the absence of CO-vBSs for caching compressed videos in the NVC scheme. In addition, from Fig. 6, we can see that both the increasing of cache space for CA-vBSs and CO-vBSs can achieve high-cache hit ratio. However, it may not be feasible if MNO would like to provide more video services with users at the edge by only increasing the cache space of CO-vBSs. This is because the CO-vBSs can not serve users if the computing resource of CO-vBS is not sufficient to transcode compressed videos.

Fig. 7 focuses on average delay and cost versus the number of users K under the five different schemes. For our proposed scheme, the average delay and cost arise almost linearly with the increasing of K. The behind reasons are explained as follows.

1) A higher number of users indicates more video requests. While the number of users is increasing, the probability that current vBSs have cached all the requested videos will reduce, and the current vBSs are more likely to fetch the request videos from other vBSs even remote video server. Therefore, retrieval delay $D_v^{b,re}(t)$ and retrieval cost $E_{re}^b(t)$ will rise up.
2) A higher number of users also implies more traffic loads. With the extension of the number of users, the current vBSs consume more spectrum resource and power to transmit more videos to users via access link, thereby leading to the increasing of transmission cost in the access link $E_{ac}^b(t)$. Given the limited communication resource within vBSs, the surge in the number of users inherently leads to the increase of the transmission delay in the access link $D_{k,v}^{b,ac}(t)$.

For the other four schemes, there are similar variations in average delay and cost. However, the four baseline schemes all perform worse than the proposed scheme. Specifically,

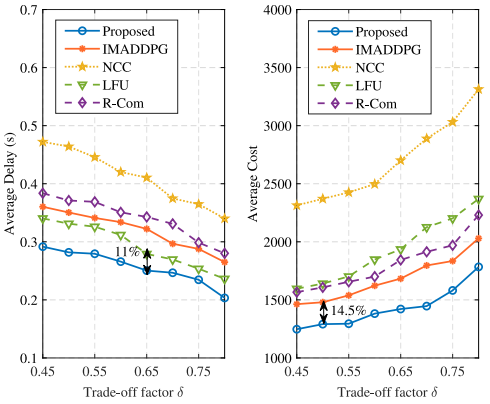


Fig. 8. Average delay and cost versus tradeoff factor δ under different schemes.

the average delay and cost of NCC surpass those of the other schemes, because NCC has no video sharing mechanism among vBSs and each vBS can only acquire videos that are not cached in local cache space from remote video server, resulting in the increasing of total user delay and system operative cost due to higher backhaul-link delay and larger retrieval cost. When K surpasses 20, the R-Com scheme achieves the highest average delay. This is because it can not adjust an effective communication resource allocation strategy as the number of users increases under limited spectrum and power resource, thereby contributing to the fast growth of the transmission delay of access link and average delay of R-Com. Therefore, in the case of limited communication resource and cache capacity at the edge, the MNO should adopt our proposed scheme, which reduces average delay and saves more cost than the other four schemes.

Fig. 8 presents the average delay and cost versus the tradeoff factor δ of five schemes. The increasing of tradeoff factor δ means that the problem focuses more on user delay instead of system operative cost. It is consistent with the results of Fig. 8, where the average delay decreases and average cost arises with the increment of δ. This result indicates the reduction in average delay is accompanied by the increase of the average system operative cost, which proves that there is a tradeoff, controlled by δ, between average delay and cost. It is noteworthy that the average delay and cost of the proposed scheme are always lower than that of the above four schemes. Specifically, the proposed scheme can reduce the average user delay by nearly 11% and save 14.5% in system operative cost than the best two baselines (LFU and IMADDPG). The improvement verifies the proposed scheme's effectiveness in making joint decisions related to cooperative caching, video sharing, computation, and communication allocation under different tradeoff factor δ.

