Edge-Assisted Short Video Sharing With Guaranteed Quality-of-Experience

Fahao Chen, Peng Li, *Senior Member, IEEE*,
Deze Zeng, *Member, IEEE*, and Song Guo, *Fellow, IEEE*

Abstract—As a rising star of social apps, short video apps, e.g., TikTok, have attracted a large number of mobile users by providing fresh and short video contents that highly match their watching preferences. Meanwhile, the booming growth of short video apps imposes new technical challenges on the existing computation and communication infrastructure. Traditional solutions maintain all videos on the cloud and stream them to users via contend delivery networks or the Internet. However, they incur huge network traffic and long delay that seriously affects users' watching experiences. In this article, we propose an edge-assisted short video sharing framework to address these challenges by caching some highly preferred videos at edge servers that can be accessed by users via high-speed network connections. Since edge servers have limited computation and storage resources, we design an online algorithm with provable approximation ratio to decide which videos should be cached at edge servers, without the knowledge of future network quality and watching preferences changes. Furthermore, we improve the performance by jointly considering video fetching and user-edge association. Extensive simulations are conducted to evaluate the proposed algorithms under various system settings, and the results show that our proposals outperform existing schemes.

Index Terms—Short video, edge computing, online algorithm design, mobile networks

1 Introduction

Short video apps, e.g., TikTok [1], Bermi [2] and Kwai [3], have gained great popularity in recent years. These apps allow users to create and publish short videos of 15-60 seconds. Meanwhile, users can watch videos created by others via these apps. It has been reported [4] that TikTok has more than 500 million active users worldwide and it is the most downloaded app on the Apple App Store in 2019, beating popular apps, e.g., Youtube, Instagram and Facebook. More than 1 million videos are viewed every day via these short video apps.

Short video apps are different from traditional video sharing platforms, e.g., Youtube, due to several unique features. The first one is the access pattern. In traditional platforms, video length is usually from several minutes to hours, and users can freely select preferred videos to watch on their mobile devices or desktops. In contrast, short video apps provide videos of 15-50 seconds and push them to users by the recommendation algorithm that exploits users' preferences from historical watching records. When the current playing video finishes, the apps automatically start to play the next one. If users dislike the current playing video, they can quickly switch to the next one by flicking the smartphone screen. Second, the video length also affects the system design. Short videos are usually less than 1 minutes and users could watch a large number of videos in 1 hour, which suggests that video popularity changes quickly. Some recent trance analysis shows that the number of accesses of some videos decreases by 10 times within 1 hour [5]. Third, the numbers of videos served by two systems are different, which also leads to different design policies. If we optimize the placement of short videos by following the ideas of long video systems [6], [7], [8], [9], [10], [11], i.e., making caching decisions for each video, there would be a large number of variables and the search space of the formulated problem would be huge. It makes the problem solving intractable. Finally, short video apps mainly reside on mobile devices, e.g., smartphones and tablets, whose network connection is unstable. Therefore, the effect of user mobility should be considered in the video sharing system design.

Traditionally, video contents are maintained at cloud data centers [12], [13] and they are delivered to users via content delivery networks (CDNs) [14], [15] or the Internet [16]. Since cloud data centers are usually far from mobile users, the traditional architecture may incur high latency in video replaying, which severely affects user experiences. Especially, users of short video apps are sensitive to latency because video lengths are small and users may frequently switch to new ones if the current playing one is not preferred. In addition, traditional CDNs and caching policies cannot efficiently handle short video sharing due to its quickly changed video popularity and user mobility.

In this paper, we propose an edge-assisted short video sharing framework to improve users' quality-of-experience (QoE). As illustrated in Fig. 1, some modest-size edge servers are deployed close to mobile users, and they fetch video contents from the cloud according to users' watching preferences.

- *Fahao Chen and Peng Li are with the University of Aizu, Aizuwakamatsu 965-8580, Japan. E-mail: {m5232105, pengli}@u-aizu.ac.jp.*
- *Deze Zeng is with the China University of Geoscience, Wuhan, Hubei 430074, China. E-mail: deze@cug.edu.cn.*
- *Song Guo is with the Department of Computing, Hong Kong Polytechnic University, Kowloon, Hong Kong. E-mail: song.guo@polyu.edu.cn.*

Manuscript received 25 May 2020; revised 25 Feb. 2021; accepted 16 Mar. 2021. Date of publication 23 Mar. 2021; date of current version 8 Mar. 2023.
(Corresponding author: Peng Li.)
Recommended for acceptance by A. Zomaya.
Digital Object Identifier no. 10.1109/TCC.2021.3067834

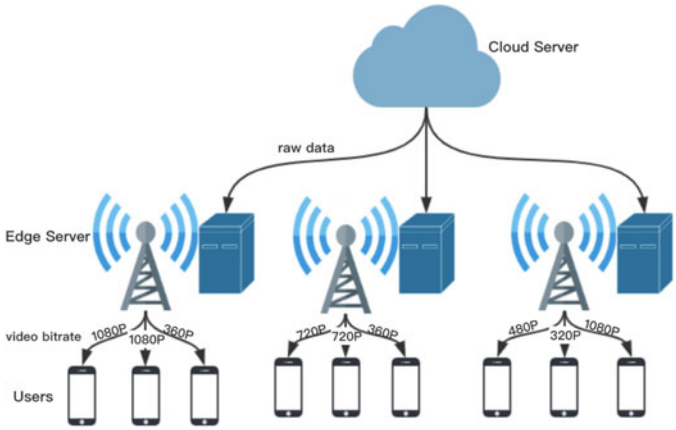


Fig. 1. System architecture.

The videos downloaded from cloud data centers are raw data, and they are then encoded and compressed into several versions with different bitrates by edge servers. These versions are stored at local storage of edge servers and then pushed to mobile users within the service range. Mobile users can enjoy video versions whose bitrates match their current network speeds.

Although edge-assisted short video sharing is promising, we are facing several critical challenges unsolved by existing work [17], [18], [19], [20]. First, existing work cannot fully exploit user preferences in content caching at edge servers. Since each edge server has limited storage and computational capability, we need to decide which videos should be cached and encoded. There are many research efforts on content caching at edge servers, e.g., [21], but they mainly focus on caching hot contents requested by most of the users. However, in short video services, users have different preferences on videos, according to their hobbies, cultures, educational backgrounds, lifestyles and so on. Even though a video have been seldom watched, it should be cached if it is highly preferred by users. Second, the short-video sharing system is highly dynamic because of user mobility and watching preference changes. Users enjoy short videos on mobile devices, e.g., smartphones or tablets, and their quality of network connection to edge servers changes as they move around. Meanwhile, users' watching preferences change quickly in short video apps. Therefore, each edge server needs to adjust its cached contents according to the latest watching preferences of users within its service range.

In this paper, we address the above challenges by proposing online algorithms with the objective of maximizing the utility that is defined as the total preferences of videos watched by users minus the video type replacing cost across time slots. We first design a video fetching algorithm for edge servers by supposing that the set of mobile users connected to each edge server is given. The basic idea of our algorithm is to solve a well-designed optimization problem that considers potential future video replacing in each time slot. The competitive ratio of the proposed algorithm is derived. After that, we show that the decision of which user should connect to which edge server, referred to as user-edge association, can affect the video fetching policy and can further improve the performance. Therefore, we design an online algorithm by jointly optimizing video fetching and user-edge association. The main contributions of this paper are summarized as follows.

- We propose an edge-assist short video sharing framework by exploiting the computation and communication resources at edge servers. To maximize the total utility, we design an online algorithm to decide how much of each type of videos should be fetched by edge servers. The performance of this algorithm is theoretically guaranteed by a provable competitive ratio.
- We show that user-edge association can affect video fetching decisions. To further improve the performance, we design a fast online algorithm that jointly considers video fetching and user-edge association.
- We conduct extensive simulations to evaluate our proposed algorithms under various system settings, and results show that they significantly outperform existing work.

The rest of this paper is organized as follows. Some recent related work is reviewed in Section 2. We present our system model in Section 3. The online algorithm design for video fetching is presented in Section 4. An extension of jointly optimizing video fetching and user-edge association is presented in Section 5. The performance evaluation is given in Section 6. Section 7 concludes this paper finally.

