

RESEARCH

Open Access



# Joint optimization strategy for QoE-aware encrypted video caching and content distributing in multi-edge collaborative computing environment

Zhi Liu<sup>1</sup>, Bo Qiao<sup>2</sup> and Kui Fang<sup>1\*</sup>

## Abstract

The video request service of users in 5G network will explode, and adaptive bit rate technology can provide users with reliable video response. Placing video resources on edge servers close to users can overcome the problem of excessive network load similar to traditional centralized cloud platform solutions. Moreover, multiple edge servers can provide caching and transcoding support by collaboration mechanisms, which further improves users' Quality of Experience (QoE). However, the design difficulty of video caching and content distribution strategies is increased due to the diversity of collaboration mechanisms and the competition between local and collaborative services of edge servers for computing and storage resources. In order to solve this problem, video cache and content distribution problem is modeled as random integer programming problem in the multi-edge server at most two-hop collaboration scenario. In order to improve the security of video data transmission, the video stream is encrypted using an encryption algorithm based on Logistic chaotic-Quantum-dot Cellular Automata (QCA). For improving the efficiency of solving integer programming problems, this paper uses a pyramid intelligent evolution algorithm based on optimal cooperation strategy to solve this problem. Simulation experiments show that our proposed method can obtain higher QoE value compared with several newer methods. In addition, the average access delay of proposed method is shortened by more than 27.98%, which verifies its reliability.

**Keywords:** Multi-edge collaborative computing, Video data security, Video content distribution, Pyramid structure, Quality of experience, Swarm intelligence evolution algorithm, Random integer programming

## Introduction

In 5G networks, video services will become the mainstream, and the contradiction between explosive growth of data volume and QoE is becoming increasingly prominent [1]. In other words, in a 5G network, when users initiate a request for a certain bit-rate video resource, remote storage devices must respond to users by codec operations within the shortest possible time [2]. Therefore, research on QoE-aware video caching and content

distribution technology has attracted extensive attentions from academia and industry.

Due to differences in user own hardware processing capabilities, network channel conditions, etc., different users usually request video files of different quality from remote video storage devices [3]. According to behavioral characteristics of users, Adaptive Bit Rate (ABR) technology is widely used in video services to improve QoE of users [4, 5].

There are obvious differences in user own hardware processing capabilities and network channel conditions, the energy carried by user terminals such as mobile phones and tablet computers is often limited. Therefore,

\* Correspondence: [fk@hunau.net](mailto:fk@hunau.net)

<sup>1</sup>Hunan Engineering Technology Research Center of Rural & Agricultural Informationization, Hunan Agricultural University, Changsha 410128, Hunan, China

Full list of author information is available at the end of the article

it is generally not used to superimpose and decode multiple coding layers on the user side to obtain the bit rate video required by users, such as Scalable Video Coding (SVC) technology [6]. Conversely, for same video files, the more common implementation of ABR is that the remote end is first encoded as a video version with different formats and resolutions. Then, according to users' request and network conditions, a variant file is selectively send to users [7, 8]. This method uses "storage/compute-transmit" mode, and its advantage is that the decoding overhead of mobile terminals is avoided, thereby saving corresponding energy consumption. However, the corresponding disadvantage is that when the number of users is very large and video request operations are frequent, the video traffic in network will show an explosive growth. This will cause the transformation time of video files to be too long and QoE to drop.

Further, the available technical routes can be divided into centralized cloud computing solutions and decentralized edge collaborative computing solutions in the ABR technology adopting "storage/computing-transmission" mode. In centralized cloud computing solutions, video storage and codec operations are all implemented on the remote cloud platform. Users need to obtain corresponding video resources from the far ends [9, 10]. In decentralized edge collaborative computing solutions, video files with different bit rates and formats are stored on edge computing devices closer to users. In addition, different edge computing devices can implement ABR functions by cooperative codec and resource interaction [11, 12]. Compared with centralized cloud computing solutions, decentralized edge computing solutions can effectively reduce video traffic overhead in the network. Since edge computing devices are closer to the user side, the QoE of users is correspondingly higher [13]. However, it should be pointed out that the computing resources and storage resources of edge computing devices are often limited. A single edge computing device cannot satisfy the explosive growth of video requests in 5G networks [14]. However, the optional response methods of edge computing devices are very diverse when faced with the same video request. For example, local nodes cache and directly transmit, neighbor nodes cache and transmit, neighbor nodes decode and then cache and transmit, and neighbor nodes cache and decode and transmits, which undoubtedly increases the difficulty of designing video caching and content distribution mechanisms. This paper mainly design a video cache and content distribution optimization strategy that is QoE-aware in a multi-edge collaborative computing environment.

### Related works

Considering the hardware resources and energy consumption requirements of user terminals, traditional SVC coding technology [6] is the representative, and

passive ABR technology that requires users to perform the decoding operation is no longer suitable for video services in 5G networks. On the contrary, another kind of resource caching and encoding/decoding operation performed by remote resource storage devices has been welcomed by academia and industry. The basic principle is that, using the "storage/computing-transmission" mode, remote resource storage devices adaptively provide video streams corresponding to the bit rate according to user requests and network conditions [15, 16]. There are two main technical solutions available, namely centralized cloud computing solutions and decentralized edge collaborative computing solutions in the "storage/computing-transmission" mode.

**Centralized cloud computing solution:** All video files are stored on a remote cloud platform. Users obtain video resources with a specific bit rate by wide area networks. Many references discuss the design of video caching and content distribution mechanisms under the cloud computing framework. For example, under the framework of centralized cloud computing, physical cache and virtual cache were designed to respond to video file storage and online transcoding computing respectively in reference [17]. Reference [18] proposed a rate adaptation algorithm that uses video characteristics to simultaneously change video encoding and transmission rate, which improves the amount of video resources that the network can tolerate. Reference [19] modeled the cache management of video stream files as a constrained optimization problem considering the server storage resource constraints. Reference [20] verified an online architecture capable of using Docker for real-time video transcoding in a cloud environment based on Kubernetes. The random forest regressor used in this framework provided the best overall performance in terms of transcoding speed, resource CPU consumption and accuracy of the number of transcoding tasks implemented, but the work efficiency of reinforcement learning is low. However, in the centralized cloud computing mode, user video requests are sent to users after caching and encoding/decoding operations on the cloud platform. Thus, in order to ensure users' QoE, the cloud computing model has high requirements on network bandwidth, hardware conditions of storage and computing equipment, and has the disadvantage of high construction and maintenance costs [21].

**Decentralized edge computing solution:** In this solution, video files with different bit rates and formats are first stored on multiple edge computing devices close to the user side. When a user requests a video service with a certain bit rate and format, the edge computing device implements video caching, codec, and transmission operations in a cooperative manner [22, 23]. Compared with the centralized cloud computing model, the network backhaul time between edge computing devices and users is shorter, and the hardware performance

requirements are also lower. But it should be pointed out that the implementation of ABR technology is diverse under edge computing schemes. For example, for the same video file request, edge computing devices can directly transmit by local caches, cache from the neighbor nodes and then transmit, decode from the neighbor nodes and then cache and then transmit, and cache from the neighbor nodes and then decode and then transmit. Similar work was shown in [24]. Summary, a video cache and content distribution mechanism with good performance must be flexibly adjusted according to network conditions, network topology, edge device working status, etc. to obtain satisfactory QoE in a multi-edge collaborative computing environment [25].

The existing work has carried out preliminary research on the video caching and content distribution mechanism in multi-edge collaborative computing environment, and has achieved some beneficial results. For example, reference [26] proposed an adaptive wireless video transcoding framework in the emerging edge computing mode to achieve more detailed video transcoding. However, this solution inevitably increased the further occupation of computing resources while tracking changes in traffic. Reference [27] considered the collaboration between multiple edge servers. However, there was no cooperative transcoding service between edge servers. In other words, all video transcoding operations were performed on local servers, which requires high server computing performance and storage capacity. Reference [28] considered the collaborative caching and transcoding between edge servers. However, this solution only considered single-hop mode in which edge servers perform video caching and transcoding between only two edge servers.

Based on the above analysis, this paper proposes a multi-edge collaborative video caching and content distribution mechanism based on random integer programming. The main innovations are as follows:

- 1) Considering that video caching and content distribution can complete caching and transcoding operations on different edge servers, a video caching and content distribution mechanism including two-hop cooperation of edge servers is proposed. Compared with the traditional single-hop cooperation mode, it only considers two edge servers and the considered caching and distribution mechanism is more general;
- 2) Based on the stochastic optimization method, with the user QoE as optimization goal, video cache and content distribution problems are modeled as random integer linear programming problems. Among them, the video cache in edge devices fully takes into account the popularity of videos, which further improves the hit rate of video storage and the corresponding QoE.

