IEEE Open Journal of the Computer Society

Received 4 September 2021; accepted 5 October 2021. Date of publication 12 October 2021;
date of current version 3 November 2021. The review of this paper was arranged by Guest Editors Zhi Liu, Stephan Sigg, Xianfu Chen, Ruozhou Yu, and Takuya Fujihashi.

Digital Object Identifier 10.1109/OJCS.2021.3119366

Optimal Resource Allocation for Multimedia Applications Offloading in Mobile Edge Computing

GUOLONG CHEN[1,2], LIANG ZHAO[3] (Member, IEEE), XIANWEI LI[1], FUQI ZHAO[1], AND XIAOJIAN ZENG[4]

[1] School of Computer and Information Engineering, Bengbu University, Bengbu 236000, China
[2] School of Information Engineering, Suzhou University, Suzhou 234000, China
[3] School of Computer Science, Shenyang Aerospace University, Shenyang 110000, China
[4] School of Computer Science, Yangtze University, Jingzhou, China

CORRESPONDING AUTHOR: XIANWEI LI (e-mail: lixianwei163@163.com)

This work was supported in part by the Start up funds for scientific research of high level talents of Bengbu University under Grant BBXY2020KYQD02, in part by the Major Project of Natural Science of Education Department of Anhui Province under Grant KJ2014ZD31, in part by Key Research and Development Projects in Anhui Province under Grant 202004a05020043, in part by the Funding Project for the cultivation of outstanding talents in Colleges and Universities under Grant gxyqZD2021135, and in part by the General Natural Science Project of Bengbu University under Grant 2017ZR12.

ABSTRACT Thanks to the development of the technologies in the wireless communications and Internet of Things (IoT), the adoption of mobile devices is growing rapidly. Accordingly, the number of multimedia applications like face recognition and augmented reality generated from various mobile devices is growing at an unprecedented rate. The processing of these multimedia applications needs a lot of computation resources and has to be processed as quickly as possible. However, as these mobile devices have limited computation resources, the undesirable response delay will occur. By offloading the multimedia applications to the edge cloud close to the access point (AP) or the cellular base station (BS), mobile edge computing (MEC) is considered as a prospective approach to improve the quality of service (QoS) and enhance the computing capacity of mobile devices. Multimedia applications offloading in a MEC system are studied in this paper. The objective of the studied problem is to minimize the execution delay of multimedia applications of all mobile devices by allocating both the communication resource and the computing resource in the edge servers. An optimization problem is formulated and an efficient multimedia applications offloading scheme is proposed to get the solution. Simulation results are conducted to verify the proposed application offloading method, which show that there is a significant execution delay reduction.

INDEX TERMS Applications offloading, mobile edge computing, system delay, resource allocation.

I. INTRODUCTION

In the past few years, thanks to the development of the technologies in communication and the Internet of Things (IoT), the accelerated acquisition of mobile devices, taking tablets and smart phones as examples, is growing explosively. Cisco anticipated that the number of connected IoT devices by the Internet reached 50 million by the end of 2020, which means that one person holds 6 devices [1]. Accordingly, varieties of multimedia applications generated from mobile devices, such as face recognition and augmented reality, are growing at an unprecedented rate and receiving a lot of interests [2]. According to the report from Cisco, the mobile video traffic will account for roughly 79% of the world's mobile data traffic by 2022, as shown in Fig. 1 [6]. The processing of the significant amount of multimedia applications from mobile devices has strict requirements on the quality of service (QoS) [3], [7]. Furthermore, processing these multimedia applications typically requires high computation capacity [4]. However, as the resources (e.g., CPU cores and battery power) in mobile devices are insufficient, they must offload their multimedia applications to the cloud for execution [5].

Traditionally, the public clouds provide high computation capacity for the processing of delay-tolerant applications. These applications include image processing and financial

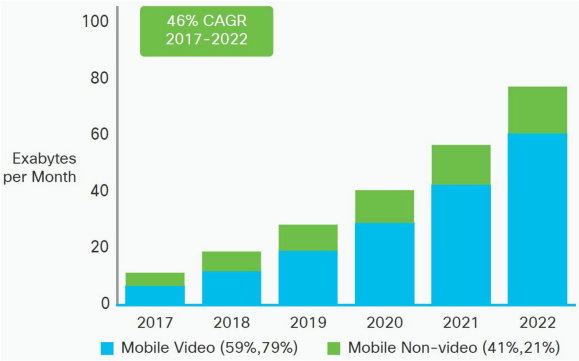


FIGURE 1. The forecast of Global Mobile Data Traffic [6].