2 Related Work

Video Streaming. The video streaming get a great success in recent years and have attracted many research attentions. Some main research directions include video content extraction [22], [23], video quality improvement [24], video caching [25] and so on. Shao *et al.* [26] have extracted destination images of the Forbidden City and the Badaling Great Wall on Douyin platform and compare the result with textual blogs to verify the validity of visual destination image extraction method based on video-AI. Modern video codecs have been evaluated in [27]. David *et al.* [28] have proposed an automated recording platform that can balance the cost of mobility and video quality. A new algorithm for queuing has been proposed in [29] to reduce the decrease of the quality for the transfer of video streams to users. In [30], Li *et al.* have proposed a window-based rate control scheme to reduce the system latency while optimizing the quality of videos. Ghadiyaram *et al.* [31] have presented a new mobile video database called LIVE Mobile Stall Video Database-II to assist the research on improving Quality of Experience (QoE) prediction models. Different from these works, we focus on not only the video content and quality but also the network situations of users. We choose the videos according to the users' preferences and then choose the most suitable bitrate version by collecting the users' network situations. DeepCast [32] has been proposed to improve personalized QoE via deep reinforcement learning in crowdcast. However, they cannot provide theoretical performance guarantee and do not consider the user-edge association.

Edge Computing. Edge computing emerges as a promising paradigm for solving the latency problem of traditional cloud computing. A number of modest-size edge servers are deployed close to mobile users, so that they can provide quickly response. Edge computing has been applied in many applications, but it has been seldom studied for short video sharing. A novel technical framework for deploying latency-sensitive Internet-of-Things(IoT) applications on edge devices has been proposed in [33]. Nikouei *et al.* [34] have proposed a novel model to oversee human objects in video frames by using a lightweight Convolutional Neural Network (L-CNN) on edge devices. Mobile edge computing (MEC) can take the computing resources close to the mobile devices, and hence Li *et al.* [35] have evaluated the heart rate detection APP in MEC environment and demonstrated the best performance can be achieved. Sridhar *et al.* [36] have identified the voice-drive interaction pipelines based on weak-edge devices get a lower response latency than cloud services.

Edge-Based Video Sharing. To reduce both backhaul traffic and video content access latency, several works have been proposed to use video content caching in the mobile edge computing (MEC) servers. Tran *et al.* [6] have proposed a collaborative joint cache strategy for video on demand applications on Radio Access Network (RAN), where MEC servers collaboratively cache the video contents. By considering both supporting adaptive bitrate (ABR)-video streaming and the video popularity, They have also proposed the ABR-aware proactive cache placement to minimize the video retrieval cost on the collaborative MEC video cache system [7]. Baccour *et al.* [8] have proposed a collaborative video cache system that focuses on the video chunks instead of the total video content to optimize the cache resource utilization. Based on this work, Baccour *et al.* [9] have combined device-to-device (D2D) connections to provide data offloading in users' devices, which further optimize the use of cellular and backhaul bandwidth. Xe *et al.* [10] have combined video caching and ABR streaming technology together and formulated the caching problem with the Stackelberg game to deal with the caching resources allocation, further enhancing the video service. Zhang *et al.* [11] have considered both the cache placement problem and the user-BS association problem. They propose a linearization and rounding algorithm to support multiple bitrates video streaming and utilize both storage and computing resources in MEC efficiently. To ensure the video segments that the user requires could be cached in time, Huang *et al.* [37] have refined the MEC cache based on video popularity, content importance, and user playback status. Moreover, Liang *et al.* [38] have proposed a joint optimization mechanism of both bandwidth configuration and adaptive video streaming with software-defined (SDN) wireless networks to reduce service delay and improve quality of experience (QoE). In addition, Huang *et al.* [39] have considered caching policy, power allocation, user-BS association, and adaptive video streaming and proposed a joint caching scheme to improve both the system spectrum efficiency and QoE, however, a greater burden on backhaul traffic is also caused.

However, although edge-based video sharing has been extensively studied by the above works, their techniques and algorithms cannot be directly applied to solve the short video sharing for several reasons. First, due to the unique access pattern, we can not formulate the short video caching problem based on video requests like long video systems do [6], [7], [8], [9], [10], [11]. Second, the assumption that each user only requires videos from the nearest server in on-demand video systems [6], [7], [8], [9] is not suitable for short video caching system. Third, in traditional video caching system, they focus on minimizing the system cost based on the Least Recent Used (LRU) or video popularity. But short video systems aim to push videos to users to maximize the users' preferences. Furthermore, traditional video systems make caching decision for each video [6], [7], [8], [9], [10], [11], [37], [38], however, it is time-costly to follow a similar idea in the systems that include massive amount of short videos. Finally, existing works lack sufficient theoretical analysis to demonstrate the effectiveness of these proposed algorithms.

3 System Model

We consider a short video sharing system consisting of a cloud, a set E of edge servers, and a set U of mobile users. The storage capacity of the edge server $e \in E$ is S_e. The cloud maintains a number of videos that can be classified into $|V|$ types, where V denotes the type set. Since the quantity of short video is huge, we make video recommendation based on types (e.g., games, cooking, and dances), instead of individual ones, to reduce complexity. Due to the bandwidth limitation of existing Internet infrastructure, mobile users cannot always enjoy high-speed network connection with the cloud, which incurs video playing lagging or choppy. In order to improve the quality-of-experience of video watching, we let edge servers fetch some videos from cloud and cache them at local storage, so that mobile users can enjoy smooth and high-quality video playing via the high-speed network connection to edge servers. As shown in Fig. 1, when an edge server decides to cache some videos, it downloads raw video data and encodes them into several versions with different bitrates $R = \{r^0, r^1, r^2, \ldots r^*\}$. The videos belonging to the same type have similar sizes and the average size of type-v videos with bitrate r is denoted by s_v^r. We have $s_v^r > s_v^{r'}$ if $r > r'$. For a type-v video, its computational cost of encoding and communication cost of downloading is β_v.

We consider a discrete-time model that divides continuous time into multiple time slots of length τ. Note that the length τ is a given system parameter that depends on the estimation accuracy of network bandwidths and watching preferences. We let $B_u^e(t)$ denote the network bandwidth between the user $u \in U$ and the edge server $e \in E$ in time slot t. We assume that mobile users maintain stable network bandwidth $B_u^e(t)$ within each time slot, but it may change across time slots due to user mobility. The user u's preference on the type-v videos with bitrate $r \in R$ in time slot t is denoted by $\omega_{uv}^r(t)$. Note that watching preferences may also change across time slots. Usually, we have $\omega_{uv}^r(t) > \omega_{uv}^{r'}(t)$ if $r > r'$. The value of $\omega_{uv}^r(t)$ is computed by the given recommendation algorithm according to user's watching history, hobby and so on. We omit the details of the watching recommendation algorithm design because it is orthogonal with our video sharing problem studied in this paper. For clarity, we summarize the main notations used in this paper in Table 1.

4 Video fetching Algorithm Design

In this section, we first present the problem description of short video fetching. Then, we give the online algorithm design as well as its theoretical performance analysis.

4.1 Problem Description

In order to improve QoE of users, an intuitive idea is to let edge servers fetch and cache videos that are highly

TABLE 1
Notations

E	The set of edge servers.
V	The set of video types.
U	The set of mobile users.
T	The set of time slots.
R	The set of video bitrates.
s_v^r	The size of video v with bitrate r.
S_e	The storage capacity of edge server $e \in E$.
$B_u^e(t)$	The network bandwidth between user $u \in U$ and edge server $e \in E$ in time slot t.
$\omega_{uv}^r(t)$	The preference of user u for video v with bitrate r in time slot t.
τ	The length of each time slot
U_e	The set of users associated with edge server $e \in E$.
$x_v(t)$	A variable representing the number of type-v videos fetched in time slot t.
β_v	The computation cost of encoding and communication cost of downloading for a type-v video.
ϵ	The algorithm parameter.
η	The algorithm parameter based on ϵ and $\|V\|$.
β	$\min_v \beta_v$
ω_{min}	$\min_{u,v,r,t} \omega_{uv}^r(t)$
$\lambda_v^t, \mu^t, \theta_v^t$	The Lagrangian multipliers.
$x_v^e(t)$	A variable indicating the number of type-v videos are cached by edge server $e \in E$ in time slot t.
$y_{uv}^{re}(t)$	A variable indicating the number of type-v videos are pushed to user $u \in U$ with bitrate $r \in R$ from edge server $e \in E$ in time slot t.
$\phi_u^e(t)$	A variable representing whether the user $u \in U$ is associated with the edge server $e \in E$ in time slot t.
D_e	The maximum number of users associated with edge server $e \in E$
Δ	The number of computing rounds in the VF-UEA

preferred by users. Due to user mobility and changes of watching preferences, we need to replace some cached videos with the ones of higher preferences in each time slot. Unfortunately, video replacement incurs the communication cost of downloading videos from cloud, as well as the computational cost of encoding them into different versions. Therefore, in this section, we design an algorithm to decide how many videos of each type $v \in V$ should be cached in each time slot, so that we can make a good tradeoff between the total watching preferences and the replacement cost.