V. Conclusion

In this work, we investigated cooperative edge caching of videos relying on video compression while considering video sharing and delivery. With the goal of minimizing user delay and system operative cost, we formulated an optimization problem while satisfying users' delay requirements by jointly optimizing decisions on video caching, video sharing, communication, and computation resource allocation. To address the problem's complexity, we decomposed it into two subproblems: 1) the joint video sharing and computation resource allocation subproblem and 2) the joint video caching and communication resource allocation subproblem. Then, we proposed a two-layer DRL algorithm, integrating an alternating optimization method at the inner layer and MADDPG at the outer layer, to solve these two subproblems. Simulation results verified the convergence of the proposed algorithm and indicated that the proposed algorithm can reduce the average user delay and system operative cost by distributively making video caching and sharing policies and managing the available communication and computation resources. Additionally, our proposed scheme achieved a superior tradeoff between average users' delay and operational cost compared to benchmark schemes, reducing over 11% user delay and saving 14.5% system operative cost under different tradeoff factor settings.

Appendix
Proof of Lemma 1

Since problem (18) is convex with respect to $\mathbf{F}_b^t$, Karush–Kuhn–Tucker (KKT) conditions are employed to derive the optimal solution of the computation resource allocation at time slot t. The Lagrangian function of problem (18) is written as

$$L\left(F_b^t, \theta_b(t)\right) = \sum_{v\in\mathcal{V}}\sum_{l\in\mathcal{B}\cup o}\sum_{k\in\mathcal{K}_b} y_{v_2}^{b,l}(t) q_{k,v}^b(t) \frac{z_{v_2}\varepsilon_1^{'}\lambda_{k,v}^b}{f_k^b(t)} + \sum_{k\in\mathcal{K}_b} w_f \varepsilon_2^{'} f_k^b(t) + \theta^b(t)\left(\sum_{k\in\mathcal{K}_b} f_k^b(t) - F\right). \tag{32}$$

Accordingly, the KKT conditions of problem (18) are given as follows:

$$\begin{cases} \frac{\partial L(F_b^t,\theta_b(t))}{\partial f_k^b(t)} = w_f\varepsilon_2^{'} + \theta^b(t) - \frac{\sum_{v\in\mathcal{V}}\sum_{l\in\mathcal{B}\cup o} y_{v_2}^{b,l}(t) q_{k,v}^b(t) z_{v_2}\varepsilon_1^{'}\lambda_{k,v}^b}{\left[f_k^b(t)\right]^2} = 0 \\ f_k^b(t) \geq \frac{\sum_{v\in\mathcal{V}}\sum_{l\in\mathcal{B}\cup o} q_{k,v}^b(t) y_{v_2}^{b,l}(t) z_{v_2}\lambda_{k,v}^b}{D_{k,th}^b(t) - \sum_{v\in\mathcal{V}} q_{k,v}^b(t)\left(D_{k,v}^{b,re}(t) + D_{k,v}^{b,ac}(t)\right)} \\ \sum_{k\in\mathcal{K}_b} f_k^b(t) - F = 0 \\ \theta^b(t) \geq 0. \end{cases} \tag{33}$$

The optimal solution of the computation resource allocation satisfies the KKT conditions (33). By solving the above equations and inequalities of (33) and (19) is obtained. In addition, the value of Lagrange multiplier $\theta^b(t)$ is related to the equation $\sum_{k\in\mathcal{K}_b} f_k^b(t) - F = 0$. The proof is completed.

References

[1] X. Jiang, F. R. Yu, T. Song, and V. C. M. Leung, "A survey on multi-access edge computing applied to video streaming: Some research issues and challenges," *IEEE Commun. Surveys Tuts.*, vol. 23, no. 2, pp. 871–903, 2nd Quart., 2021.

[2] "mobile data traffic outlook, mobility report." Ericsson. Accessed: Aug. 1, 2022. [Online]. Available: https://www.ericsson.com/en/reportsand-papers/mobility-report/dataforecasts/mobile-traffic-forecast.

[3] X. Zhu, C. Jiang, L. Kuang, and Z. Zhao, "Cooperative multilayer edge caching in integrated satellite-terrestrial networks," *IEEE Trans. Wireless Commun.*, vol. 21, no. 5, pp. 2924–2937, May 2022.

[4] Y. Han, L. Ai, R. Wang, J. Wu, D. Liu, and H. Ren, "Cache placement optimization in mobile edge computing networks with unaware environment—An extended multi-armed bandit approach," *IEEE Trans. Wireless Commun.*, vol. 20, no. 12, pp. 8119–8133, Dec. 2021.