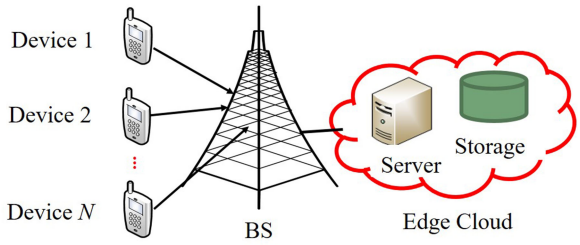


FIGURE 2. The MEC System Model [33].

analysis, which do not have real-time requirements. With the number of delay-sensitive applications increasing, especially the increasing number of multimedia applications, such as online video games and augmented reality (AR) video, it is becoming important to process them with low delay. Although there are abundant computation resources in the public clouds, the long transmission delay between the wireless devices and the data centers will be induced, which is not the best choice for processing the delay-sensitive multimedia applications.

Mobile edge computing (MEC) is introduced as a complementary to the public clouds to overcome the shortcoming of long transmission delay [8]–[10], [11]. As the illustration in Fig. 2, the edge cloud deploys computing resources close to the access point (AP) or the base station (BS) of a cellular network. Therefore, the transmission delay can be reduced, and fast data-processing services can be supplied to the mobile users. Thanks to the emergence of MEC, the mobile users can perform task offloading via BS to accomplish the execution of their tasks in the edge servers, which can significantly shorten the computation latency and save battery power.

Despite the advantages that the MEC systems have, several technical challenges need to be addressed to implement the MEC systems practically [12], [13]. First, tasks of mobile users can be divided into two types, namely partitionable and non-partitionable, according to their partitionability and dependency. Second, the performance optimization of task offloading depends highly on jointly allocating bandwidth and computing resources [14]. Third, as the future wireless network will consist of massive mobile devices, one BS will serve an enormous amount of mobile devices; how to efficiently allocate the communication resource and the computing resource for the edger servers is a challenging issue.

In this paper, task offloading for multimedia applications processing in a MEC system is investigated. The objective is to minimize the sum execution latency of mobile devices by allocating the wireless communication resources and the edger servers' computation resources. Although a great number of previous works have been paid on task offloading, there are several differences between the multimedia application offloading and these previous works. First, in the local compression model, the multimedia applications are compressed firstly in mobile devices and then transmitted to be stored in the edge cloud; Second, in the edge cloud compression model, the multimedia applications are firstly compressed and then stored. However, in the traditional task offloading, the applications are just computed without needing the storage resources.

The contributions for this paper are summed up as follows:

- In a MEC system, we study the allocation of communication and computing resources for multimedia applications offloading, the aim of which is minimizing the execution time of all mobile devices.
- We consider two application execution models, namely, local compression model and compression model of edge cloud, separately formulate the latency-minimization problem and propose a solution method for each model.
- We propose an efficient multimedia applications offloading scheme algorithm based on the solutions of latency-minimization problems for each compression models.
- Simulation results are performed to verify the theoretical analysis and the performance of the proposed multimedia applications offloading algorithm. The experiments show that the proposed algorithm can outperform the three benchmark methods in terms of execution delay.

We organize the rest of study of the paper as follows. Related work is presented and analyzed in Section II. The system model is presented in Section III. Problem formulation and the proposed solution method of the local compression model are presented in Section IV. Problem formulation and the proposed solution method of the compression in the edge cloud model are presented in Section V. In Section VI, the simulation results are conducted to verify the efficacy of the proposed algorithm. Finally, the conclusions of this paper and future research works are shown in Section VII.