In this section, we focus on algorithm design for the scenario that the set of users associated with each edge server is already given. This scenario is common because we can simplify the system design by decoupling the modules of video fetching and user-edge association. For example, we can use the algorithm designed in this section for video fetching, and conduct user-edge association by letting each user connect to the edge server with the fastest connection. Such a kind of user-edge association has been widely adopted due to its simplicity. In the next section, we will show that it is not always the best solution and study the joint optimization of video caching and user-edge association.

We define a variable $x_v(t)$ to denote the number of type-v videos fetched in time slot t. We let $\mathbf{X}(t) = \{x_0(t), x_1(t), \ldots, x_{v-1}(t)\}$. Given a solution of $\mathbf{X}(t)$, the edge server always sends users videos whose bitrates match their current bandwidths. In addition, the types with higher watching preferences are sent with higher priorities. We model the system benefit as a non-decreasing convex function $f(\cdot)$, and we assume it is continuously differentiable. Hence, our target problem can be formulated as:

$$\max \sum_{v,t} f(\mathbf{X}(t)) - \beta_v [x_v(t) - x_v(t-1)]^+$$

$$\sum_{v,r} s_v^r x_v(t) \le S_e, \forall t \in T; \tag{1}$$

$$x_v(t) \ge 0, \forall v \in V, t \in T. \tag{2}$$

The objective function denote the total utility across all time slots, which contains two parts: the benefit of watched videos and replacement cost. Note that $[x]^+ = \max\{x, 0\}$. The total size of cached videos cannot exceed the storage capacity of edge servers, as shown in (1). The number of fetched videos cannot be negative, which is shown in (2).

4.2 Algorithm Design

Solving the above linear programming problem needs the information of watching preferences of users and their network bandwidths over all time slots. Unfortunately, it would be difficult to have these information in practice because of uncertainty of user mobility and changes of watching preferences. Therefore, we design an online algorithm that can decide which videos should be fetched in each time slot based on observed information.

Before presenting the details of online algorithm design, we simplify the above formulation as follows.

$$\max \sum_{v,t} f(\mathbf{X}(t)) - \beta_v z_v(t)$$

$$z_v(t) \ge x_v(t) - x_v(t-1), \forall v \in V, t \in T; \tag{3}$$

$$z_v(t) \ge 0, \forall v \in V, t \in T; \text{and } (1). \tag{4}$$

Based on this simplified formulation, we design our online algorithm VF whose pseudo codes are shown in Algorithm 1. In each time slot, instead of solving the original problem, we solve a problem with a revised objective function that considers potential future video replacement cost. Specifically, we first initialize the adjusted system parameters ϵ and η, as well as the caching amount for $v-$type videos x_v at time slot 0. Then in each time slot $t > 0$, we obtain the edge servers' storage capacity S_e and the computation cost for encoding a $v-$type video, which is shown in line 1. Next, we solve the objective function (5), which includes the total preferences and the regularized switching costs. We regularize the switching costs by using the relative entropy plus a linear term parameterized by ϵ. The solution in each time slot is determined greedily and independently of time slots prior to $t-1$. Since (5) is a continuous convex function and (1) bounds the solution in a convex set, the problem in each time slot can be solved in polynomial time by standard convex optimization techniques, such as interior-point methods. Finally, we obtain the solution $\tilde{x}_v(t)$ in the current time slot t in the line 1 and move to the next time slot optimizations.

Algorithm 1. Video Fetching With Given User-Edge Association (VF)

```
Require: $\epsilon > 0$ and $\eta = \ln(1+|V|/\epsilon)$
Ensure: $\tilde{x}_v(t)$
1: Initialize $x_v(0) = 0, \forall v \in V$;
2: for each time slot $t$ do
3:     obtain $S_e$ and $\beta_v, \forall v \in V$;
4:     Solve the following optimization problem:
           $\max f(\mathbf{X}(t)) - \frac{1}{\eta}\sum_v \beta_v \Big[\Big(x_v(t) + \frac{\epsilon}{|V|}\Big) \cdot \ln\Big(\frac{x_v(t)+\epsilon/|V|}{\tilde{x}_v(t-1)+\epsilon/|V|}\Big) - x_v(t)\Big]$   (5)
           subject to: constraint (1);
       Extract the values of $x_v(t)$ from the solutions and denote it by $\tilde{x}_v(t)$;
5: end for
```

For online algorithm designs, a commonly used theoretical performance metric is competitive ratio, which describes how close the proposed online algorithms approximate the optimal solution with full information over all time slots. The formal definition is as follows.

Definition 1. *An online algorithm with solution ALG is ρ-competitive if $\rho \leq \frac{ALG}{OPT}$, where OPT is the optimal solution given all information across time slots.*

With the above definition, we have the following theorem for VF.

Theorem 1. *VF is $\left(1-\frac{\beta-C}{\omega_{min}-C}\right)$-competitive, where $C = \frac{\sum_{v,t}(\tilde{x}_v(t)+\epsilon/|V|)\frac{\beta_v\tilde{x}_v(t)}{\eta}\ln\left(\frac{\tilde{x}_v(t)+\epsilon/|V|}{\tilde{x}_v(t-1)+\epsilon/|V|}\right)}{|V||T|}$, $\omega_{min} = \min_{u,v,r,t}\omega^r_{uv}(t)$, and $\beta = \min_{v,t}\beta_v$.*

Proof. To prove this theorem, we first write the Lagrangian function of the original formulation as follows.

$$
\begin{aligned}
&L(\lambda^t_v, \mu^t, \theta^t_v, x_v(t), z_v(t))\\
=&\sum_{v,t}[f(\mathbf{X}(t)) - \beta_v z_v(t)]\\
&+\sum_{v,t}\lambda^t_v(z_v(t) - x_v(t) + x_v(t-1))\\
&+\sum_t \mu^t(S_e - \sum_{v,r}s^r_v x_v(t)) + \sum_{v,t}\theta^t_v z^t_v\\
=&\sum_{v,t,r}[f(\mathbf{X}(t)) + (-\lambda^t_v + \lambda^{t+1}_v - \mu^t s^r_v)x_v(t)]\\
&+\sum_{v,t}(\lambda^t_v - \beta_v + \theta^t_v)z_v(t) + \sum_t S_e\mu^t.
\end{aligned}
$$

Therefore, the dual function can be written as

$$
\begin{aligned}
D(\lambda^t_v, \mu^t, \theta^t_v) &= \max_{x_v(t), z_v(t)} L(\lambda^t_v, \mu^t, x_v(t), z_v(t))\\
&= \max_{x_v(t)}\sum_{v,t,r}[f(\mathbf{X}(t)) + (-\lambda^t_v + \lambda^{t+1}_v - \mu^t s^r_v)x_v(t)]\\
&\quad + \max_{z_v(t)}\sum_{v,t}(\lambda^t_v - \beta_v + \theta^t_v)z_v(t) + \sum_t S_e\mu^t.
\end{aligned}
$$

From weak duality, we can easily get:

$$
D_p = \max_{x_v(t), z_v(t)} L(\lambda^t_v, \mu^t, \theta^t_v, x_v(t), z_v(t)) \geq O_p,
$$

where O_p is the unconstrained form of the original problem while D_p is its duality form. Thus we can get:

$$
\frac{ALG}{O_p} \geq \frac{ALG}{D_p}
$$

Since (5) is convex, by given $\tilde{\mu}^t$ in **VF**, we have the following equations according to the KKT conditions:

$$
\begin{gathered}
(f(\tilde{\mathbf{X}}(t)))' - \frac{\beta_v}{\eta}\ln\left(\frac{\tilde{x}_v(t)+\epsilon/|V|}{\tilde{x}_v(t-1)+\epsilon/|V|}\right) - \tilde{\mu}^t s^r_v = 0,\\
\forall u \in U, \forall v \in V, \forall r \in R, \forall t \in T,
\end{gathered} \tag{6}
$$

$$
\tilde{\mu}^t\left(S_e - \sum_{v,r}s^r_v\tilde{x}_v(t)\right) = 0, \forall t \in T. \tag{7}
$$