- 3) In order to improve the security of video data transmission, the video stream is encrypted using an encryption algorithm based on Logistic chaotic-Quantum-dot Cellular Automata (QCA).
- 4) In order to improve the efficiency of solving integer programming problems, a pyramid structure intelligent evolution algorithm based on optimal cooperation strategy is proposed to solve the problem.

### System model

As shown in Fig. 1, let the number of edge servers be  $N$ . It can be cached from the video resource library in remote servers, and can also perform codec operations. Two-way data exchange between high-speed links between edge servers. At the same time, it can also be directly linked to remote servers by the backhaul link. The cache and codec operations of edge servers are subject to scheduling and control of the control center.

### Video coding encryption technology

The rapid development of communication technology provides users with diversified and differentiated video requirements. At the same time, the importance of video information transmission security to both video providers and users cannot be ignored. At present, the industry generally uses H.264/AVC encoding standard to compress and transmit videos with low distortion. This paper uses an encryption algorithm based on Logistic chaotic-Quantum-dot Cellular Automata (QCA) [29] to encrypt video coding.

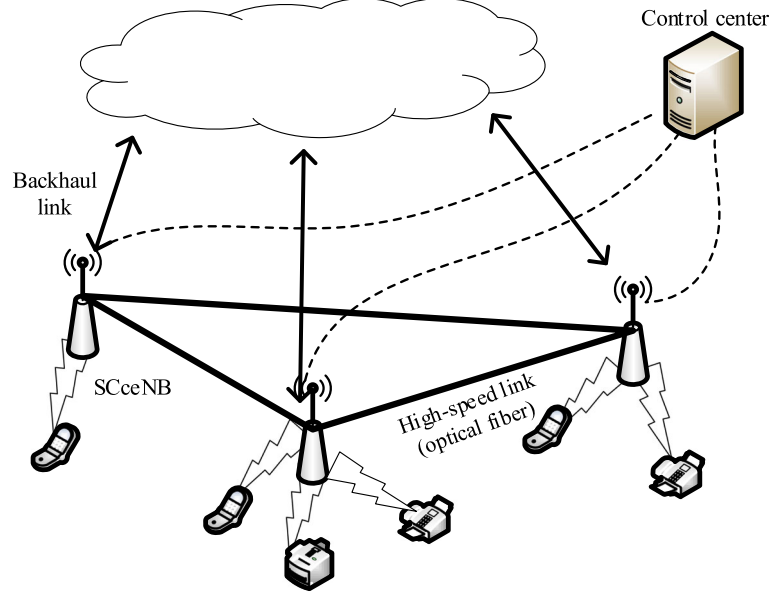
The flow of Logistic chaotic-QCA key generation algorithm is shown in Fig. 2. Among them, Quantum Cellular Neural Network (QCNN) matrix  $A$  is obtained by QCA using Logistic chaotic system for  $h$  consecutive iterations.  $h$  is a multiple of 512, and Logistic chaotic system is shown in formula (1).

$$X_{h+1} = f(X_n) = \Psi X_n(1 - X_n) \quad (1)$$

where  $\Psi \in (0, 4)$ ,  $X_{h+1} \in (0, 1)$ ,  $h = 1, 2, 3, \dots$ ;  $X_n$  ( $n = 1, 2, 3, 4$ ) is the QCA state vector, which satisfies

$$\begin{cases} X_1 = -2\omega_{01}\sqrt{1-X_1^2}\sin x_2 \\ X_2 = -\omega_{02}(X_1 - X_3) + 2\omega_{01}\frac{X_1}{\sqrt{1-X_1^2}}\cos x_2 \\ X_3 = -2\omega_{03}\sqrt{1-X_3^2}\sin x_4 \\ X_4 = -\omega_{04}(X_3 - X_4) + 2\omega_{03}\frac{X_3}{\sqrt{1-X_3^2}}\cos x_4 \end{cases} \quad (2)$$

where  $x_1, x_3$  are the polarizability;  $x_2, x_4$  are the quantum phases;  $\omega_{01}, \omega_{03}$  are the coefficients proportional to the energy between points in each cell;  $\omega_{02}, \omega_{04}$  are the weighted effects of difference in the polarizability of adjacent cells coefficient.



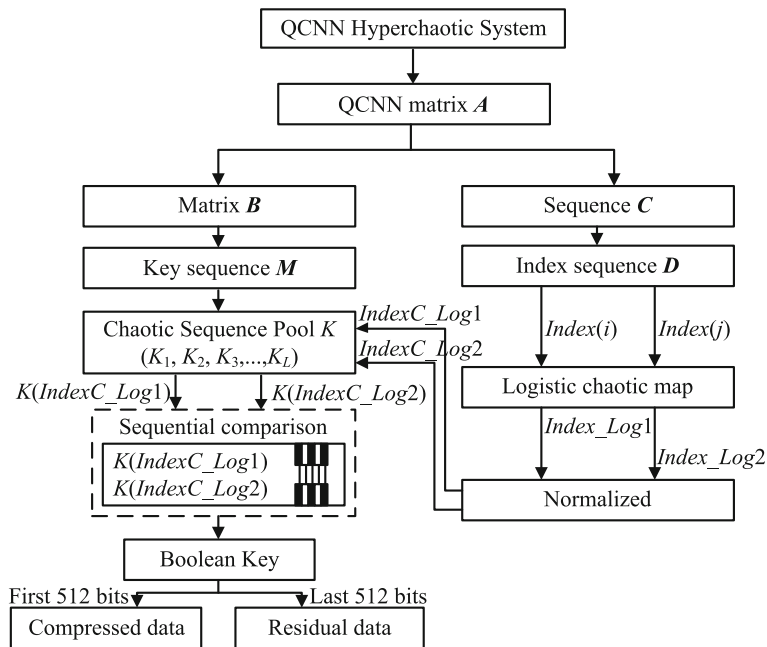
**Fig. 1** Video caching and content distribution model in a multi-edge collaborative computing environment

Suppose the generated matrix  $A$  satisfies

$$A = \begin{bmatrix} X_{11} & X_{12} & \cdots & X_{1(n-1)} & X_{1n} \\ X_{21} & X_{22} & \cdots & X_{2(n-1)} & X_{2n} \\ X_{31} & X_{32} & \cdots & X_{3(n-1)} & X_{3n} \\ X_{41} & X_{42} & \cdots & X_{4(n-1)} & X_{4n} \end{bmatrix} \quad (3)$$

Split matrix  $A$  into matrix  $B$  composed of the first 3 rows of elements and matrix  $C$  composed of the last row of elements, that is

$$B = \begin{bmatrix} X_{11} & X_{12} & \cdots & X_{1(n-1)} & X_{1n} \\ X_{21} & X_{22} & \cdots & X_{2(n-1)} & X_{2n} \\ X_{31} & X_{32} & \cdots & X_{3(n-1)} & X_{3n} \end{bmatrix} \quad (4)$$



**Fig. 2** Logistic-QCA encryption process

$$C = [X_{41} \ X_{42} \ \cdots \ X_{4(n-1)} \ X_{4n}] \quad (5)$$

The matrix  $B$  is further processed and converted into a row vector to form a key sequence, denoted as  $S$ , as shown in the following formula

$$S = \{S_1, S_2, \dots, S_{3 \times n}\} \\ = \{X_{11}, X_{12}, \dots, X_{21}, X_{22}, \dots, X_{3(n-1)}, X_{3n}\} \quad (6)$$

The elements in  $S$  are in order, and each 512 forms a group to form a chaotic sequence pool  $H$ , as shown in Eq. (7).

$$H = \{h_1, h_2, \dots, h_L\} \quad (7)$$

$$\begin{cases} h_1 = \{S_1, S_2, \dots, S_{512}\} \\ h_2 = \{S_{513}, S_{514}, \dots, S_{1024}\} \\ \vdots \\ h_L = \{S_{(L-1) \times 512 + 1}, S_{(L-1) \times 512 + 2}, \dots, S_{L \times 512}\} \end{cases} \quad (8)$$

where  $L$  is an integer between  $(0, 3n/512]$ .

For the elements in sequence  $C$ , the index sequence *Index* is generated according to formula (9):

$$Index = \text{map min max}(C, 0, 1) \quad (9)$$

where  $\text{map min max}(C, 0, 1)$  means to map the value in sequence  $C$  to interval  $[0, 1]$ .