II. RELATED WORK

Resource allocation and applications offloading in MEC networks have been hotly studied in the existing literature. In [15], Sun *et al.* studied the resource allocation in MEC with an edge server serving multiple IIoT devices. They proposed two dynamic pricing schemes based on two double auction schemes for the provision of edge cloud services. In [16], the authors proposed a novel machine learning task offloading framework for IIoT. Their objective is to get the total delay-minimization of sensing devices with the constraints

of the computing capacity of the devices and communication bandwidth. In [17], Jie *et al.* studied game-theoretic resource allocation in the IIoT based on fog, the problem of which is formulated as a double-stage Stackelberg game, and is solved by the proposed scheme. In [18], Zhao *et al.* studied full task offloading for minimizing the energy consumption of smart mobile devices by jointly allocating of radio resources and computational resources of edge servers. In [19], Kabir and Masouros studied full task offloading to investigate the energy consumption and latency tradeoff in a MEC system with full-duplex. They formulated two optimization problems for the two objectives. Two schemes were proposed to solve the formulated problems. Chen *et al.* studied computation offloading in a multi-user MEC system [8]. The computation offloading decision problem is formulated as a game with multiple users, which was solved by proposing an offloading algorithm. In [20], Dink *et al.* studied task offloading considering the scenarios of both single mobile device and multiple devices. They aimed to minimize the latency of task execution and the energy that mobile devices consumed by optimizing the CPU frequency. In [22], Chen *et al.* proposed a task offloading framework combining the deep imitation learning (DIL) and knowledge distillation (KD). In [23], Yang *et al.* studied task offloading decisions, communication resource allocation, and caching decisions in a non-orthogonal multiple access (NOMA) based MEC framework. A reinforcement learning-based method was proposed to get the solution to the formulated problem. In [24], the authors studied a mobile device (MD) whose application has multiple tasks to be executed. They tried to optimize the offloading decision and CPU power allocation by proposing a deep reinforcement learning method. In [28], Chen *et al.* studied task offloading performance optimization in the virtual MEC systems under the time-varying network conditions and proposed an algorithm based on DQN. However, the allocation of communication and computation resources was not taken into consideration. In [25], Li *et al.* studied computation offloading and resource allocation in the multi-user scenario, especially the analyzed the MEC system with multiple servers. As the optimization problem is a MINP problem, based on genetic algorithms, they proposed an optimization algorithm to solve it. In [26], Zhang and Ansari studied latency minimization in the unmanned aerial vehicles (UAVs)-aided MEC network, as the IoT devices cloud have the limitation of computing resources. As the formulated UAV-MEC problem is NP-hard, three subproblems are decomposed. They proposed an approximation method with the advantage of low complexity for the solution of it. In [34], Wen *et al.* jointly studied caching, computation offloading and time allocation for minimizing the energy consumption. In [37], Chen *et al.* studied network slicing to support the demand of diverse services of mobile users without investigating computation offloading.

For the task offloading in the MEC systems, the joint allocation of communication and computing resources is not considered for many of these existing works. Kuang *et al.* studied partial task offloading and transmission power allocation to minimize the execution delay and energy consumption under power constraints of tasks for MEC systems [27]. As the formulated problem is MINP, a Lagrangian dual decomposition based framework is proposed to solve it. Wu *et al.* studied secrecy-based partial computation offloading in the scenarios of one delay-aware smart device and a group of smart devices in MEC to minimize the overall delay [29]. The put forward two efficient algorithms to solve the problems of each scenario. In [30], a cooperative computing scheme was proposed by Huang *et al.* to minimize energy consumption while maximizing the offloading data problems by jointly allocate communication and computation resources of the user and helper in a three-node MEC system, where the access point (AP) adopts a NOMA. In [31], Yang *et al.* investigated efficient resource allocation for partial task offloading to get the minimization of completion time and energy in MEC networks with NOMA. In [13], Wang *et al.* studied a multi-user MEC system with the NOMA technique for task offloading and minimizing the energy consumption by allocating the CPU frequencies and transmission power. They considered both the partial and binary offloading cases. In [32], Yu *et al.* studied the allocation of power for energy efficiency the fog computing. However, they did not consider the allocation of communication and computation resources. In [21], Gao *et al.* studied service level optimization in a MEC system by allocating transmission power. In [9], Kuang *et al.* studied a cooperative allocation of the edge-cloud resources and computation offloading to minimize delay in the mixed edge-cloud systems. But the radio resources allocated for mobile devices are considered as fixed values in this work.

In contrast to many of the existing works that either ignored the joint allocation of communication resource and the edge server's computing resource or without considering the latency minimization, we study multimedia applications offloading in a MEC system for the minimization of total latency of mobile devices. We take the allocation of the wireless communication resources and the edge servers' computing resources in the edge cloud into account.

We compare the objectives and resource optimization of our study with some related work in the MEC systems, the results of which are shown in Table 1. It is obvious that our study can overcome the shortcomings of many previous works.