By setting $\lambda^t_v = \frac{\beta_v}{\eta}\ln\left(\frac{1+\epsilon/|V|}{\tilde{x}_v(t-1)+\epsilon/|V|}\right)$, $\mu^t = \tilde{\mu}^t$, and $\theta^t_v = 0$, we can have $\beta_v > \lambda^t_v$. Hence the dual function can be written as follows:

$$
\begin{aligned}
&D(\lambda^t_v, \mu^t, \theta^t_v)\\
&= \max_{x_v(t)}\sum_{v,t,r}\left[f(\mathbf{X}(t)) - \left(\frac{\beta_v}{\eta}\ln\left(\frac{\tilde{x}_v(t)+\epsilon/|V|}{\tilde{x}_v(t-1)+\epsilon/|V|}\right)\right.\right.\\
&\quad \left.\left. + \tilde{\mu}^t s^r_v\right)x_v(t)\right] + \sum_t \tilde{\mu}^t S_e + \max_{z_v(t)}\sum_{v,t}\left[\left(-\beta_v\right.\right.\\
&\quad \left.\left. + \frac{\beta_v}{\eta}\ln\left(\frac{1+\epsilon/|V|}{\tilde{x}_v(t-1)+\epsilon/|V|}\right)\right)z_v(t)\right]\\
&= \sum_{v,t,r}\left[f(\tilde{\mathbf{X}}(t)) - \frac{\beta_v}{\eta}\ln\left(\frac{\tilde{x}_v(t)+\epsilon/|V|}{\tilde{x}_v(t-1)+\epsilon/|V|}\right)\tilde{x}_v(t)\right].
\end{aligned}
$$

Due to the weak duality, the competitive ratio can be derived as follows.

$$\frac{ALG}{OPT} \geq \frac{ALG}{D_p}$$

$$= \frac{\sum_{v,t,r}[f(\tilde{\mathbf{X}}(t)) - \beta_v \tilde{z}_v(t)]}{\sum_{v,t,r}\left[f(\tilde{\mathbf{X}}(t)) - \frac{\beta_v}{\eta}\ln\left(\frac{\tilde{x}_v(t)+\epsilon/|V|}{\tilde{x}_v(t-1)+\epsilon/|V|}\right)\tilde{x}_v(t)\right]}$$

$$\geq 1 - \frac{\sum_{v,t}\left[\beta_v \tilde{z}_v(t) - \frac{\beta_v}{\eta}\ln\left(\frac{\tilde{x}_v(t)+\epsilon/|V|}{\tilde{x}_v(t-1)+\epsilon/|V|}\right)\tilde{x}_v(t)\right]}{\sum_{v,t}\left[f(\tilde{\mathbf{X}}(t)) - \frac{\beta_v}{\eta}\ln\left(\frac{\tilde{x}_v(t)+\epsilon/V}{\tilde{x}_v(t-1)+\epsilon/V}\right)\tilde{x}_v(t)\right]}$$

$$\geq 1 - \frac{\sum_{v,t}\left[\beta_v \tilde{x}_v(t) - \frac{\beta_v}{\eta}\ln\left(\frac{\tilde{x}_v(t)+\epsilon/|V|}{\tilde{x}_v(t-1)+\epsilon/|V|}\right)\tilde{x}_v(t)\right]}{\sum_{v,t}\left[\omega^r_{uv}(t)_{min}\tilde{x}_v(t) - \frac{\beta_v}{\eta}\ln\left(\frac{\tilde{x}_v(t)+\epsilon/|V|}{\tilde{x}_v(t-1)+\epsilon/|V|}\right)\tilde{x}_v(t)\right]}$$

$$\geq 1 - \frac{\beta - C}{\omega_{min} - C}$$

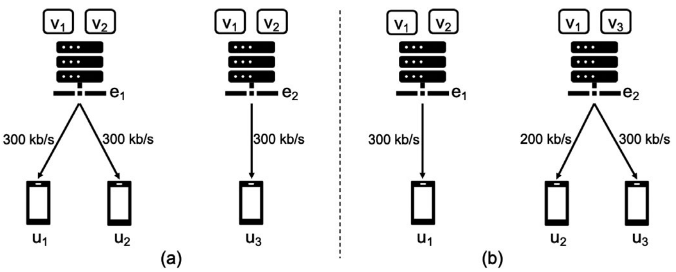
where

$$C = \frac{\sum_{v,t}(\tilde{x}_v(t) + \epsilon/|V|)\frac{\beta_v \tilde{x}_v(t)}{\eta}\ln\left(\frac{\tilde{x}_v(t)+\epsilon/|V|}{\tilde{x}_v(t-1)+\epsilon/|V|}\right)}{|V||T|}. \quad (8)$$

Note that $\omega_{min} = min_{u,v,r,t}\omega^r_{uv}(t)$, which indicates the lowest preference of fetched videos across all time slots, and $\beta = \max_v \beta_v$, which denotes biggest cost of fetched videos. Finally, we conclude that our VF algorithm is $\left(1 - \frac{\beta-C}{\omega_{min}-C}\right)$-competitive. □

5 Joint Optimization of Video Fetching and Edge-User Association

In practice, each mobile user can connect to multiple edge servers with different network bandwidths. In an intuitive design, each mobile user always chooses the edge server with the fastest network connection, which unfortunately is not the optimal solution considering the storage limitation and switching cost of edge servers. In this section, we study the joint optimization of video fetching and user-edge association to further increase the performance of short video sharing. In order to have a better understanding of the motivation, we use an example shown in Fig. 2 to explain the benefits of such a kind of joint optimization. In this example, we consider three users $\{u_1, u_2, u_3\}$, three types of videos $\{v_1, v_2, v_3\}$ and two edge servers $\{e_1, e_2\}$. each type of videos is encoded into a single version by edge servers. The users' watching preferences to all types of videos are shown in the table. User u_2 can connect to both edge servers, but its connection with server e_1 is faster than that with e_2. For simplicity, we consider only one time slot and study to maximize the total preferences of watched videos. By default, user u_2 always connects to the server e_1 due to its faster connection. In the optimal fetching solution, both edge servers fetch video types $\{v_1, v_2\}$ as shown in Fig. 2a and the total preference is 3.2. We find that user u_2 has higher preference on v_3 cached on e_2. If we let u_2 connect to e_2, the optimal caching solution for e_2 will be changed to $\{v_1, v_3\}$, as shown in Fig. 2b, which leads to an increased total preference of 3.4.



Preference	v1	v2	v3
u1	0.9	0,9	0,1
u2	0,1	0,1	0,5
u3	0,9	0.3	0.1

Fig. 2. An example showing the benefits of the joint optimization of video fetching and user-edge association. We assume the size of each type of videos as 150KB and each edge server has 400KB storage limitation.

To formulate the joint optimization problem, we define a new variable $\phi^e_u(t)$ for user-edge association as follows:

$$\phi^e_u(t) = \begin{cases} 1, & \text{if user } u \in U \text{ is associated with edge server} \\ & e \in E \text{ at time slot } t \in T; \\ 0, & \text{otherwise.} \end{cases}$$

In addition, we define a variable $y^{re}_{uv}(t)$ to denote the number of type-v videos pushed to user $u \in U$ with $r \in R$ bitrate in time slot $t \in T$. Here we consider a specific benefit of total preferences of cached videos and the problem of joint optimizing video fetching and user-edge association can be formulated as follows.

$$\max \sum_{u,v,r,e,t} \omega^r_{uv}(t) y^{re}_{uv}(t) - \sum_{v,e,t} \beta_v [x^e_v(t) - x^e_v(t-1)]^+ \quad (9)$$

$$y^{re}_{uv}(t) \leq x^e_v(t), \forall u \in U, v \in V, r \in R, e \in E, t \in T;$$

$$\sum_{v,r} s^r_v y^{re}_{uv}(t) \phi^e_u(t) \leq B^e_u(t)\tau, \forall u \in U, e \in E, t \in T; \quad (10)$$

$$y^{re}_{uv}(t) \geq 0, \forall u \in U, v \in V, r \in R, e \in E, t \in T; \quad (11)$$

$$\sum_{v,r} s^r_v x^e_v(t) \leq S_e, \forall e \in E, t \in T; \quad (12)$$

$$x^e_v(t) \geq 0, \forall v \in V, e \in E, t \in T; \quad (13)$$

$$\sum_{e} \phi^e_u(t) = 1, \forall u \in U, t \in T; \quad (14)$$

$$\sum_{u} \phi^e_u(t) \leq D_e, \forall e \in E, t \in T. \quad (15)$$

Here, $x^e_v(t)$ is similar to $x_v(t)$ to indicate the video caching associated with the server e. Therefore, the constraints (12) - (13) are the extensions of (1) - (2), respectively. In addition, we define a new binary variable $\phi^e_u(t)$ to denote the whether user u is associated with the edge server e in time slot t. We have constraint (14) because each user must be associated with only one edge server in each time slot. A user can watch videos cached at its associated edge server, which is represented by (10). Finally, we use (15) to constrain the number of users that can be connected to each edge server, which cannot exceed a predefined limit D_e.