[5] Z. Chen, Z. Zhou, and C. Chen, "Code caching-assisted computation offloading and resource allocation for multi-user mobile edge computing," *IEEE Trans. Netw. Service Manag.*, vol. 18, no. 4, pp. 4517–4530, Dec. 2021.

[6] Q. Li, A. Nayak, X. Wang, D. Wang, and F. R. Yu, "A collaborative caching-transmission method for heterogeneous video services in cache-enabled terahertz heterogeneous networks," *IEEE Trans. Veh. Technol.*, vol. 71, no. 3, pp. 3187–3200, Mar. 2022.

[7] P. Lin, Q. Song, J. Song, A. Jamalipour, and F. R. Yu, "Cooperative caching and transmission in CoMP-integrated cellular networks using reinforcement learning," *IEEE Trans. Veh. Technol.*, vol. 69, no. 5, pp. 5508–5520, May 2020.

[8] J. Peng, Q. Li, X. Tang, D. Zhao, C. Hu, and Y. Jiang, "A cooperative caching system in heterogeneous edge networks," *IEEE Trans. Mobile Comput.*, early access, Nov. 29, 2023, doi: 10.1109/TMC.2023.3336955.

[9] C. Li, L. Toni, J. Zou, H. Xiong, and P. Frossard, "QoE-driven mobile edge caching placement for adaptive video streaming," *IEEE Trans. Multimedia*, vol. 20, no. 4, pp. 965–984, Apr. 2018.

[10] G. Zhou, L. Zhao, Y. Wang, G. Zheng, and L. Hanzo, "Energy efficiency and delay optimization for edge caching aided video streaming," *IEEE Trans. Veh. Technol.*, vol. 69, no. 11, pp. 14116–14121, Nov. 2020.

[11] M. A. Khan et al., "A survey on mobile edge computing for video streaming: Opportunities and challenges," *IEEE Access*, vol. 10, pp. 120514–120550, 2022.

[12] H. Zhang, R. Xu, Z. Li, D. Wu, and R. Wang, "Resource-aware video delivery in fog radio access networks: A joint QoE and QoS perspective," *IEEE Trans. Veh. Technol.*, vol. 72, no. 5, pp. 6669–6682, May 2023.

[13] D. Ren, X. Gui, and K. Zhang, "Adaptive request scheduling and service caching for MEC-assisted IoT networks: An online learning approach," *IEEE Internet Things J.*, vol. 9, no. 18, pp. 17372–17386, Sep. 2022.

[14] T. Zhang, Z. Wang, Y. Liu, W. Xu, and A. Nallanathan, "Joint resource, deployment, and caching optimization for AR applications in dynamic UAV NOMA networks," *IEEE Trans. Wireless Commun.*, vol. 21, no. 5, pp. 3409–3422, May 2022.

[15] T.-Y. Kuo, M.-C. Lee, J.-H. Kim, and T.-S. Lee, "Quality-aware joint caching, computing and communication optimization for video delivery in vehicular networks," *IEEE Trans. Veh. Technol.*, vol. 72, no. 4, pp. 5240–5256, Apr. 2023.

[16] M. Moghimi, A. Zakeri, M. R. Javan, N. Mokari, and D. W. K. Ng, "Joint radio resource allocation and cooperative caching in PD-NOMA-based HetNets," *IEEE Trans. Mobile Comput.*, vol. 21, no. 6, pp. 2029–2044, Jun. 2022.

[17] P. Lin, Z. Ning, Z. Zhang, Y. Liu, F. R. Yu, and V. C. M. Leung, "Joint optimization of preference-aware caching and content migration in cost-efficient mobile edge networks," *IEEE Trans. Wireless Commun.*, early access, Oct. 17, 2023, doi: 10.1109/TWC.2023.3323464.

[18] A. Asheralieva and D. Niyato, "Game theory and Lyapunov optimization for cloud-based content delivery networks with device-to-device and UAV-enabled caching," *IEEE Trans. Veh. Technol.*, vol. 68, no. 10, pp. 10094–10110, Oct. 2019.