III. SYSTEM MODEL

The system model is presented in this section, which consists of one edge cloud and N mobile devices, as shown in Fig. 2. These devices are connected by the edge cloud, who offers computing and storage services to them. The system model of this paper is mainly motivated by [33]. Assume that each device has a raw video that will be processed in the edge cloud. For the mobile device i, $i \in \{1, 2, \ldots, N\}$, it has an application which can be described by L_i and C_i, where L_i is the size of the raw video and C_i is the required CPU cycles to compress this application. We also assume that the same video technology is applied by all mobile devices and the system of the edge cloud, such as the MPEG4, such that all

TABLE 1. Comparing With Some Related Work

Reference	Delay	Communication	Computation
[37]	×	✓	×
[17]	×	✓	✓
[38]	×	×	✓
[39]	×	×	✓
[35]	×	×	×
[45]	×	×	✓
[44]	×	×	✓
[16]	✓	×	×
[18]	✓	×	×
[19]	✓	×	×
[22]	✓	✓	×
[40]	✓	×	×
[20]	✓	✓	×
[33]	×	✓	×
[27]	✓	×	×
[41]	✓	×	×
[43]	✓	×	×
[8]	✓	×	×
Our Work	✓	✓	✓

the videos may be compressed simultaneously in the cloud. Besides, the time of video segmentation, stitching and storage can not be considered as they are very small compared with the communication and compression delays [33]. Similar to reference [30], the system model of this paper can also be applied to a lot of practical scenarios, like the surveillance systems, where a large number of video data coming from network cameras have to be further analyzed and stored.

A. LOCAL COMPRESSION

In the model of local compression, the raw videos will be firstly compressed in mobile devices and then offloaded to be stored by the edge cloud. Therefore, for the device i, the delay of compressing L_i bits of raw video is

$$D_{i,f} = \frac{L_i}{V_i^l} \tag{1}$$

After the completion of the compression in mobile devices, the delay caused by transmitting the number of βL_i compressed videos to the edge servers is

$$D_{i,t}^l = \frac{\beta L_i}{r_i} \tag{2}$$

where $\beta \in (0, 1)$ denotes the ratio of the compressed video data size to the total raw video data size, and r_i is the achieved data rate, which is given as follows,

$$r_i = B_i \log_2 \left(1 + \frac{p_i h_i}{N_0}\right) \tag{3}$$

where B_i denotes the allocated bandwidth, p_i is the transmission power, h_i is the channel gain, and N_0 is the noise power.

Therefore, the total delay for mobile device i to finish its task in local compression is

$$D_{i,l} = D_{i,f} + D_{i,t}^l = \frac{L_i}{V_i^l} + \frac{\beta L_i}{r_i} \tag{4}$$

B. EDGE CLOUD COMPRESSION

In the model of the compression in the edge cloud, mobile devices will directly offload their applications to the edge cloud. In this case, the edge cloud will allocate computing resources to compress all the raw videos simultaneously. Therefore, the transmission delay of offloading L_i bits of raw video in this model is

$$D_{i,t}^e = \frac{L_i}{r_i} \tag{5}$$

The delay of compressing L_i bits of raw video at the edge cloud is

$$D_{i,e}^c = \frac{L_i}{V_i^e} \tag{6}$$

The total delay of compressing L_i bits of raw video at the model of edge cloud is

$$D_{i,e} = D_{i,t}^e + D_{i,e}^c = \frac{L_i}{r_i} + \frac{L_i}{V_i^e} \tag{7}$$

For convenient analysis, the notations of this paper are summarized and shown in Table 1.

IV. PROBLEM FORMULATION AND SOLUTION METHOD FOR LOCAL COMPRESSION MODEL

A. PROBLEM FORMULATION

Our objective is to get the minimization latency of all multimedia applications in this compression model. The latency includes the local compression delay and the transmission delay. According to Eq.(4), we get the following latency minimization problem.

P1:

$$\begin{aligned} &\min_{B_i} \sum_{i=1}^{N} \alpha_i \left(\frac{L_i}{V_i^l} + \frac{\beta L_i}{r_i}\right) \\ &\sum_{i=1}^{N} B_i \le B \end{aligned} \tag{8}$$

where α_i denotes the fairness among mobile devices, which satisfies $\sum_{i=1}^{N} \alpha_i \le 1$, and the constraint is the constraint of a total communication resource.