Algorithm 2. Joint Optimization of Video Fetching and User-Edge Association (VF-UEA)

```
1:  for each time slot $t$ do
2:      obtain $B_u^e(t), \omega_{uv}^r(t), S_e, \forall u \in U, v \in V, r \in R, e \in E$;
3:      $\hat{x}_v^e(t) = \tilde{x}_v^e(t-1), \forall v \in V, e \in E$;
4:      while $\delta \leq \Delta$ do
5:          Solve the following optimization problem:
                $\max \sum_{u,v,r,e} \omega_{uv}^r(t) y_{uv}^{re}(t)$
                subject to: constraints $(9)-(11), (14)-(15)$;
6:          Extract the values of $\phi_u^e(t)$, which is denoted by $\tilde{\phi}_u^e(t)$ ;
7:          Then solve the following optimization problem:
                $\max \sum_{u,v,r,e} \omega_{uv}^r(t) y_{uv}^{re}(t) - \frac{1}{\eta} \sum_{v,e} \beta_v \left[ \left( x_v^e(t) + \frac{\epsilon}{|V|} \right) \cdot \ln \left( \frac{x_v^e(t) + \epsilon/|V|}{\tilde{x}_v^e(t-1) + \epsilon/|V|} \right) - x_v^e(t) \right]$    (16)
            subject to: $(9), (11)-(13)$, and
            $\sum_{v,r} s_v^r y_{uv}^{re}(t) \tilde{\phi}_u^e(t) \leq B_u^e(t)\tau, \forall u \in U, e \in E, t \in T$;    (17)
8:          Extract the value of $x_v^e(t)$, which is denoted by $\tilde{x}_v^e(t)$;
9:          $\hat{x}_v^e(t) = \tilde{x}_v^e(t)$ and $\delta = \delta + 1$;
10:     end while
11: end for
```

The formulated joint optimization problem is harder than the one in the previous section because we need to handle not only the cross-time video fetching decisions but also the user-edge association. We propose an algorithm to approximate the optimal solution by iteratively solving the sub-problems of video fetching and user-edge association (VF-UEA). The presudo codes are shown in Algorithm 2. In each time slot t, we obtain the parameters, such as network bandwidth $B_u^e(t)$ between user u and edge server e, user preference $\omega_{uv}^r(t)$ for $v-$type videos with $r-$bit version, as shown in the line 2. Then we solve two sub-problems by δ iterations in the lines 4-10. Specifically, we first solve the edge-user association $\tilde{\phi}_u^e(t)$ in the line 5, and then feed $\tilde{\phi}_u^e(t)$ to the video caching problem to obtain the video fetching solution $x_v^e(t)$ by solving the convex optimization problem (16), as shown in the lines 6-7. Problem (16) attempts to achieve a trade-off between users' preferences and the regularized switching costs. By iteratively optimizing these two sub-problems, we can approximate the optimal solution.

6 Performance Evaluation

In this section, we evaluate the performance of our proposed algorithms via extensive simulations. The simulation settings are presented first, followed by results of comparison with existing work.

6.1 Simulation Settings

We consider a number of short videos whose sizes are randomly distributed within [10MB, 100MB]. Each raw video is encoded into 5 versions with different bitrates at edge servers, and these versions have different sizes. The watching preferences $\omega_{uv}^r(t)$ of users are randomly generated within (0,1). The value of parameter ϵ used in our algorithms is set to 5. The maximum number of accepted connections of each edge server is 50, i.e., $D_e = 50$. We compare our proposals with the Receding Horizon Control(RHC) algorithm that has been widely applied in solving online optimization problems [40], [41], [42].

- Receding Horizon Control (RHC): RHC, which is also called Model Predictive Control (MPC), algorithm has a long history in control theory research. Specifically, in each time slot t, RHC solves the target optimization problem over a prediction window $[t, t+w]$ based on an initial state $x_v^e(t-1)$ and functions $(g_{uvr}^{et}, g_{uvr}^{et+1} \dots g_{uvr}^{et+w})$, which can be formally described as

$$\max \sum_{v,e} \sum_{t}^{t+w} g(\mathbf{X}(t)) - \sum_{v,e} \sum_{t}^{t+w} \beta_v [x_v^e(t) - x_v^e(t-1)]^+$$
$$\text{subject to (12).}$$

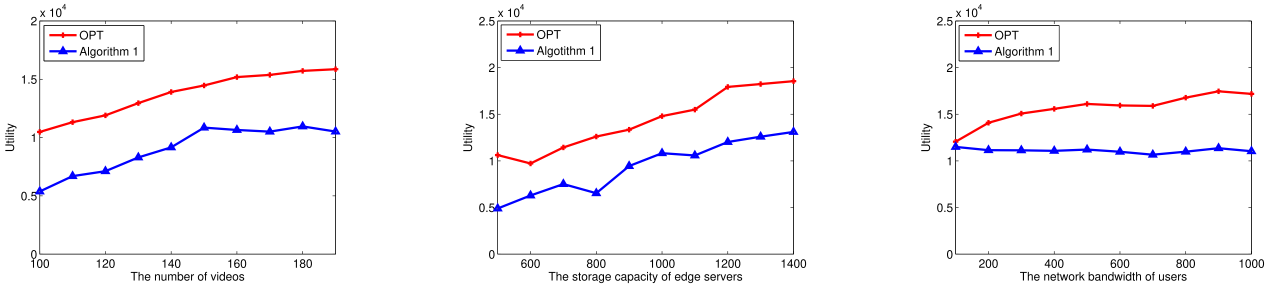
- ABR-Aware Proactive Cache Placement (APCP): In each time slot, we obtain the users' preferences information, which can be regarded as the popularity of videos. Then we follow the ABR-Aware Proactive Cache Placement (APCP) [7] to make fetching decision. Moreover, we also extend APCP by considering edge-user association, which is referred to as APCP-UEA.

6.2 Simulation Results

6.2.1 Performance Gap With the Optimal Solutions

We first evaluate the performance gap between our proposed VF with the optimal solutions of video fetching under various network settings. Due to the difficulty of obtaining the optimal solutions of large-scale problems, we consider relatively small-scale problems including 10 users, 150 video types and 20 time slots.

The performance gap under different number of videos types is shown in Fig. 3a. The storage capacity of the edge server is randomly generated with a mean of 1024MB. We can see that utility values increase as the growth of number of videos. That is because when the number of videos increases, we have more chances to choose the types that are preferred by more users. We then change the storage capacity of the edge server and study its effect on the performance gap. Specifically, we set the number of videos types as 150. We assume the storage capacity on edge servers close to the normal distribution, and we change the mean of the distribution to compare the utility of both solutions. As shown in Fig. 3b, the utility value increases as the growing of storage capacity. That is because we can download more videos in each time slot, and decrease the video replacement cost at the next time slot. Fig. 3c shows the performance gap

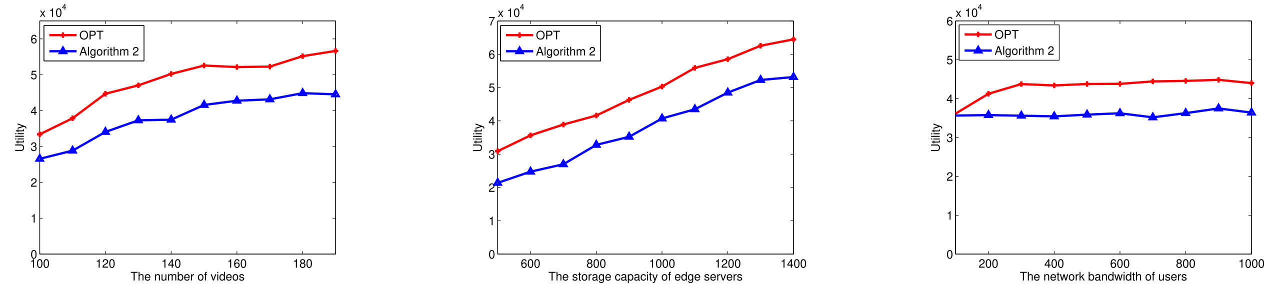


(a) The utility versus different number of videos.

(b) The utility versus different storage capacity of edge servers.

(c) The utility versus different network bandwidths of users.

Fig. 3. The preference gap between Algorithm 1 and the optimal solution of video fetching.



(a) The utility versus different number of videos.

(b) The utility versus different storage capacity of edge servers.

(c) The utility versus different network bandwidths of users.

Fig. 4. The preference gap between Algorithm 2 and the optimal solution of joint optimization of video fetching and user-edge association.

under different network bandwidths of users. We assume that network bandwidths of users belong to the normal distribution. We set the mean of storage capacity on edge server as 1024MB, and the number of videos types as 150. We can find the performance in our algorithm just has tiny fluctuations with the increasing of the bandwidth capacity on users. That is because if we increase the bandwidth capacity on users, our algorithm will choose to push more videos to users at the current time slot, therefore the edge server will cache videos the users both like. Hence the video replacement cost will increase with the changing of users.