$Index(i)$  and  $Index(j)$  are selected from matrix *Index* as the initial values and  $Index\_Log1$  and  $Index\_Log2$  are generated by perform Logistic transformation. And further we round up according to formula (10) to map to the integers  $IndexC\_Log1$  and  $IndexC\_Log2$  in interval  $[1, L]$ .

$$\begin{cases} IndexC\_Log1 = \lceil Index\_Log1 \times L \rceil \\ IndexC\_Log2 = \lceil Index\_Log2 \times L \rceil \end{cases} \quad (10)$$

Finally, the key generation is selected from key sequence  $S$  according to  $IndexC\_Log1$  and  $IndexC\_Log2$ , and the key is obtained by comparing bit by bit according to formula (11), until a 512-bit complete key is obtained.

$$Key = S_{Index\_Log1 \geq (S_{Index\_Log2} ? 1 : 0)} \quad (11)$$

The original video is encoded with H.264/AVC and contains two types of compressed video data and residual data [30]. In order to improve the reliability of video transmission, this paper uses different keys to encrypt these two types of data. That is, for the compressed video data, the first 256 bits of the *Key* are used for encryption. For residual data, the last 256 bits of the *Key* are used for encryption.

#### Multi-rate video cache model

Let collection  $V = \{1, 2, \dots, s, \dots, S\}$  be the video collection. Each video can be encoded into  $M$  different versions of

files. The definition set  $v_s = \{v_{sm} | m = 1, 2, \dots, M\}$  is represented as a variant set of the  $s$  video file. For video files, we can use bit rate and playback duration to characterize. In addition, note that for the same video, the playback duration of different variant files is the same. Therefore, video file  $v_{sm}$  can be described by the following binary vector, namely

$$v_{sm} : (r_{sm}, l_s) \quad (12)$$

where  $r_{sm}$  and  $l_s$  are the bit rate and playing time of  $v_{sm}$  respectively.

Without loss of generality, let each variant file in  $v_s$  be stored in ascending order of bit rate, that is,  $r_{s1} < r_{s2} < \dots < r_{sM}$ . Besides, low bit rate files can be transcoded from high bit rate files. Let each edge server have a cache user of size  $C$  to store a copy of the video, and  $C$  is greater than the size of the video corresponding to maximum bit rate, i.e.

$$C = \alpha r_{sM} \quad (13)$$

where  $\alpha$  is a coefficient greater than 1.

Based on the above analysis, first it assume that the cache capacity of each edge server is limited. However, video files of any bit rate can be cached to meet the video requests of different users. However, it should be noted that although videos generally have multiple versions with different bit rates. However, considering user QoE and its own network conditions, the current mainstream commercial streaming media system usually adaptively adjusts the transmitted video files to a level that matches the current network conditions. Thus, as shown in Fig. 3, the video caching strategy in this paper is:

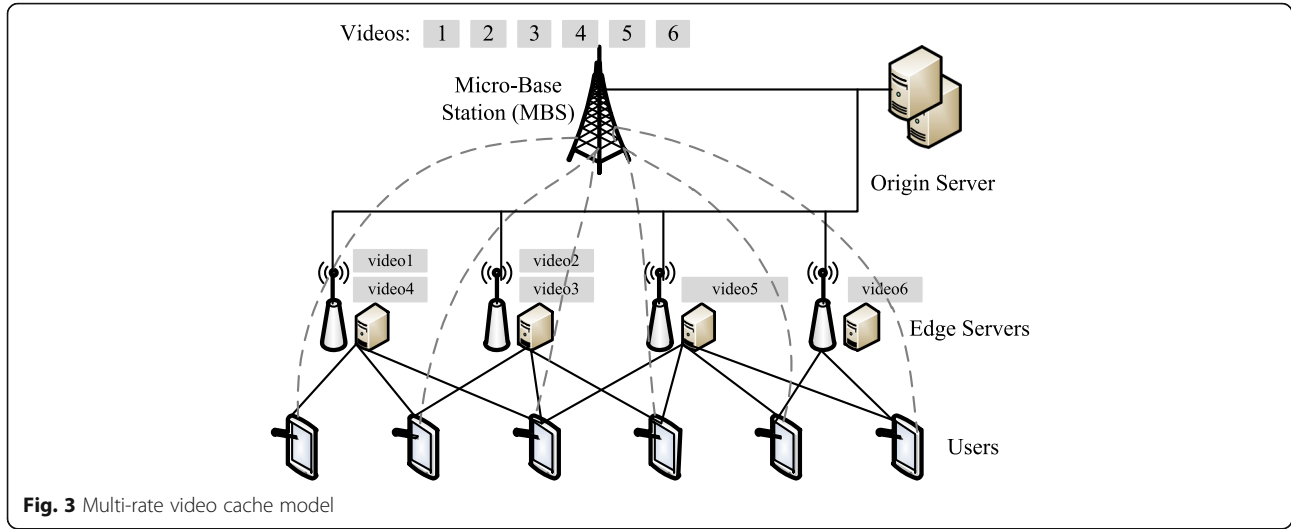
When a user  $k(=1, 2, \dots, K)$  requests video  $v_s$ , if no edge server within  $k$  single-hop communication range caches a video copy of any bit rate of video  $v_s$ , then  $k$  will directly use the lowest bit rate that the current network can afford. The remote server directly caches video  $v_s$ . Conversely, user  $k$  will cache the highest bit rate version cached by edge servers within the single-hop communication range.

#### Video distribution strategy design

Under the edge server collaborative computing framework, user video requests can be obtained from a remote server or an edge server directly connected to users by a single-hop connection, and can also be obtained by other edge server transcoding operations. Figure 4 shows all possible distribution methods when a user requests a 360p version of a video file.

Combining with Fig. 4, the following 8 binary variables are introduced to characterize the feasible video distribution scheme in multi-edge collaborative computing environment:





**Fig. 3** Multi-rate video cache model

- 1)  $a_n^{sm}(t) = 1$  means that users directly obtain video  $v_{sm}$  from the cache area of edge servers connected to it with a single hop (may be set to  $n$ , the same below); if not,  $a_n^{sm}(t) = 0$ , as shown in Fig. 4(a);
- 2)  $b_n^{sm}(t) = 1$  means that video  $v_{sm}$  requested by users is obtained by transcoding operation of the high-bit rate version by edge server  $n$  connected to it with a single hop; if not, then  $b_n^{sm}(t) = 0$ , as shown in Fig. 4(b);
- 3)  $c_{nn'}^{sm}(t) = 1$  means that video  $v_{sm}$  requested by users is directly obtained from edge server  $n'$  ( $n \neq n'$ ); if not, then  $c_{nn'}^{sm}(t) = 0$ , as shown in Fig. 4(c);
- 4)  $d_{nn'}^{sm}(t) = 1$  means that video  $v_{sm}$  requested by users is obtained by transcoding from the higher version in edge server  $n'$  ( $n \neq n'$ ); if not, then  $d_{nn'}^{sm}(t) = 0$ , as shown in Fig. 4(d);
- 5)  $e_{nn'}^{smm'}(t) = 1$  indicates that video  $v_{sm}$  requested by users is first obtained by edge server  $n$  from edge server  $n'$  ( $n \neq n'$ ), and then obtained after performing a forwarding operation locally; if not,  $e_{nn'}^{smm'}(t) = 0$ , as shown in Fig. 4(e);
- 6)  $f_{nn'}^{sm}(t) = 1$  indicates that video  $v_{sm}$  requested by users is first obtained by edge server  $n'$  ( $n \neq n'$ ) from edge server  $n$  with a high bit rate version. Then edge server  $n'$  ( $n \neq n'$ ) performs the forwarding operation and returns it to edge server  $n$ ; if not,  $f_{nn'}^{sm}(t) = 0$ , as shown in Fig. 4(f). It should be noted that there is a working mechanism of the two-hop cooperation of edge servers in this distribution mode;
- 7)  $g_{nn'n''}^{smm}(t) = 1$  indicates that video  $v_{sm}$  requested by users is first obtained by edge server  $n''$  from edge server  $n'$  and then transcoded to edge server  $n$ ; if not, then  $g_{nn'n''}^{smm}(t) = 0$ , as shown in Fig. 4(g). It should be noted that in this distribution mode,

there is also a working mechanism of the two-hop cooperation of edge servers;

- 8)  $h^{sm}(t) = 1$  means that video  $v_{sm}$  requested by users is obtained from a remote server; if not,  $h^{sm}(t) = 0$  is shown in Fig. 4(h).

## Problem modeling

### Problem objective and QoE function

The problem of video caching and content distribution is modeled as a constrained optimization problem.