B. SOLUTION METHOD

It is easily verified that the problem **P1** is convex. Its Lagrangian function is

$$L(B_i) = \sum_{i=1}^{N} \alpha_i \left(\frac{L_i}{V_i^l} + \frac{\beta L_i}{r_i}\right) + \mu \left(\sum_{i=1}^{N} B_i - B\right) \tag{9}$$

where $\mu \geq 0$ is the Lagrangian multiplier. By the KTT conditions [35], we get the below necessary and sufficient conditions

$$\frac{\partial L}{\partial B_i} = -\frac{\alpha_i \beta L_i}{B_i^2 \log_2\left(1+\frac{p_i h_i}{N_0}\right)} + \mu = 0 \tag{10}$$

$$\mu\left(\sum_{i=1}^{N} B_i - B\right) = 0 \tag{11}$$

$$\mu \geq 0 \tag{12}$$

From the Eq.(10), it is obvious that $\mu > 0$, hence, we have

$$B_i = \sqrt{\frac{\alpha_i \beta L_i}{\mu \log_2\left(1+\frac{p_i h_i}{N_0}\right)}} \tag{13}$$

and

$$\sum_{i=1}^{N}(B_i - B) = 0 \tag{14}$$

Substituting the Eq. (13) into the Eq.(14), we get

$$\mu = \left[\frac{\sum_{i=1}^{N}\sqrt{\frac{\alpha_i \beta L_i}{\log_2\left(1+\frac{p_i h_i}{N_0}\right)}}}{B}\right]^2 \tag{15}$$

Combining with Eq.(14), the optimal value of B_i is

$$B_i^* = \frac{\sqrt{\alpha_i \beta L_i}}{\sum_{i=1}^{N}\sqrt{\alpha_i \beta L_i}} B \tag{16}$$

V. PROBLEM FORMULATION AND SOLUTION METHOD FOR EDGE CLOUD COMPRESSION MODEL

In this section, the problem of the latency minimization for the edge cloud compression model is formulated, and the solution method is proposed.

A. FORMULATED PROBLEM

The delay minimization problem is denoted as:

P2:

$$\min_{B_i, V_i^e} \sum_{i=1}^{N} \alpha_i \left(\frac{L_i}{r_i} + \frac{L_i}{V_i^e}\right)$$
$$\sum_{i=1}^{N} B_i \leq B \tag{17}$$
$$\sum_{i=1}^{N} V_i^e \leq V$$

where the first and the second constraints constrain total communication and computation resources, respectively.

B. SOLUTION METHODS

It is easily verified that the problem **P2** is convex. Its Lagrangian function can be formulated as

$$L(B_i, V_i^e) = \sum_{i=1}^{N} \alpha_i \left(\frac{L_i}{r_i} + \frac{L_i}{V_i^e}\right) + u\left(\sum_{i=1}^{N} B_i - B\right) + v\left(\sum_{i=1}^{N} V_i^e - V\right) \tag{18}$$

where $u \geq 0$ and $v \geq 0$ are the Lagrangian multipliers.

According to the KTT conditions [35], we get the following necessary and sufficient conditions

$$\frac{\partial L}{\partial B_i} = -\frac{\alpha_i L_i}{B_i^2 \log_2\left(1+\frac{p_i h_i}{N_0}\right)} + u = 0 \tag{19}$$

$$\frac{\partial L}{\partial v_i^e} = -\frac{\alpha_i L_i}{v_i^{e(2)} \log_2\left(1+\frac{p_i h_i}{N_0}\right)} + v = 0 \tag{20}$$

$$u\left(\sum_{i=1}^{N} B_i - B\right) = 0 \tag{21}$$

$$v\left(\sum_{i=1}^{N} V_i^e - V\right) \tag{22}$$

$$u \geq 0 \tag{23}$$

$$v \geq 0 \tag{24}$$

Therefore, the optimal solution for the problem **P2** is

$$B_{i,e}^* = \frac{\sqrt{\alpha_i L_i}}{\sum_{i=1}^{N}\sqrt{\alpha_i L_i}} B \tag{25}$$

$$V_{i,e}^* = \frac{\sqrt{\alpha_i L_i}}{\sum_{i=1}^{N}\sqrt{\alpha_i L_i}} V \tag{26}$$

C. OPTIMAL RESOURCE ALLOCATION AND APPLICATION OFFLOADING ALGORITHM.

After getting the optimal solutions for each of the compression models, the resource allocation optimization and application offloading algorithm is proposed in Algorithm 1.