The performance gaps between the VF-UEA and the corresponding optimal solution of joint optimization of video fetching and user-edge association are shown in Figs. 4a, 4b, and 4c. The results under different number of videos types are shown in Fig. 4a. We set the storage capacity on edge server as 1024MB. From the result, we can find that VF-UEA also performs well and can achieve about 80percent of optimal solutions. When the number of videos types increases, the server can have more choices to cache videos, leading to higher utility values. In Fig. 4b, we compare VF-UEA with optimal solutions under different storage capacity of the edge server. We set the number of videos types as 150. The reason for utility increasing with storage capacity on edge server is similar to that of Fig. 3b. In Fig. 4c, when the bandwidth increases, users can get more videos to get higher preferences, but it causes edge server to cache more videos with a higher videos replacement cost in online scenarios. So the growth trend of VF-UEA based on different bandwidth capacity is not obvious. In summary, VF-UEA can quickly converge to results that are close to the optimal solutions.

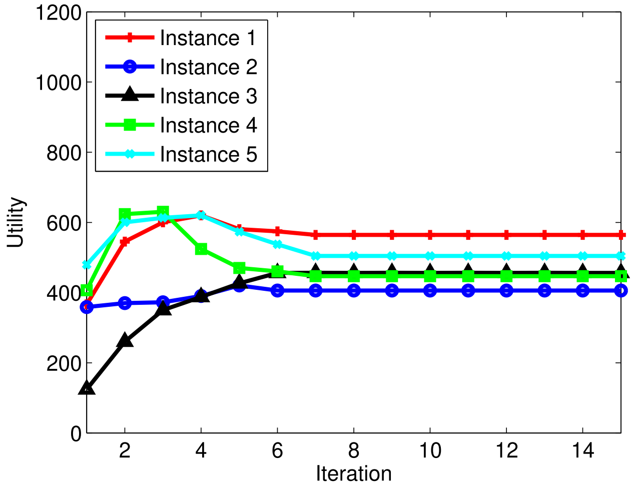


Fig. 5. The utility values in 15 iterations of VF-UEA.

6.2.2 Results of Large-Scale Simulations

We then consider large-scale scenarios consisting of 50 mobile devices, 3000 videos types, and 20 time slots. We first study the convergence of VF-UEA by showing its utility values in different iterations. We consider 5 random problem instances and show their convergence results in Fig. 5. The utility of all instances becomes stable after 6 iterations. We repeat the experiments on many other random problem instances and obtain similar results, which demonstrates that our proposed VF-UEA has small overhead in practice.

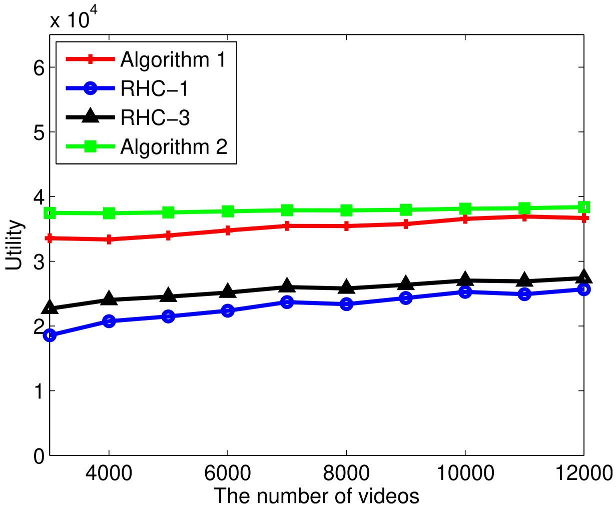


Fig. 6. The utility versus different number of videos.

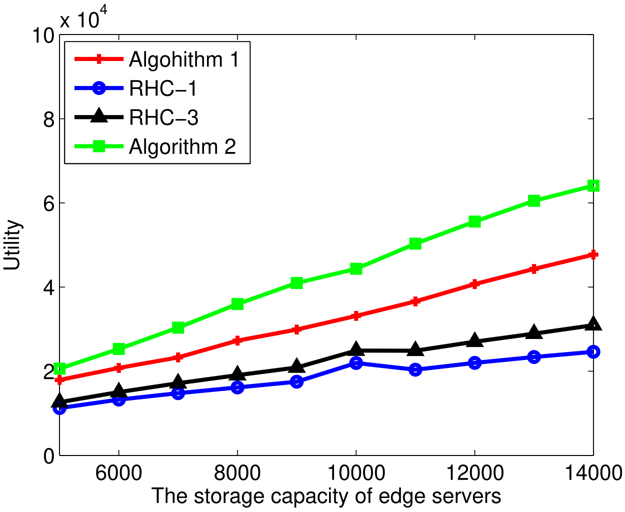


Fig. 8. The utility versus different storage capacity of edge servers.

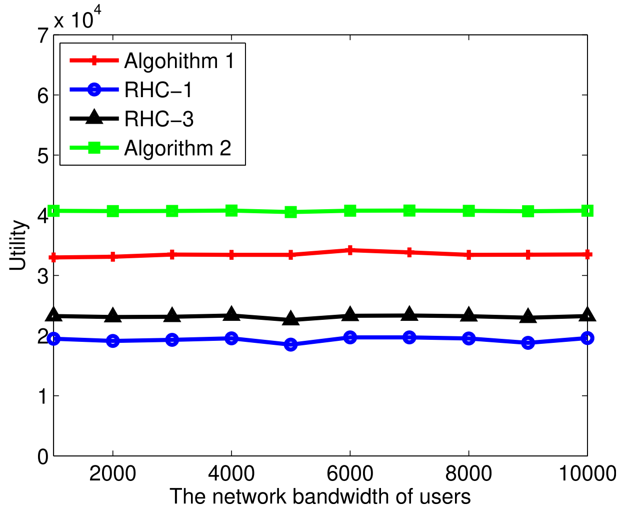


Fig. 9. The utility versus different network bandwidths of users.

We then study the performance of our proposed algorithms by comparing them with RHC. Specifically, we consider RHC under two settings with different window sizes, which are denoted by RHC-1 ($w = 1$) and RHC-3 ($w = 3$), respectively. The utility of these algorithms under different number of videos types is shown in Fig. 6. The number of users is set to 30. We can see that VF and VF-UEA always outperform other two algorithms in all settings. But the utility of these algorithms does not increase significantly as the growth of number of users. Furthermore, the performance gap between VF and VF-UEA becomes smaller when more videos are available.

Next, we set the number of videos types as 300 and change the number of users from 50 to 300. The results are shown in Fig. 7. The utility of all algorithms increases as more users joining the system. Moreover, VF-UEA achieves the highest utility compared with others.

The utility under different storage capacity of edge servers is shown in Fig. 8. We set the number of videos types and users as 300 and 50. The storage capacity of edge servers is randomly generated and we change the mean value from 5000MB to 15000MB. Our proposed algorithms outperform RHC-1 and RHC-3 and their utility increases as the growing of edge storage capacity. That is because we can cache more videos at the edge servers when their storage becomes larger. Even though users cannot watch all videos cached at edge servers in each time slot due to network

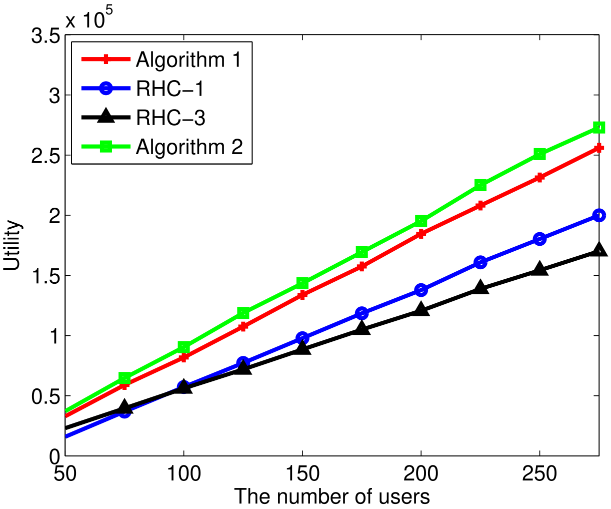


Fig. 7. The utility versus different number of users.