First, define the following cache strategy set:

$$X = V \times R \times N = \{(v, r, n) : v \in V, r \in Q, n \in N\} \quad (14)$$

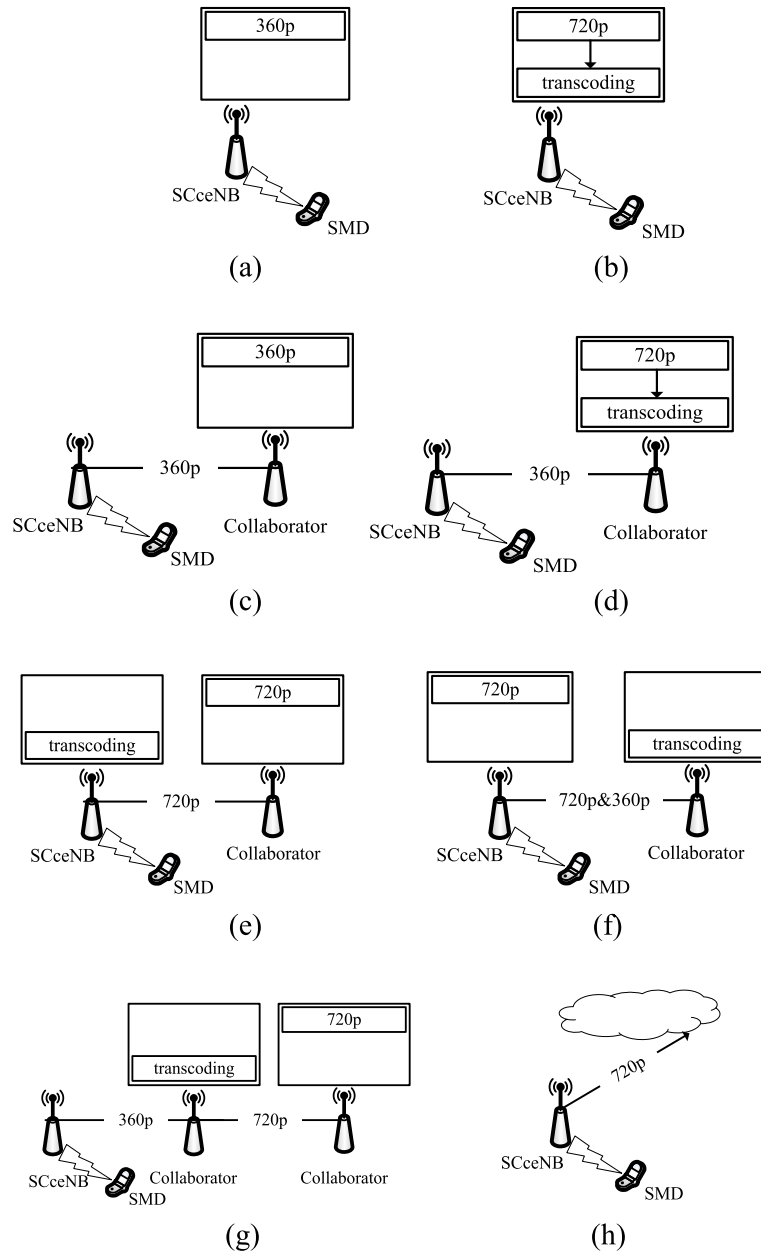
The above set indicates that  $r$  bit rate version of video  $v$  is cached in edge server  $n$ . According to the adaptive bit rate caching strategy designed in section 2.1, the bit rate selected by user  $k$  can be expressed as

$$r_{k,v} = r_{[\arg \max_{r \in R} \{\exists n \in N, (v, r, n) \in X\}]^{1+}} \quad (15)$$

where  $[x]^{1+} = x$  is only if  $x \geq 1$ , otherwise  $[x]^{1+} = 1$ .

Further, the delay in distributing video content is discussed. Let  $\tau_{n0}$  and  $\tau_{nn'}$  be the unit delay when edge server  $n$  obtains video from remote servers and edge server  $n'$  respectively. Generally,  $\tau_{n0} \gg \tau_{nn'}$ . It should be noted that because of the limited network bandwidth between edge servers, the network topology between  $\tau_{nn'}$  and edge servers is highly correlated. Let  $\tau_T$  be the delay of video forwarding operations. The delays under different distribution strategies are given below.

- 1) For the distribution mode shown in Fig. 4(a), the content access delay is  $\tau_1^n = 0$ , that is, it can be directly transmitted to users without buffering from other servers;



**Fig. 4** All possible multi-edge collaborative computing video caching and distribution models

- 2) For the distribution mode shown in Fig. 4(b), the content access delay is

$$\tau_2^n = \tau_T b_n^{sm}(t) \quad (16)$$

- 3) For the distribution mode shown in Fig. 4(c), the content access delay is

$$\tau_3^n = \sum_{n \neq n'} r_{sm} l_s \tau_{nn'}^{sm} c_{nn'}^{sm}(t) \quad (17)$$

- 4) For the distribution mode shown in Fig. 4(d), the content access delay is

$$\tau_4^n = \sum_{n \neq n'} (r_{sm} l_s \tau_{nn'}^{sm} + \tau_T) d_{nn'}^{sm}(t) \quad (18)$$

- 5) For the distribution mode shown in Fig. 4(e), the content access delay is

$$\tau_5^n = \sum_{n \neq n'} \sum_{m'=1}^{m-1} (r_{sm'} l_s \tau_{nn'} + \tau_T) e_{nn'}^{smm'}(t) \quad (19)$$

where  $m' > m$  (the same below);

- 6) For the distribution mode shown in Fig. 4(f), the content access delay is

$$\tau_6^n = \sum_{n \neq n'} \sum_{m'=1}^{m-1} (r_{sm} l_s \tau_{nn'} + r_{sm'} l_s \tau_{nn'} + \tau_T) f_{nn'}^{sm}(t) \quad (20)$$

- 7) For the distribution mode shown in Fig. 4(g), the content access delay is

$$\tau_7^n = \sum_{n \neq n'} \sum_{n'' \neq n' \neq n} \sum_{m'=1}^{m-1} (r_{sm} l_s \tau_{nn''} + r_{sm'} l_s \tau_{n''n'} + \tau_T) g_{nn'n''}^{smm'}(t) \quad (21)$$

- 8) For the distribution mode shown in Fig. 4(h), the content access delay is

$$\tau_8^n = r_{sm} l_s \tau_{n0} h^{sm}(t) \quad (22)$$

Therefore, for a certain bit rate video, the content access delay is

$$\tau_\Sigma^n = \sum_{i=1}^8 \tau_i^n \quad (23)$$

Moreover, since the video popularity follows Zipf distribution [31], that is, the video service requested by users often has a positive correlation with popularity. Therefore, edge servers often choose more popular videos for caching in order to shorten the time delay when acquiring from remote servers. Thus, the following QoE function is defined in this paper

$$J_n = (\tau_\Sigma^n)^{\gamma_s} \quad (24)$$

where  $\gamma_s$  is the popularity of video  $v_s$ .

## Constraints

Considering the existence of “cache before transmission”, “transcoding high bit rate version to low bit rate version” principles of videos, the limited computing and storage resources and the uniqueness of distribution strategies, video cache and content distribution constraints in the actual process can be divided into the following five categories.

### 1) Execution order constraints

All video requests from any user first need to verify whether the video file is cached on edge servers. Thus, the binary decision variable  $\delta_n^{sm} = 1$  defined as follows indicates that there are  $m$  bit rate copies of  $s$  video files on edge server  $n$ , otherwise it is 0. Therefore, there is

$$a_n^{sm}(t) \leq \delta_n^{sm} \quad (25)$$

$$c_{nn'}^{sm}(t) \leq \delta_{n'}^{sm} \quad (26)$$

$$e_{nn'}^{smm'}(t) \leq \delta_{n'}^{sm'} \quad (27)$$

$$f_{nn'}^{smm'}(t) \leq \delta_n^{sm'} \quad (28)$$

$$g_{nn'n''}^{smm'}(t) \leq \delta_{n'}^{sm'} \quad (29)$$

### 2) Transcoding order constraints

For distribution modes 2 and 4 involving transcoding operations, there is a unidirectional operation for transcoding from a high bit rate version to a low bit rate version. So there is

$$b_n^{sm}(t) \leq \min \left\{ 1, \sum_{m'=1}^{m-1} \delta_n^{sm'} \right\} \quad (30)$$

$$d_{nn'}^{sm}(t) \leq \min \left\{ 1, \sum_{m'=1}^{m-1} \delta_{n'}^{sm'} \right\} \quad (31)$$

### 3) Edge server storage capacity constraints

For edge server  $n$ , the size of video files it stores will not exceed its storage limit, that is

$$\sum_{n \in N} \sum_{m \in M} \delta_n^{sm} r_{sm} l_s \leq C_n \quad (32)$$

where  $C_n$  is the storage capacity of edge server  $n$ .