VI. EXPERIMENTAL RESULTS

In this section, experimental simulations are performed, and the simulation results are to validate the efficacy of the proposed offloading scheme. In a detailed manner, we analyze the offloading decisions and the system delay with respect to different parameters like the mobile device number, the compression capacity of mobile devices, and the compression capacity for the edge cloud.

We compare the following two benchmark compression models with our proposed compression method:

Algorithm 1: Optimal Resource Allocation and Multimedia Applications Offloading

Input:

The number of N multimedia applications;

Output:

the multimedia application offloading decision of each device and delay of all devices;

Calculate the optimal allocated bandwidth of each mobile device by solving problem **P1**;

Then the optimal value B_i is obtained, which is expressed in Eq.(16);

Get the optimal solution of total local delay of all mobile devices according to Eq.(4);

Calculate the computing resource allocation of each mobile device by solving problem **P2**;

Then the corresponding values are obtained and expressed in Eq.(25) and Eq.(26), respectively;

Get the optimal solution of total delay for the edge cloud compression model according to Eq.(7);

while $i \leq N$ **do**

  if $D_{i,l} \geq D_{i,e}$ **then**

    $a_i = 1$;

  else

    $a_i = 0$;

  end if

end while

Local Compression Model: In this compression model, mobile devices compress their multimedia applications locally.

Edge Cloud Compression Model: In this compression model, mobile devices offload their multimedia applications and compress these applications in the edge cloud.

Random Compression Model: In this compression model, some mobile devices offload their multimedia applications and are compressed by the edge cloud while the left ones compress their multimedia applications in their own devices.

A. PARAMETER SETTING

Consider a MEC system where we set the default parameters as follows by referring to [33], [36] unless otherwise stated. The weights of all mobile devices are set as $\alpha_i = \frac{1}{N}$, the transmission power is $p_i = 0.01$ W, the variance of the AWGN is $N_0 = 10^{-7}$, the bandwidth is $B = 10$ MHz, the video sizes and the compression capacity of the devices are uniformly distributed with $L_l = 10, L_m = 100]$ Mbits and $V_i^l \in [0.5, 2]$ Mbps, the capacity for the edge cloud V_i^e is 40 Mbps, and the value of compression ratio $\beta = 0.01$, which is a typical value when adopting the technique of MPEG4 video compression. The main simulation parameter values are summarized in Table 2 .

B. THE EFFECT OF THE MOBILE DEVICE NUMBER

Firstly, the effect of the mobile device number N on the computation offloading decisions and the system delay are

TABLE 2. Notations Summary

Notation	Description
i	$i \in \{1, 2, ..., N\}$, which is the device set
V_i^l	the compression capability of the mobile device i
L_i	the size of the raw video of device i in bits
C_i	the required amount of CPU cycles to compress the raw video of device i
B	the total bandwidth
B_i	the allocated channel bandwidth to the device i
a_i	the task offloading decision of device i
p_i	the transmission power of the mobile device i
h_i	the channel gain from the device i to the BS
N_0	the noise power
r_i	the achieved rate of the device i
V_i^e	the allocated computational resource of the edge cloud to the device i
$D_{i,f}$	the compression delay for processing L_i bits of raw video
$D_{i,t}^l$	the delay of transmitting the number of βL_i compressed video to the edge cloud
$D_{i,l}$	the total delay for mobile device i to accomplish its task in local compression
$D_{i,t}^e$	the transmission delay of offloading L_i bits of raw video to the edge cloud
$D_{i,e}^e$	the total delay of compressing L_i bits of raw video at the edge cloud
α_i	the fairness among devices

TABLE 3. Simulation Parameters

Parameters	Values
N	10
B	10MHz
α_i	$\frac{1}{N}$
L_i	$[L_l, L_m]$ Mbits
V_i^l	[0.5, 2] Mbps
p_i	0.01 W
h_i	1
N_0	10^{-7} W
V	40 Mbps
β	0.01

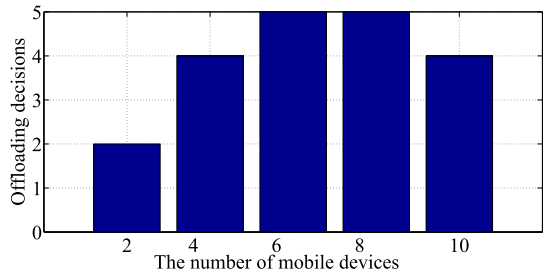


FIGURE 3. The offloading decisions with respect to the compression capacity of mobile devices.