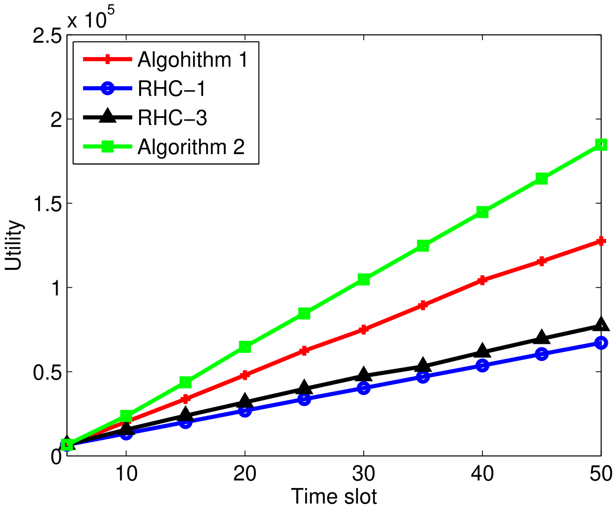
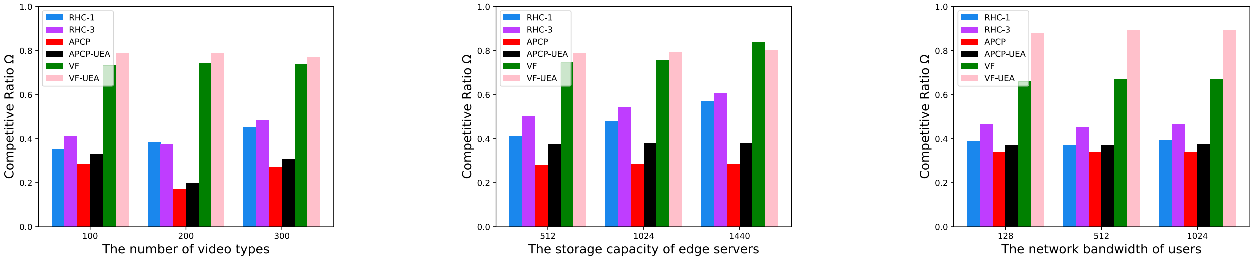


Fig. 10. The utility in different time slots.



(a) The ratio versus different number of video types.

(b) The ratio versus different storage capacity of edge servers.

(c) The ratio versus different network bandwidths of users.

Fig. 11. The competitive ratio of different caching algorithms.

bandwidth constraints, the video replacement cost can be reduced, leading to higher utility. Our proposed VF-UEA can outperform RHC-3 by 2.5 times when the mean of storage capacity is 14000MB.

The effect of network bandwidth of mobile users is shown in Fig. 9, where we set the number of users to 50, and the number of videos types is set to 300. Similar to the previous simulations, our proposed algorithms show higher utility than RHC-1 and RHC-3. We can also observe that the utility values do not have big changes as the increasing of network bandwidth. That is because with the bandwidth increasing, the users can watch more videos, hence edge servers need to cache more videos at each time slot, leading to higher video replacement cost.

We finally show the utility changes of all algorithms over time slots in Fig. 10. The number of users is set to 50, and the number of videos types is 300. We can see that all algorithms show similar performance in the beginning, but their performance gaps become bigger as the system evolves as time. Our proposed algorithms show great improvement of at most 3 times on RHC-1 after 50 time slots.

6.2.3 Comparison With APCP

In addition, we conduct experiments to compare our VF and VF-UEA algorithms with APCP [7] as well as its extension APCP-UEA. We first change the number of videos from 100 to 300 and show the comparison results in Fig. 11a. The storage capacity of MEC server is set to 1024 MB. We can find that our online algorithms outperform APCP by more than 2x. It is because that APCP cannot address the heavy switching cost as the increasing of video number.

Then we study their performance gap under different storage capacities. We set the number of video types to 150 and change the storage capacity from 512MB to 1440MB. As shown in Fig. 11b, our online algorithms outperform APCP by more than 1.5x. Since the edge server can cache more videos, APCP can maintain a stable ratio with decreased switching cost.

Finally, We study the influence of network bandwidth. As shown in Fig. 11c, we can find that APCP can achieve a stable 40percent of performance of the optimal solution when the users' network bandwidth changes from 128MB to 1024MB. It is because that APCP can improve the total preferences for performance enhancement. However, it brings higher switching cost. Our proposed algorithms outperform APCP and APCP-UEA thanks to the joint consideration of both preferences and switching costs.

7 Conclusion

In this paper, we propose an edge-assisted short video sharing framework to guarantee users' quality-of-experience (QoE) by exploiting the features of short videos and the benefits of powerful edge computing. A critical research challenge of this framework is to decide which videos should be cached at edge servers with limited storage capacity. Our proposed framework considers not only users' watching preferences to videos but also users' network bandwidth to make video fetching decisions. To address the challenge of dynamic watching preferences and user mobility, we design an online algorithm to optimize video fetching strategy without future information. Furthermore, we improve the performance by jointly considering video fetching and user-edge association and design an online algorithm that can quickly converge to the results close to the optimal solutions. Extensive simulation results demonstrate that our proposed algorithms significantly outperform existing work.

Acknowledgments

This work was supported by the funding from Hong Kong RGC Research Impact Fund (RIF) with the Project No. R5060-19 and R5034-18, General Research Fund (GRF) with the Project No. 152221/19E and 15220320/20E, Collaborative Research Fund (CRF) with the Project No. C5026-18G, the National Natural Science Foundation of China under Grants 61872310, No.61772480, No. 62073300, and Open Research Projects of Zhejiang Lab under Grant 2021KE0AB02.

References

[1] TikTok, 2016. [Online]. Available: https://www.tiktok.com/en
[2] Bermi, 2018. [Online]. Available: https://bermi.tv/en/
[3] Kwai, 2012. [Online]. Available: https://www.kwai.com/
[4] M. Mohsin, "10 tiktok statistics that you need to know in 2020," 2019. [Online]. Available: https://www.oberlo.com/blog/tiktok-statistics
[5] Y. Zhang *et al.*, "Challenges and chances for the emerging short video network," in *Proc. IEEE Conf. Comput. Commun. Workshops*, 2019, pp. 1025–1026.
[6] T. X. Tran, P. Pandey, A. Hajisami, and D. Pompili, "Collaborative multi-bitrate video caching and processing in mobile-edge computing networks," in *Proc. 13th Annu. Conf. Wireless On-Demand Netw. Syst. Service*, 2017, pp. 165–172.

[7] T. X. Tran and D. Pompili, "Adaptive bitrate video caching and processing in mobile-edge computing networks," *IEEE Trans. Mobile Comput.*, vol. 18, no. 9, pp. 1965–1978, Sep. 2018.

[8] E. Baccour, A. Erbad, K. Bilal, A. Mohamed, and M. Guizani, "PCCP: Proactive video chunks caching and processing in edge networks," *Future Gener. Comput. Syst.*, vol. 105, pp. 44–60, 2020.

[9] E. Baccour, A. Erbad, A. Mohamed, M. Guizani, and M. Hamdi, "CE-D2D: Collaborative and popularity-aware proactive chunks caching in edge networks," in *Proc. Int. Wireless Commun. Mobile Comput.*, 2020, pp. 1770–1776.

[10] X. Xu, J. Liu, and X. Tao, "Mobile edge computing enhanced adaptive bitrate video delivery with joint cache and radio resource allocation," *IEEE Access*, vol. 5, pp. 16 406–16 415, 2017.

[11] T. Zhang and S. Mao, "Joint video caching and processing for multi-bitrate videos in ultra-dense hetnets," *IEEE Open J. Commun. Soc.*, vol. 1, pp. 1230–1243, Aug. 2020.

[12] Y. Huang, Z. Li, G. Liu, and Y. Dai, "Cloud download: Using cloud utilities to achieve high-quality content distribution for unpopular videos," in *Pro. 19th ACM Int. Conf. Multimedia*, 2011, pp. 213–222.

[13] Y. Wang, W.-T. Chen, H. Wu, A. Kokaram, and J. Schaeffer, "A cloud-based large-scale distributed video analysis system," in *Proc. IEEE Int. Conf. Image Process.*, 2016, pp. 1499–1503.

[14] H. Yin *et al.*, "Design and deployment of a hybrid CDN-P2P system for live video streaming: experiences with livesky," in *Proc. 17th ACM Int. Conf. Multimedia*, 2009, pp. 25–34.

[15] R. Torres, A. Finamore, J. R. Kim, M. Mellia, M. M. Munafo, and S. Rao, "Dissecting video server selection strategies in the youtube CDN," in *Proc. 31st Int. Conf. Distrib. Comput. Syst.*, 2011, pp. 248–257.

[16] B. Li and H. Yin, "Peer-to-peer live video streaming on the internet: Issues, existing approaches, and challenges [peer-to-peer multimedia streaming]," *IEEE Commun. Mag.*, vol. 45, no. 6, pp. 94–99, Jun. 2007.

[17] Z. Chen, Q. He, Z. Mao, H.-M. Chung, and S. Maharjan, "A study on the characteristics of douyin short videos and implications for edge caching," in *Proc. ACM Turing Celebration Conf.-China*, 2019, pp. 1–6.

[18] X. Cheng, J. Liu, and C. Dale, "Understanding the characteristics of internet short video sharing: A youtube-based measurement study," *IEEE Trans. Multimedia*, vol. 15, no. 5, pp. 1184–1194, Aug. 2013.