### 4) Edge server computing capacity constraints.

Under the multi-edge server collaboration framework, it is considered that each edge server not only needs to



process the video transcoding operations of users connected to its single hop, but also assists other edge servers in transcoding operations. And all video transcoding operations should not exceed the maximum capacity they can handle. Let the number of requests for video  $v_{sm}$  from edge server  $n$  be  $N_n^{sm}(t)$ .  $\beta_n$  is the unit bit transcoding time of edge server  $n$ , and  $T_n^{\max}$  is the maximum computing delay of edge server  $n$ . So there are:

$$\beta_n \left( \sum_{n \in N} \sum_{m \in M} N_n^{sm}(t) b_n^{sm}(t) + \sum_{n' \neq n} \sum_{m'=1}^{m-1} e_{nn'}^{smm'}(t) + \sum_{n' \neq n} \sum_{n \in N} \sum_{m \in M} N_{n'}^{s'm}(t) d_{nn'}^{sm}(t) + \sum_{m'=1}^{m-1} \left( f_{nn'}^{smm'}(t) + \sum_{n'' \neq n' \neq n} g_{nn'n''}^{smm'}(t) \right) \right) \leq T_n^{\max} \quad (33)$$

#### 5) Unique constraint of distribution strategy

When any user's video request arrives, the control center can only give a distribution strategy to respond to the user's request, namely

$$\begin{aligned} & a_n^{sm}(t) + b_n^{sm}(t) + \sum_{n' \neq n} \left( c_{nn'}^{sm}(t) + d_{nn'}^{sm}(t) + \sum_{m'=1}^{m-1} \left( e_{nn'}^{smm'}(t) + f_{nn'}^{smm'}(t) + \sum_{n'' \neq n' \neq n} g_{nn'n''}^{smm'}(t) \right) \right) \\ & + h^{sm}(t) = 1 \end{aligned} \quad (34)$$

Thus, the multi-edge collaborative computing video caching and content distribution strategy mentioned in this paper can be modeled as the following integer optimization model with constraints:

$$\begin{aligned} & \text{(Objective): } \min J_n \\ & \text{Constraints: formula (25) – (34)} \end{aligned}$$

## Optimization problem solving

### Problem analysis

Integer optimization is an NP-hard problem [32], and so far no effective general method has appeared. Often, heuristic search algorithms [33, 34] and branch and bound methods [35] are needed to approximate the global optimal solution. Besides, the model built in Section 3.3 is more accurately called a random integer optimization problem. This is because the exact number of user video requests,  $N_n^{sm}(t)$ , cannot be obtained in advance. However, on a longer time scale, the number  $N_n^{sm}(t)$  of user requests exhibits a self-similar characteristic. Therefore, this paper proposes the following strategies:

- 1) First, the number  $\{N_n^{sm}(t)\} (m = 1, 2, \dots, m)$  of historical user requests constitutes a set, and each element  $N_n^{sm}(t)$  in the set satisfies:

$$\min\_N_n^{sm}(t) \leq N_n^{sm}(t) \leq \max\_N_n^{sm}(t) \quad (35)$$

Changes in the number of video requests from a small range of users will not have a significant impact on QoE of all users. Thus, without loss of generality, the number of running scenarios of edge servers can be set as:

$$Num_n = \left\lceil \frac{\max\_N_n^{sm}(t) - \min\_N_n^{sm}(t)}{\lambda_n} \right\rceil \quad (36)$$

where  $\lceil \cdot \rceil$  is rounded up and  $\lambda_n$  is the division interval. Therefore, the running scene can be numbered 1, 2, ...,  $Num_n$  according to the number  $N_n^{sm}(t)$  of user video requests in ascending order. Therefore, for  $N$  edge servers, the number of possible running scenarios is  $\prod_{n \in N} Num_n$ .

It should be pointed out that the division operation can effectively reduce the number of running scenarios, thereby reducing the difficulty of solving optimization model.

- 2) For each running scenario,  $N_n^{sm}(t)$  is taken as the upper bound of current interval. Therefore, the random integer optimization model in Section 3.3 degenerates into the classic integer programming model. However, considering that there are too many decision variables involved in this problem and there is a competitive relationship (such as the consumption of computing resources by local decoding and collaborative decoding), it may take too long to solve using classic solvers such as CPLEX. Thus, this paper further adopts pyramid evolution intelligent evolution algorithm based on optimal cooperation to solve this problem.

According to the above two steps, the best video caching and content distribution strategy in different scenarios can be formed and stored in the control center in an offline form. In actual application, the best caching and distribution strategy can be selected in the form of table look-up only by the number of current user video requests. It should be pointed out that the advantage of this scheme is that it avoids extra calculation time overhead introduced when solving the online optimal strategy. The problem complexity of table lookup method is  $O(n)$ , and it still has high computational efficiency even when the number of running scenarios is large.

### Pyramid group intelligent evolutionary solution algorithm based on optimal cooperation

The block diagram of group intelligent evolutionary solution algorithm based on pyramid structure is shown in Fig. 5. The global optimal solution is mainly composed of the following formulas:

$$[F', I] = \text{sort}(F) \quad (37)$$

$$X = [X_4, X_3, X_2, X_1] \quad (38)$$

$$\sum_{p=1}^4 \text{length}(X_p) = \text{length}(X) \quad (39)$$

$$\text{length}(X_p) < \text{length}(X_{p+1}), i = 1, 2, 3 \quad (40)$$

$$x_p^{q+1} = x_p^q + R_p^q \times \sigma \quad (41)$$

$$R_p^q \leq R_{p+1}^q, P = 1, 2, 3 \quad (42)$$

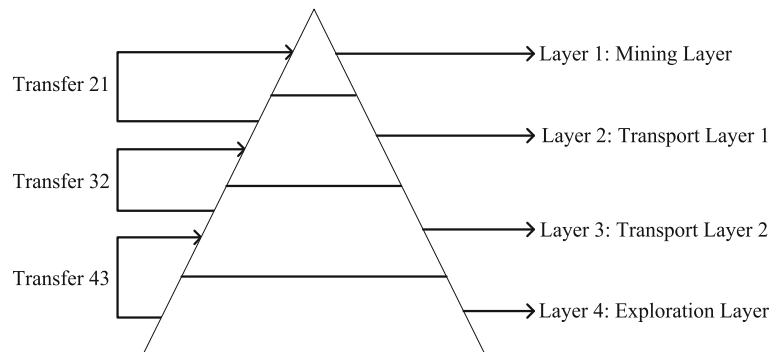
$$R_p^{q+1} = R_p^0 \times \mu^q \quad (43)$$

$$x = x_p^{q+1} + \theta(x_p^{q+1} - x_p^q) \quad (44)$$

Equation (26) sorts in ascending order according to the size of fitness value  $F$  of population  $X$  to obtain the sorted fitness value  $F'$  and index  $I$ . The sorted group  $X_1$  is divided into four parts according to formula (38) and formula (39). The number of individuals in each part satisfies Eq. (40), that is, the number of groups  $X_1$  composed of excellent individuals is the smallest, and the number of groups  $X_4$  composed of the worst individuals is the largest. Group  $X_1$  is called mining layer,  $X_2$  and  $X_3$  are called transfer layers, and  $X_4$  is called exploration layer. The individual  $x_p^q$  of each layer updates the group according to different search neighborhoods  $R_p^q$  according to Eq. (41) in each layer. The neighborhood size of each layer satisfies formula (42), that is, the mining layer generates a search step according to random number  $\sigma$  between  $[-1, 1]$  in the smaller neighborhood. The point

is to complete the mining work of population, and the exploration layer is to mine potential outstanding individuals in a large neighborhood. As the number of iterations  $q$  progresses, the population of each generation gradually approaches the global optimal solution. Therefore, the search radius  $R_p^q$  of each layer is also adaptively updated with contraction factor  $0 < \mu < 1$  according to Eq. (43), thereby improving the optimization efficiency. After generating new individuals in each layer, the algorithm collaborates between layers, that is, the excellent individuals in exploration layer and the excellent individuals in transfer layer are transferred to mining layer and exploration layer respectively. The transferred individuals are cultivated in receiving layer, and these individuals are accelerated according to formula (44) with the acceleration step  $\theta$  along the direction for generation of new individuals to obtain accelerated individual  $x$ .