[19] X. Zhong et al., "CL-ADMM: A cooperative-learning-based optimization framework for resource management in MEC," *IEEE Internet Things J.*, vol. 8, no. 10, pp. 8191–8209, May 2021.

[20] W. Xu, Q. Xu, L. Tao, and W. Xiang, "User-assisted base station caching and cooperative prefetching for high-speed railway systems," *IEEE Internet Things J.*, vol. 10, no. 20, pp. 17839–17850, Oct. 2023.

[21] L. Zhang, Y. Sun, Z. Chen, and S. Roy, "Communications-caching-computing resource allocation for bidirectional data computation in mobile edge networks," *IEEE Trans. Commun.*, vol. 69, no. 3, pp. 1496–1509, Mar. 2021.

[22] J. Du, B. Jiang, C. Jiang, Y. Shi, and Z. Han, "Gradient and channel aware dynamic scheduling for over-the-air computation in federated edge learning systems," *IEEE J. Sel. Areas Commun.*, vol. 41, no. 4, pp. 1035–1050, Apr. 2023.

[23] L. Qin, H. Lu, Y. Lu, C. Zhang, and F. Wu, "Joint optimization of base station clustering and service caching in user-centric MEC," *IEEE Trans. Mobile Comput.*, vol. 23, no. 5, pp. 6455–6469, May 2024.

[24] H. Gu, L. Zhao, Z. Han, G. Zheng, and S. Song, "AI-enhanced cloud-edge-terminal collaborative network: Survey, applications, and future directions," *IEEE Commun. Surveys Tuts.*, early access, Dec. 1, 2023, doi: 10.1109/COMST.2023.3338153.

[25] H. Zhang, M. Huang, H. Zhou, X. Wang, N. Wang, and K. Long, "Capacity maximization in RIS-UAV networks: A DDQN-based trajectory and phase shift optimization approach," *IEEE Trans. Wireless Commun.*, vol. 22, no. 4, pp. 2583–2591, Apr. 2023.

[26] J. Du, C. Jiang, J. Wang, Y. Ren, and M. Debbah, "Machine learning for 6G wireless networks: Carrying forward enhanced bandwidth, massive access, and ultrareliable/low-latency service," *IEEE Veh. Technol. Mag.*, vol. 15, no. 4, pp. 122–134, Dec. 2020.

[27] H. Zhou, Z. Zhang, Y. Wu, M. Dong, and V. C. M. Leung, "Energy efficient joint computation offloading and service caching for mobile edge computing: A deep reinforcement learning approach," *IEEE Trans. Green Commun. Netw.*, vol. 7, no. 2, pp. 950–961, Jun. 2023.

[28] X. Hou, J. Wang, C. Jiang, Z. Meng, J. Chen, and Y. Ren, "Efficient federated learning for metaverse via dynamic user selection, gradient Quantization and resource allocation," *IEEE J. Sel. Areas Commun.*, vol. 42, no. 4, pp. 850–866, Apr. 2024.

[29] J. Luo, J. Song, F.-C. Zheng, L. Gao, and T. Wang, "User-centric UAV deployment and content placement in cache-enabled multi-UAV networks," *IEEE Trans. Veh. Technol.*, vol. 71, no. 5, pp. 5656–5660, May 2022.

[30] D. Bertsekas, *Dynamic Programming and Optimal Control*, vol. 1. Amsterdam in The Netherlands: Athena Sci., 2012.

[31] W. Liu, B. Li, W. Xie, Y. Dai, and Z. Fei, "Energy efficient computation offloading in aerial edge networks with multi-agent cooperation," *IEEE Trans. Wireless Commun.*, vol. 22, no. 9, pp. 5725–5739, Sep. 2023.

[32] Z. Chen, W. Yi, A. S. Alam, and A. Nallanathan, "Dynamic task software caching-assisted computation offloading for multi-access edge computing," *IEEE Trans. Commun.*, vol. 70, no. 10, pp. 6950–6965, Oct. 2022.

[33] J. Fang, S. Chen, and M. Cai, "Mobile edge data cooperative cache admission based on content popularity," in *Proc. IEEE Edge Comput. Conf. (EDGE)*, 2021, pp. 111–118.