[19] X. Cheng and J. Liu, "Nettube: Exploring social networks for peer-to-peer short video sharing," in *Proc. IEEE INFOCOM*, 2009, pp. 1152–1160.

[20] X. Cheng, C. Dale, and J. Liu, "Statistics and social network of youtube videos," in *Proc. 16th Int. Workshop Qual. Service*, 2008, pp. 229–238.

[21] 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. 2017.

[22] A. M. Ferman and A. M. Tekalp, "Multiscale content extraction and representation for video indexing," in *Multimedia Storage and Archiving Systems II*, vol. 3229. Bellingham, WA, USA: International Society for Optics and Photonics, 1997, pp. 23–31.

[23] Y. Li, S.-H. Lee, C.-H. Yeh, and C.-C. Kuo, "Techniques for movie content analysis and skimming: Tutorial and overview on video abstraction techniques," *IEEE Signal Process. Mag.*, vol. 23, no. 2, pp. 79–89, Mar. 2006.

[24] Z. Liu, C. Zhang, and Z. Zhang, "Learning-based perceptual image quality improvement for video conferencing," in *Proc. IEEE Int. Conf. Multimedia Expo*, 2007, pp. 1035–1038.

[25] M. Reisslein, F. Hartanto, and K. W. Ross, "Interactive video streaming with proxy servers," *Inf. Sci.*, vol. 140, no. 1–2, pp. 3–31, 2002.

[26] T. Shao, R. Wang, and J.-X. Hao, "Visual destination images in user-generated short videos: An exploratory study on douyin," in *Proc. 16th Int. Conf. Serv. Syst. Service Manage.*, 2019, pp. 1–5.

[27] I. Kahalo, H. Beshley, A. Masiuk, and V. Pashkevych, "The method of transmitting real-time video streams for Wi-Fi networks with short-term channel failures," in *Proc. 3rd Int. Conf. Inf. Commun. Technol.*, 2019, pp. 356–359.

[28] M. David, R. Silber, and S. Toba, "Automated platform for recording high quality short scientific videos," in *Proc. IEEE Int. Conf. Multimedia Expo Workshops*, 2016, pp. 1–2.

[29] P. Cika and D. Starha, "Video quality assessment on mobile devices," in *Proc. 25th Int. Conf. Syst. Signal Image Process.*, 2018, pp. 1–4.

[30] Y. Li, H. Jia, C. Zhu, M. Li, X. Xie, and W. Gao, "Low-delay window-based rate control scheme for video quality optimization in video encoder," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2014, pp. 7333–7337.

[31] D. Ghadiyaram, J. Pan, and A. C. Bovik, "A subjective and objective study of stalling events in mobile streaming videos," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 29, no. 1, pp. 183–197, Jan. 2017.

[32] F. Wang *et al.*, "Intelligent edge-assisted crowdcast with deep reinforcement learning for personalized QoE," in *Proc. IEEE Conf. Comput. Commun.*, 2019, pp. 910–918.

[33] C. Avasalcai, C. Tsigkanos, and S. Dustdar, "Decentralized resource auctioning for latency-sensitive edge computing," in *Proc. IEEE Int. Conf. Edge Comput.*, 2019, pp. 72–76.

[34] S. Y. Nikouei, Y. Chen, S. Song, R. Xu, B.-Y. Choi, and T. R. Faughnan, "Real-time human detection as an edge service enabled by a lightweight CNN," in *Proc. IEEE Int. Conf. Edge Comput.*, 2018, pp. 125–129.

[35] X. Li *et al.*, "COMEC: Computation offloading for video-based heart rate detection APP in mobile edge computing," in *Proc. IEEE Int. Conf. Parallel Distrib. Process. Appl., Ubiquitous Comput. Commun., Big Data Cloud Comput. Soc. Comput. Netw., Sustain. Comput. Commun.*, 2018, pp. 1038–1039.

[36] S. Sridhar and M. E. Tolentino, "Evaluating voice interaction pipelines at the edge," in *Proc. IEEE Int. Conf. Edge Comput.*, 2017, pp. 248–251.

[37] X. Huang, L. He, X. Chen, G. Liu, and F. Li, "A more refined mobile edge cache replacement scheme for adaptive video streaming with mutual cooperation in multi-mec servers," in *Proc. IEEE Int. Conf. Multimedia Expo*, 2020, pp. 1–6.

[38] C. Liang, Y. He, F. R. Yu, and N. Zhao, "Enhancing QoE-aware wireless edge caching with software-defined wireless networks," *IEEE Trans. Wireless Commun.*, vol. 16, no. 10, pp. 6912–6925, Oct. 2017.

[39] D. Huang, X. Tao, C. Jiang, S. Cui, and J. Lu, "Trace-driven QoE-aware proactive caching for mobile video streaming in metropolis," *IEEE Trans. Wireless Commun.*, vol. 19, no. 1, pp. 62–76, Jan. 2019.

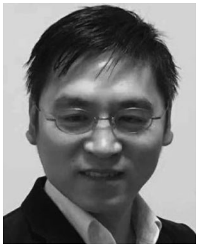
[40] D. Q. Mayne and H. Michalska, "Receding horizon control of nonlinear systems," *IEEE Trans. Autom. Control*, vol. 35, no. 7, pp. 814–824, Jul. 1990.

[41] D. Kusic, J. O. Kephart, J. E. Hanson, N. Kandasamy, and G. Jiang, "Power and performance management of virtualized computing environments via lookahead control," *Cluster Comput.*, vol. 12, no. 1, pp. 1–15, 2009.

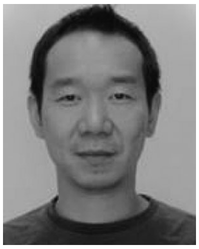
[42] X. Wang and M. Chen, "Cluster-level feedback power control for performance optimization," in *Proc. IEEE 14th Int. Symp. High Perform. Comput. Archit.*, 2008, pp. 101–110.

Fahao Chen is currently working toward the master's degree with the Graduate School of Computer Science and Engineering, The University of Aizu, Japan. His research interests include mainly focus on cloud/edge computing, and distributed systems for machine learning.

Peng Li (Senior Member, IEEE) received the BS degree from the Huazhong University of Science and Technology, China, in 2007, and the MS and PhD degrees from the University of Aizu, Japan, in 2009 and 2012, respectively. He is currently an associate professor with the University of Aizu, Japan. His research interests include mainly focus on cloud/edge computing, Internet-of-Things, machine learning systems, as well as related wired and wireless networking problems. He has published more than 100 technical articles on prestigious journals and conferences. He won the Young Author Award of IEEE Computer Society Japan Chapter in 2014. He won the Best Article Award of IEEE TrustCom 2016. He supervised students to win the First Prize of IEEE ComSoc Student Competition in 2016. He is the editor of *IEICE Transactions on Communications*, and IEEE Open Journal of the Computer Society.



Deze Zeng (Member, IEEE) received the BS degree from the School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan, China, in 2007, and the MS and PhD degrees in computer science from the University of Aizu, Aizuwakamatsu, Japan, in 2013 and 2009, respectively. He is currently a full professor with the School of Computer Science, China University of Geosciences, Wuhan. His current research interests include mainly focus on edge computing, and related technologies like network function virtualization, machine learning, and IoT. He has authored three books and more than 100 articles in refereed journals and conferences in the above areas. He received the three best article awards from IEEE/ACM conferences and the IEEE Systems Journal Annual Best Article Award of 2017. He serves on the editorial boards for the Journal of Network and Computer Applications and a guest editor for many prestigious journals. He has been in the organization or program committees of many international conferences, including ICPADS, ICA3PP, CollaberateCom, MobiQuitous, ICC, and Globecom. He is a senior member of CCF.



Song Guo (Fellow, IEEE) is currently a full professor with the Department of Computing, The Hong Kong Polytechnic University. He also holds a Changjiang chair professorship awarded by the Ministry of Education of China. His research interests include the areas of big data, edge AI, mobile computing, and distributed systems. He has published more than 500 articles in major journals and conferences and been recognized as a Highly Cited Researcher (Web of Science). He is the recipient of more than 12 best article awards from IEEE/ACM conferences, journals and technical committees. He is the editor-in-chief of the *IEEE Open Journal of the Computer Society* and the Chair of IEEE Communications Society (ComSoc) Space and Satellite Communications Technical Committee. He has served on IEEE ComSoc Board of Governors, IEEE Computer Society on Fellow Evaluation Committee, and editorial board of a number of prestigious international journals like the *IEEE Transactions on Parallel and Distributed Systems*, the *IEEE Transactions on Cloud Computing*, *IEEE Internet of Things Journal*, etc. He has also served as chair of organizing and technical committees of many international conferences. He is an ACM distinguished member.