The standard PES algorithm includes two kinds of collaboration, one is layer-to-layer group collaboration, which strengthens the communication between populations. One is the collaboration between individuals in the layer and parent individuals. Although parent individuals have a certain role in guiding the generation of new individuals in the offspring, it is not very helpful in producing excellent individuals. This will cause the convergence speed of algorithm to be slow and affect the operating efficiency of algorithm application. Particle swarm algorithm completes the updating of population by the individual extreme value and global extreme value, which has the advantage of fast convergence speed [36]. This update rule just makes up for the lack of cooperation among individuals in pyramid structure intelligent evolution algorithm. This paper integrates this idea into the standard pyramid structure intelligent evolution algorithm. By selecting the cooperation with the optimal individual of population in the current generation layer and optimal individual of the entire population to complete the update for each layer of individuals. A pyramid-based intelligent evolution algorithm based on



**Fig. 5** Pyramid structure framework

the optimal cooperation strategy is proposed, its update rules for each layer of individuals are as follows:

$$x_p^{q+1} = x_p^q + rand \times R_p^q \times \left[ w(pBest_p^q - x_p^q) + (1 - w)(gBest^q - x_p^q) \right] \quad (45)$$

where  $x_p^{q+1}$  is the new individual produced by the  $p$  layer of the  $q$  generation, and  $x_p^q$  is the parent individual of the  $p$  layer. *rand* refers to a random number between [0,1],  $R_p^q$  is the search radius of current generation.  $pBest_p^q$  is the optimal individual of the  $p$  layer of the  $q$  generation, and  $gBest^q$  is the optimal individual of entire population for the  $q$  generation.  $0 < w < 1$  reflects the size of the new individual's bias toward individual's  $pBest_p^q$  direction, and  $1 - w$  reflects the degree of bias toward  $gBest^q$  direction.

Based on the strategy of optimal cooperation, the parent individual can be searched and explored along the direction of joint force generated by the individual extreme value and global extreme value. This not only strengthens the mutual cooperation among individuals in the layer, but also establishes the connection between individuals in each layer and the globally optimal individual. Under the strategy of layer-to-layer collaboration and optimal cooperation among individuals, the convergence speed of pyramid structure intelligent evolution algorithm is accelerated, and the optimization efficiency of this algorithm is improved.

Therefore, in this paper, the steps of applying pyramid structure intelligent evolution algorithm to solve the optimal strategy for video caching and content distribution are as follows.

Step 1: Initialize parameters, set the maximum number of iterations  $I_{\max}$  and population size  $G$ , and initial search radius  $R_p^q$  of each layer;

Step 2: Randomly generate the initial population  $\{x^0\}$  of population size  $G$  and juxtapose the number of iterations  $t = 1$ ;

Step 3: Calculate the fitness value of population  $\{x^q\}$  (i.e. the current QoE function). According to the size of fitness value  $J(x^q)$ , the population is divided into four sub-populations  $X^q(q = 1, 2, 3, 4)$ , and the contemporary individual extremum  $pBest_p^q$  and global extremum  $pBest_t^q$  are recorded;

Step 4: According to Eq. (34), generate a new individual  $x_p^{q+1}$  for individual  $x_p^q$  of the  $q(q = 1, 2, 3, 4)$  layer. And select the corresponding number of individuals from each layer to pass to the upper layer, and complete the cultivation according to formula (34). Integrate the generated new individuals, passing individuals and

parent individuals, and select individuals with the corresponding number of populations as updated group  $\{x_p^{q+1}\}$  of this layer.

Step 5: Integrate updated population  $\{x_p^q\}$  of each layer into new population  $\{x_p^{q+1}\}$  generated by the  $q$  generation.

Step 6: Determine the termination iteration condition. When the number of iterations reaches the maximum number of iterations  $t = I_{\max}$ , then output  $gBest^q$  and algorithm stops. Otherwise, set  $t = t + 1$  and update the search radius  $R_p^q$  of each layer.

## Experimental results and discussion

### Experimental setup

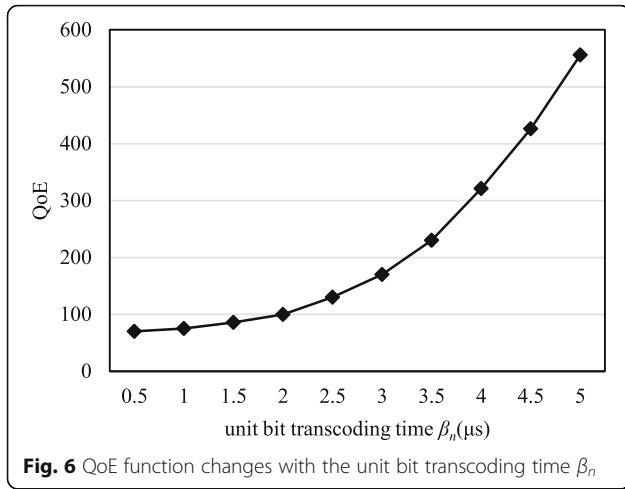
On the hardware configuration of i5-3230M CPU, 10GB RAM, 1TBHD + 512GB SSD, we use MATLAB R2016a version and Opnet network simulation software to simulate and verify the performance of proposed algorithm. The parameters used by Logistic chaotic-QCA encryption algorithm are:  $\omega_{01} = \omega_{03} = 0.28$ ,  $\omega_{02} = 0.7$ ,  $\omega_{04} = 0.5$ ,  $\Psi = 3.9$ . The video file type is 20, and the duration is evenly distributed within 5 min–45 min. Each video contains 4 versions with bit rates of 1.25, 1, 0.75 and 0.5 times the original rate (2Mbps). Without loss of generality, let the user arrival rate under each edge server follow the Poisson distribution of 50/min, while the video popularity follows Zipf distribution with parameters [0.3, 0.9]. And we suppose that each version of the same video has the same request probability. Also set  $\tau_{n0} = 100ms$ ,  $\tau_{nn}$  is evenly distributed in [5, 50] ms interval, the number of edge servers is 10. The unit bit transcoding time  $\beta_n$  reference value is uniformly set to 2  $\mu s$ , and the edge server cache capacity  $\alpha$  reference value is set to 50. The maximum calculation delay  $T_n^{\max}$  allowed by servers is set to 150 ms.

In addition, comparison algorithms are selected as the JCPNonCo scheme proposed in reference [18], the CCNonP scheme proposed in reference [27] and the APCP-OptRs scheme proposed in reference [28].

### Parameter performance analysis

#### Transcoding capability of edge servers

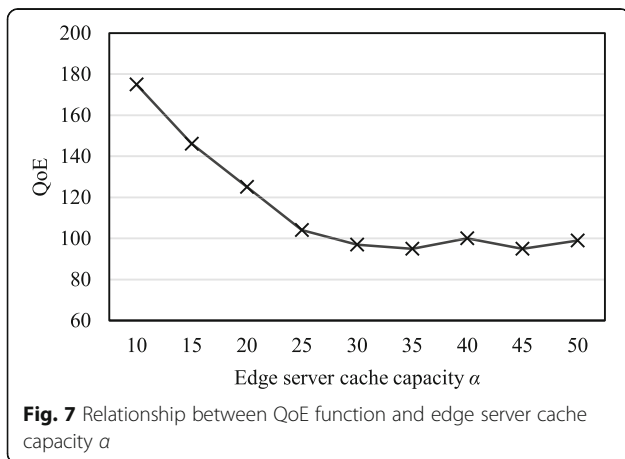
The transcoding capability of edge server  $\beta_n$  is closely related to the CPU performance of edge server itself. In other words,  $\beta_n$  reflects the performance of edge servers to a large extent. Besides, the smaller  $\beta_n$  is, the higher CPU performance is, and the stronger the transcoding capability for videos is. Figure 6 shows the change relationship of QoE function when the unit transcoding time  $\beta_n$  changes from 0.5  $\mu s$  in steps of 0.5  $\mu s$  to 5  $\mu s$ . The overall change trend is that the value of QoE function gradually increases with the increase of  $\beta_n$ . However, when  $\beta_n$  changes from 0.5  $\mu s$  to 3  $\mu s$ , the QoE



change range is relatively slow, after  $\beta_n$  exceeds  $3 \mu s$ , the QoE function value rises sharply. The reason is that when  $\beta_n$  is smaller, the transcoding capability of each server is stronger. Therefore, there are more options for video caching and content distribution strategies. When  $\beta_n$  is large, due to the long processing delay of a single server, it is no longer the optimal option to provide coordinated transcoding capabilities to other edge servers. Edge servers tend to cache videos at all bit rates to meet users' video experience. However, it should be noted that lower transcoding time means higher performance requirements of CPU, which will cause problems of rising economic costs.