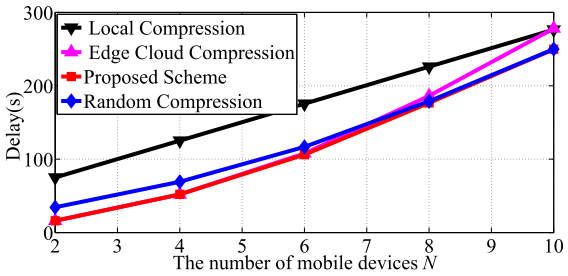


FIGURE 4. The system delay with respect to the mobile device number.

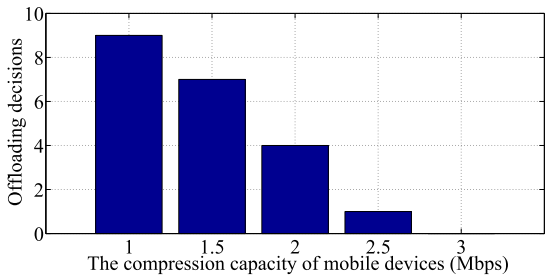


FIGURE 5. The offloading decisions with respect to the compression capacity of mobile devices.

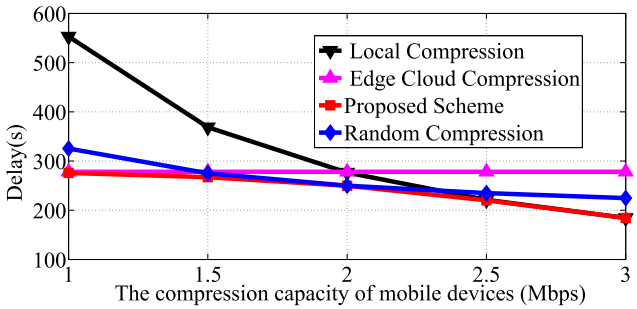


FIGURE 6. The system delay with respect to the compression capacity of mobile devices.

analyzed. Fig. 3 and Fig. 4 show the varying offloading decisions, and the system delay under different numbers of mobile devices. For a mobile device, the offloading decision means whether its application is to be processed in the edge cloud or not. Recall that if the multimedia application of this mobile device is processed in the edge cloud, then the offloading decision is 1. Otherwise, the offloading decision is 0. From Fig. 3, we see that mobile devices offload their multimedia applications to the edge cloud when the mobile device number is small. When the number for the mobile devices increases, some devices will choose the local compression model to compress their multimedia applications. This is because the communication and computing resources are limited, which leads to a higher system delay. From Fig. 4, we see that the system delays will increase if the number of mobile devices increases. By comparing with the three benchmark compression models, it is obvious that the proposed compression scheme can achieve better performance.

C. EFFECT OF THE COMPRESSION CAPACITY OF MOBILE DEVICES

We next analyze the effect of the compression capacity of mobile devices on the decisions of offloading and the system delay, the results of which are shown in Figs. 5 and 6, respectively. The number for mobile devices N is set as 10. From Fig. 5, it is evident that the number for mobile devices that choose the offloading decisions is decreasing while the compression capacity of mobile device increasing. That is due to the reason that more mobile devices will compress their applications locally as the compression capacity increases. Especially, all mobile devices will offload their applications when the compression capacity achieves 3 Mbps. From Fig. 6, it can be found that the system delays caused by the local compression and random models decrease evidently while the compression capacity of mobile devices increasing. The system delay caused by the local compression scheme is higher compared with the edge cloud scheme before the mobile device's compression capacity achieves 2 Mbps. However, the system delay caused by the local compression scheme becomes lower than that of the edge cloud scheme with the mobile device's compression capacity increasing. In addition, the proposed scheme achieves the lowest system delay compared with the three benchmark methods.

D. THE EFFECT OF THE EDGE CLOUD COMPRESSION CAPACITY

In this part, the effect of the compression capacity of the edge cloud V on the applications offloading decisions for mobile devices and the system delay is analyzed, the results of which are shown in Figs. 7 and 8, respectively. The mobile device number is fixed as 10, the capacity of mobile devices as 2 Mbps, and vary the edge cloud compression capacity. From Fig. 7, we can observe that more mobile devices will offload their multimedia applications if the compression capacity for the edge cloud increases. We can observe from Fig. 8 that the system delays under the edge cloud compression scheme and our proposed scheme decrease greatly versus the increase of the compression capacity for the edge cloud. This is due to the reason that more users of mobile device adopts the offloading method increase.