[34] Z. Chen, W. Yi, Y. Liu, and A. Nallanathan, "Robust federated learning for unreliable and resource-limited wireless networks," *IEEE Trans. Wireless Commun.*, early access, Feb. 23, 2024, doi: 10.1109/TWC.2024.3366393.

[35] Z. Chen, W. Yi, and A. Nallanathan, "Exploring Representativity in device scheduling for wireless federated learning," *IEEE Trans. Wireless Commun.*, vol. 23, no. 1, pp. 720–735, Jan. 2024.

[36] G. Qiao, S. Leng, S. Maharjan, Y. Zhang, and N. Ansari, "Deep reinforcement learning for cooperative content caching in vehicular edge computing and networks," *IEEE Internet Things J.*, vol. 7, no. 1, pp. 247–257, Jan. 2020.

[37] S. Diamond and S. Boyd, "CVXPY: A python-embedded modeling language for convex optimization," *J. Mach. Learn. Res.*, vol. 17, no. 1, pp. 2909–2913, 2016.

[38] R. Lowe, Y. I. Wu, A. Tamar, J. Harb, O. Pieter Abbeel, and I. Mordatch, "Multi-agent actor-critic for mixed cooperative-competitive environments," in *Proc. 31st Adv. Neural Inf. Process. Syst.*, 2017, pp. 6382–6393.

[39] Z. Chen, W. Yi, A. Nallanathan, and J. Chambers, "Distributed digital twin migration in multi-tier computing systems," *IEEE J. Sel. Topics Signal Process.*, early access, Jan. 26, 2024, doi: 10.1109/JSTSP.2024.3359009.

[40] C. Zheng, S. Liu, Y. Huang, and T. Q. S. Quek, "Privacy-preserving federated reinforcement learning for popularity-assisted edge caching," in *Proc. IEEE Glob. Commun. Conf. (GLOBECOM)*, 2021, pp. 1–6.

[41] K. Liang et al., "Customizable and robust Internet of Robots based on network slicing and digital twin," *IEEE Netw.*, early access, Mar. 21, 2024, doi: 10.1109/MNET.2024.3375503.

[42] T. P. Lillicrap et al., "Continuous control with deep reinforcement learning," 2015, *arXiv:1509.02971*.

[43] T. Zhang, Y. Wang, W. Yi, Y. Liu, C. Feng, and A. Nallanathan, "Two time-scale caching placement and user association in dynamic cellular networks," *IEEE Trans. Commun.*, vol. 70, no. 4, pp. 2561–2574, Apr. 2022.

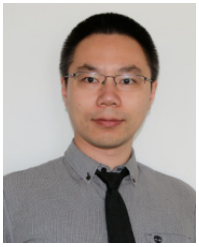
Bingjie Zhu is currently pursuing the Ph.D. degree with the School of Telecommunication Engineering, Xidian University, Xi'an, China.

Her research interests include edge computing/caching, network slicing, machine learning for communications, and open RAN.

Liqiang Zhao (Member, IEEE) received the B.Sc. degree in electrical engineering from Shanghai Jiao Tong University, Shanghai, China, in 1992, and the M.Sc. degree in communications and information systems and the Ph.D. degree in information and communications engineering from Xidian University, Xi'an, China, in 2000 and 2003, respectively.

From 1992 to 2005, he was a Research Engineer with the 20th Research Institute, Chinese Electronics Technology Group Corporation, Beijing, China. From 2005 to 2007, he was an Associate Professor with the State Key Laboratory of Integrated Service Networks, Xidian University, Xi'an, China. He was appointed as a Marie Curie Research Fellow with the Centre for Wireless Network Design, University of Bedfordshire, Luton, U.K., in June 2007, to conduct research in the GAWIND Project funded under EU FP6 HRM Program. His activities focused on the area of automatic wireless broadband access network planning and optimization. Since June 2008, he has returned Xidian University, first as an Associate Professor and later as a Professor. He has hosted/participated many national research projects, such as the National Natural Science Foundation, the 863 Program, and the National Science Technology Major Projects, and several international research projects, including the EU FP6, and FP7 plans for International Cooperation Exchange Projects, and some research projects from companies, such as Huawei. He has more than 100 published in authorized academic periodicals both in and abroad and in international science conferences, wherein 30 of which are retrieved in SCI, and more than 70 of them are EI indexed, and six national invention patents. His current research interests include mobile communication systems, spread spectrum communications, WiMAX, WLAN, wireless sensor networks, broadband wireless communications, and space communications.