#### Cache capacity of edge server

Analyzing the impact of edge server cache capacity (that is,  $\alpha$  in Eq. (2)) on QoE performance, Fig. 7 shows the change in the value of QoE function when  $\alpha$  changes from 10 to 50 and the step size is 5. When  $\alpha$  changes from 10 to 35, QoE gradually decreases as the value of  $\alpha$  changes. However, when the value of  $\alpha$  exceeds 35, the

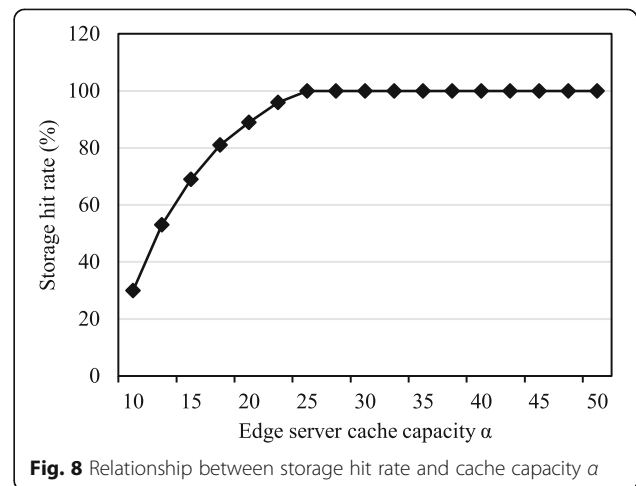


QoE performance no longer changes significantly. Further define the storage hit ratio as the ratio of sum for video types requested by users to the sum of video types locally cached by edge servers. Figure 8 shows the relationship between the storage hit ratio and cache capacity  $\alpha$ . It can be seen that as the cache capacity of edge servers gradually increases, the storage hit rate gradually increases. When  $\alpha$  exceeds 25, the storage hit rate increases to 100% and does not change with the increase of cache capacity. This is because when the cache capacity of edge servers is large, more resource copies can be stored on local edge servers. It no need to request resources from the cloud or adjacent edge servers or rely on other edge server nodes to assist in transcoding. Thus, the numerical results show that increasing the cache capacity of edge servers can only improve the QoE of users to a limited extent, and at the same time will bring about the problem of economic cost.

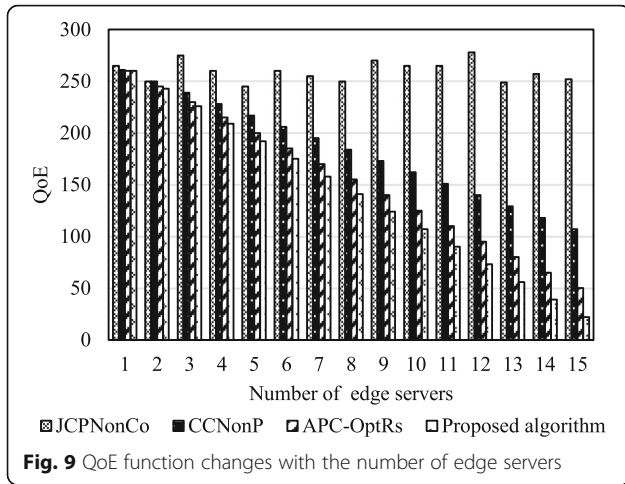
#### QoE comparison of different strategies

Set Zipf distribution parameter  $\gamma_s = 0.75$ . Figure 9 shows the value of QoE under different algorithms when the number of edge servers changes from 1 to 15.

As can be seen from Fig. 9, when the number of edge servers is only 1, there is almost no difference in the QoE values of all algorithms. This is because there is no edge server collaboration mechanism at this time, and all video caching and transcoding operations are performed on the same server. With the increase in the number of edge servers, the collaboration mechanism among multiple edge servers is not considered in reference [18]. That is, each edge server only exchanges data with a remote server and does not cooperate with other edge servers, so QoE value is maintained at about 260. References [27, 28] and the algorithm proposed in this paper all have a cooperative relationship due to their existence. Therefore, the QoE value gradually decreases as

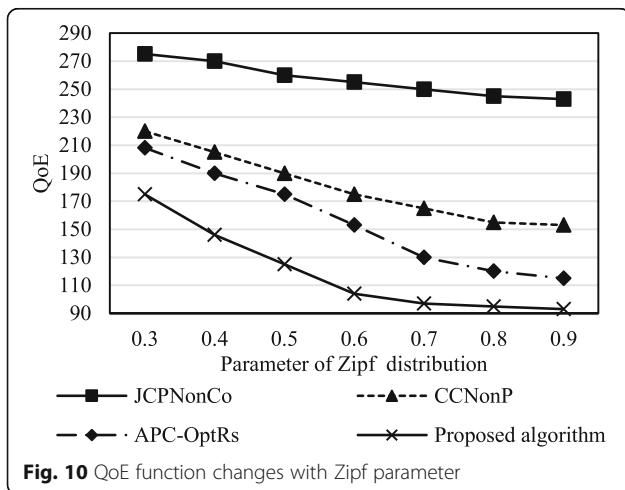






the number of edge servers increases. However, compared with the two-hop mode involving three edge servers in this paper, the collaborative transcoding service is not provided between edge servers in reference [27] and only edge servers in video cache is considered in reference [28]. Single-hop mode with transcoding only between two edge servers. Thus, the proposed algorithm provides more feasible solutions for caching and content distribution. Correspondingly, the decrease of QoE value is improved by 22.84% and 14.40% compared with the algorithms proposed in reference [27] and reference [28].

Suppose the number of edge servers is 10, and Fig. 10 shows QoE function value when the Zipf distribution parameter  $\gamma_s$  changes. No loss of generality, the Zipf distribution parameter changes from 0.3 to 0.9 in steps of 0.1. Regardless of this method, QoE function shows a similar trend. That is, as the distribution parameters change, QoE function is decreasing. This shows that caching popular videos on the edge server can



significantly improve users' video request satisfaction. However, when  $\gamma_s$  changing from 0.3 to 0.9, the algorithm proposed in this paper has a decrease rate of 35.22%, 16.40% and 2.16% compared with references [18, 27, 28]. This is because this paper additionally considers a two-hop collaboration scenario among three edge servers when designing a video content distribution strategy, thereby providing additional strategy options.

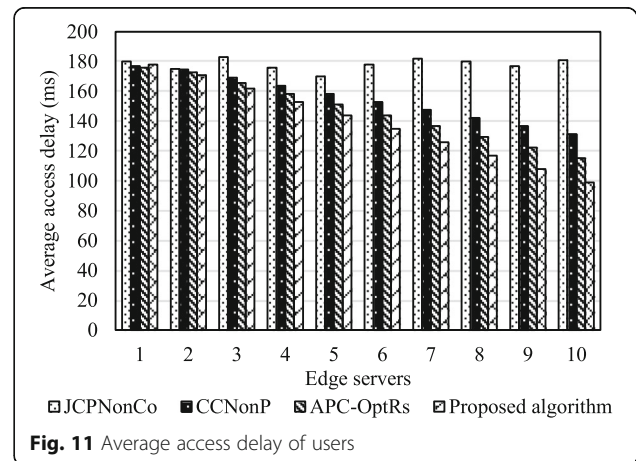
### Comparison of average access delay of different strategies

Further, the relevant algorithm performance test is conducted in Opnet environment. Figure 11 shows the average access delay of users under each edge server under different video caching and content distribution strategies when edge server is 10.

It can be seen from Fig. 11 that the algorithm proposed in reference [18] does not consider collaboration between edge servers, so the average access delay of users remains unchanged and is the largest of the four methods. References [27, 28] and our proposed algorithm all have a cooperative relationship, so the average user access delay is shorter than the method described in reference [18]. The proposed algorithm further considers the video caching and transcoding operations in a three-hop cooperative manner among three edge servers, which is essentially the expansion and derivation of references [27, 28]. This provides additional decision-making options for control center, thereby making the multi-edge server work in a more efficient state. Therefore, the access delay of users is shortened by 45.21%, 24.66% and 14.06% respectively, compared with references [18, 27, 28], and the average shortening is 27.98%.

### Conclusion

This paper proposes a video caching and content distribution mechanism in a multi-edge collaborative computing environment. Based on the definition of user QoE,



the video caching and content distribution mechanism is modeled as a random integer programming problem, and is solved using an intelligent evolution algorithm based on optimal cooperation pyramid structure. Experimental examples show that our proposed algorithm enriches the decision-making options for video caching and content distribution strategies due to considering two-hop collaboration scenarios among three edge services. Besides, it also guarantees that edge servers have high computational efficiency and objective storage space, so that it has a better engineering application in practical background.

The established model is offline working mode, thus we will further study the real-time online multi-edge server collaboration video caching and content distribution strategy in the future.

#### Acknowledgements

The authors thank the editor and anonymous reviewers for their helpful comments and valuable suggestions. I would like to acknowledge all our team members.

#### About the authors

Zhi Liu, Post-Graduate, Title of Network Engineer. At present, He is mainly responsible for the information construction of continuing education in Hunan Agricultural University. His research interests include complex network and intelligent information processing.

Bo Qiao, Ph.D. of Agriculture Science, Lecturer. Graduated from Hunan Agricultural University in 2019. Worked in Hunan Agricultural University. His research interests include agricultural knowledge graph and natural language processing.

Fang Kui, Ph.D. of Computer Science, Professor. Graduated from National University of Defense Technology in 2000. Worked in Hunan Agricultural University. His research interests include Graphic image processing and agricultural information engineering.

#### Authors' contributions

The main idea and experimental guidance of this article is done by Kui Fang. The program design, coding and article writing are mainly completed by Zhi Liu and Bo Qiao. The author(s) read and approved the final manuscript.

#### Funding

This work was supported by the Natural Science Foundation of Hunan Province, China (No.2019JJ40133).

#### Availability of data and materials

All the data and materials in this article are available.

#### Competing interests

The authors of this article declare that they have no competing interests.

#### Author details

<sup>1</sup>Hunan Engineering Technology Research Center of Rural & Agricultural Informationization, Hunan Agricultural University, Changsha 410128, Hunan, China. <sup>2</sup>College of Information and Intelligence, Hunan Agricultural University, Changsha 410128, Hunan, China.

Received: 23 June 2020 Accepted: 17 September 2020

Published online: 02 October 2020

#### References

- Bairagi AK, Abedin SF, Tran NH et al (2018) QoE-enabled unlicensed spectrum sharing in 5G: a game-theoretic approach[J]. *IEEE Access* 6:50538–50554
- Schwarzmann S, Marquezan C C, Bosk M, et al. Estimating video streaming QoE in the 5G architecture using machine learning[C]// the 4th internet-QoE workshop. 2019
- Segura-Garcia J, Felici-Castell S, Garcia-Pineda M (2018) Performance evaluation of different techniques to estimate subjective quality in live video streaming applications over LTE-advance mobile networks[J]. *J Netw Comput Appl* 107(1):22–37
- Qi L, Zhang X, Dou W, Ni Q (2017) A distributed locality-sensitive hashing based approach for cloud service recommendation from multi-source data. *IEEE J Selected Areas Commun* 35(11):2616–2624
- Gao G, Zhang H, Hu H et al (2018) Optimizing quality of experience for adaptive bitrate streaming via viewer interest inference[J]. *IEEE Trans Multimed* 20(12):3399–3413
- Deng KY, Yuan L, Wan Y et al (2018) Optimized cross-layer transmission for scalable video over DVB-H networks[J]. *Signal Process Image Commun* 63(9):81–91
- Qi L, Dou W, Wang W, Li G, Yu H, Wan S (2018) Dynamic Mobile crowdsourcing selection for electricity load forecasting. *IEEE Access* 6:46926–46937
- Qi L, Dou W, Hu C, Zhou Y, Yu J (2015) A context-aware service evaluation approach over big data for cloud applications. *IEEE Trans Cloud Comput*. <https://doi.org/10.1109/TCC.2015.2511764>
- Zhu H (2019) A simplified deniable authentication scheme in cloud-based pay-TV system with privacy protection[J]. *Int J Commun Syst* 32(11):3967–3979
- Yan H, Li X, Wang Y et al (2018) Centralized duplicate removal video storage system with privacy preservation in IoT[J]. *Sensors* 18(6):1814–1826
- Sugathapala I, Hanif MF, Lorenzo B et al (2018) Topology adaptive sum rate maximization in the downlink of dynamic wireless networks[J]. *Commun IEEE Trans* 66(8):3501–3516
- Dijiang H, Huijun W (2018) Edge clouds-pushing the boundary of Mobile clouds[J]. *Mobile Cloud Computin[M]*. Elsevier:153–176
- Aral A, Ovatman T (2018) A decentralized replica placement algorithm for edge computing[J]. *IEEE Trans Netw Serv Manag* 17(2):516–529
- Ma X, Lin C, Zhang H et al (2018) Energy-aware computation offloading of IoT sensors in cloudlet-based mobile edge computing[J]. *Sensors* 18(6):1945–1953
- Psaras I, Saino L, Pavlou G (2014) Revisiting resource pooling: the case for in-network resource sharing[C]// HotNets-XIII: 13th ACM workshop on hot topics in networks, 27–28 October 2014. ACM, Los Angeles
- Sung-Yen C, Chin-Feng L, Yueh-Min H (2012) Dynamic adjustable multimedia streaming service architecture over cloud computing[J]. *Comput Commun* 35(15):1798–1808
- Gao G, Wen Y, Cai J (2017) vCache: supporting cost-efficient adaptive bitrate streaming[J]. *IEEE Multimed* 24(3):19–27
- Pedersen H, Dey S (2016) Enhancing mobile video capacity and quality using rate adaptation, RAN caching and processing[J]. *IEEE/ACM Trans Networking* 24(2):996–1010
- Zhang W, Wen Y, Chen Z, et al. QoE-driven cache management for HTTP adaptive bit rate (ABR) streaming over wireless networks[C]// global communications conference. IEEE, 2013
- Pääkkönen P, Heikkinen A, Aihkialo T (2019) Online architecture for predicting live video transcoding resources[J]. *J Cloud Comput* 8(9):1–24
- Ivan S, Mirko S, Lea SK (2018) Game categorization for deriving QoE-driven video encoding configuration strategies for cloud gaming[J]. *ACM Trans Multimed Comput Commun Appl* 14(3):1–24
- Ananthanarayanan G, Bahl P, Bodik P et al (2017) Real-time video analytics: the killer app for edge computing[J]. *Computer* 50(10):58–67
- Long C, Cao Y, Jiang T et al (2017) Edge computing framework for cooperative video processing in multimedia IoT systems[J]. *IEEE Trans Multimed* 20(5):1126–1139
- Shuping P, Oscar FJ, Khodashenas PS et al (2017) QoE-oriented mobile edge service management leveraging SDN and NFV[J]. *Mob Inf Syst* 2017:1–14
- Shun-Ren Y, Yu-Ju T, Chen-Chia H et al (2019) Multi-access edge computing enhanced video streaming: proof-of-concept implementation and prediction/QoE models[J]. *IEEE Trans Veh Technol* 68(2):1888–1902
- Desheng W, Yanrong P, Xiaoqiang M et al (2018) Adaptive wireless video streaming based on edge computing: opportunities and approaches[J]. *IEEE Trans Serv Comput* 12(5):685–697
- Li C, Toni L, Zou J et al (2018) QoE-driven mobile edge caching placement for adaptive video streaming[J]. *IEEE Trans Multimed* 20(4):965–984
- Tran TX, Pompili D (2019) Adaptive bitrate video caching and processing in mobile-edge computing networks[J]. *IEEE Trans Mob Comput* 18(9):1965–1978



29. Das B, Paul AK, De D (2019) An unconventional arithmetic logic unit design and computing in actin quantum cellular automata. *Microsyst Technol*. <https://doi.org/10.1007/s00542-019-04590-1>
30. Pavlic J, Burkeljca J (2019) FFmpeg based coding efficiency comparison of H.264/AVC, H.265/HEVC and VP9 video coding standards for video hosting websites[J]. *Int J Comput Appl* 182(37):1–8
31. Osman AM, Osman NI (2018) A comparison of cache replacement algorithms for video services[J]. *Int J Comput Sci Inf Technol* 10(2):95–111
32. Zhou YF, Yu HX, Li Z, Su JF, Liu CS (2020) Robust optimization of a distribution network location-routing problem under carbon trading policies [J]. *IEEE Access* 8(1):46288–46306
33. Yufeng Z, Na C (2019) The LAP under facility disruptions during early post-earthquake rescue using PSO-GA hybrid algorithm[J]. *Fresenius Environ Bull* 28(12):9906–9914
34. Jiafu S, Yu Y, Tao Y (2018) Measuring knowledge diffusion efficiency in R&D networks[J]. *Knowl Manag Res Pract* 16(2):1–12
35. Jian J, Guo Y, Jiang L et al (2019) A multi-objective optimization model for green supply chain considering environmental benefits[J]. *Sustainability* 11(21):5911–5931
36. Sierra MR, Coello Coello AC (2005) Improving PSO-based multi-objective optimization using crowding, mutation and E-dominance[J]. *Lect Notes Comput Sci* 3410:505–519

## Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

**Submit your manuscript to a SpringerOpen<sup>®</sup> journal and benefit from:**

- Convenient online submission
- Rigorous peer review
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

---

Submit your next manuscript at ► [springeropen.com](https://www.springeropen.com)