Dr. Zhao was awarded by the Program for New Century Excellent Talents in University, Ministry of Education, China, in 2008.



Wenqiang Yi (Member, IEEE) received the Ph.D. degree in electrical engineering from Queen Mary University of London, London, U.K., in 2020.

He is currently an Assistant Professor with the School of Computer Science and Electronic Engineering, University of Essex, Colchester, U.K., since 2023. From 2020 to 2023, he was a Postdoctoral Researcher with Communication Systems Research Group, School of Electronic Engineering and Computer Science, Queen Mary University of London. His research interests include AI in wireless communications, RF sensing, and stochastic geometry.

Dr. Yi received the Exemplary Reviewer of the IEEE COMMUNICATION LETTERS and the IEEE TRANSACTIONS ON COMMUNICATIONS in 2019 and 2020. He serves as an Editor for IEEE OPEN JOURNAL OF THE COMMUNICATIONS SOCIETY, in the area of Big Data and Machine Learning for Communications. He served as the Symposium Chair for reconfigurable intelligent surfaces at IEEE ICCT. He has served as a TPC Member for many IEEE conferences, such as GLOBECOM and ICC. He also served as the Secretary for the Special Interest Group on Next Generation Multiple Access (NGMA) by the SPCC Technical Committee and the Emerging Technologies Initiatives on NGMA by the Emerging Technologies Committee till 2022.

Zhixiong Chen (Graduate Student Member, IEEE) received the B.S. and M.S. degrees from Chongqing University, Chongqing, China, in 2018 and 2021, respectively. He is currently pursuing the Ph.D. degree with the School of Electronic Engineering and Computer Science, Queen Mary University of London, London, U.K.

His research interests include machine learning for wireless networks and signal processing, reinforcement learning, and wireless federated learning.

Arumugam Nallanathan (Fellow, IEEE) received the B.Sc. (First Class) degree in electrical engineering from the University of Peradeniya, Peradeniya, Sri Lanka, in 1991, the CPGS degree in electrical engineering from the University of Cambridge, Cambridge, U.K., in 1995, and the Ph.D. degree in electrical engineering from the University of Hong Kong, Hong Kong, in 2000.

He is a Professor of Wireless Communications and the Head of the Communication Systems Research Group with the School of Electronic Engineering and Computer Science, Queen Mary University of London, London, U.K., since September 2017. He was with the Department of Informatics, King's College London, London, from December 2007 to August 2017, where he was the Professor of Wireless Communications from April 2013 to August 2017 and a Visiting Professor from September 2017. He published nearly 700 technical papers in scientific journals and international conferences. He was an Assistant Professor with the Department of Electrical and Computer Engineering, National University of Singapore, Singapore, from August 2000 to December 2007. His research interests include artificial intelligence for wireless systems, beyond 5G wireless networks, and Internet of Things (IoT).

Dr. Nallanathan is a co-recipient of the Best Paper Awards presented at the IEEE International Conference on Communications 2016 (ICC'2016), IEEE Global Communications Conference 2017 (GLOBECOM'2017) and IEEE Vehicular Technology Conference 2018 (VTC'2018). He is also a co-recipient of IEEE Communications Society Leonard G. Abraham Prize in 2022. He is an IEEE Distinguished Lecturer. He has been selected as a Web of Science Highly Cited Researcher in 2016, 2022, and 2023. He is a Senior Editor for IEEE WIRELESS COMMUNICATIONS LETTERS. He was an Editor for IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, IEEE TRANSACTIONS ON COMMUNICATIONS, IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, and IEEE SIGNAL PROCESSING LETTERS. He received the IEEE Communications Society SPCE Outstanding Service Award 2012 and IEEE Communications Society RCC Outstanding Service Award 2014. He served as the Chair for the Signal Processing and Communication Electronics Technical Committee of IEEE Communications Society and Technical Program Chair and the member of Technical Program Committees in numerous IEEE conferences.