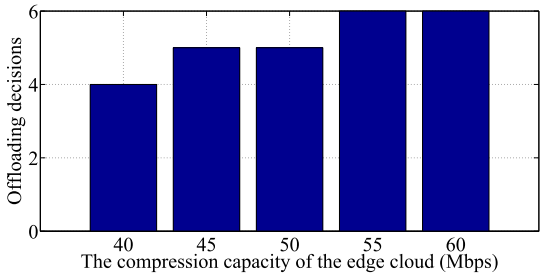


FIGURE 7. The offloading decisions made by mobile devices with respect to the compression capacity of the edge cloud.

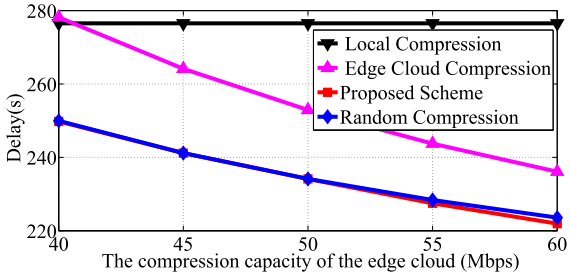


FIGURE 8. The system latency with respect to the compression capacity of the edge cloud.

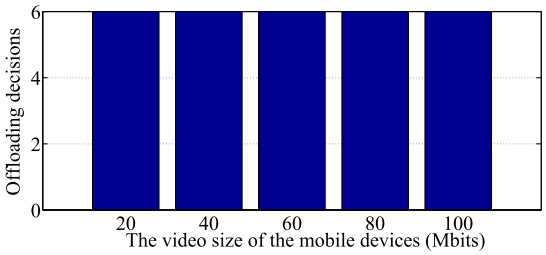


FIGURE 9. The effect of the video size on the offloading decisions of mobile devices.

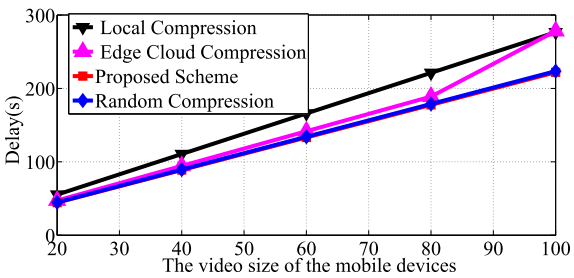


FIGURE 10. The effect of the video size on the system delay.

E. THE EFFECT OF THE VIDEO SIZE

The effect of the video size in this part is analyzed. We set the mobile device number as 10, the capacity of the edge cloud as 40 Mbps, the compression capacity of each mobile device is 2 Mbps, and we vary the video size L_m from 20 Mbits to 100 Mbits. Figs. 9 and 10 illustrate how the video size affects the offloading decisions made by mobile devices and the system delay, respectively. We observe from Fig. 9 that the mobile device number adopting offloading decisions does not change even if the video sizes increases. The offloading decisions made by mobile devices are determined by the delay from the local compression and that from the edge cloud compression. When the capacities of mobile devices and the edge cloud are fixed, with the video sizes of mobile devices increasing, the delay of local compression and that of the edge cloud compression also increase accordingly. Observing from Fig. 9, we find that only 6 mobile devices decide to offloading their applications. From Fig. 10, it is evident that the system delay increases with the video size increasing. This is intuitive as when the computing resources for mobile devices and edge cloud are fixed; larger video sizes will cause higher system delay. By comparing with the three baseline methods, the proposed compression scheme can achieve the lowest delay.

VII. CONCLUSIONS AND FUTURE WORKS

In this paper, we have studied the multimedia applications offloading problem in a MEC system minimizing the system delay by allocating the communication resources and computing resource in the edger servers. Two problems under the models of local compression and the edge cloud compression are formulated, and a solution method is proposed for each of the models, based on which an efficient multimedia application offloading scheme is proposed. Simulation results are conducted to verify the performance for our proposed offloading scheme. The results show that the scheme proposed in this paper can outperform the two benchmark methods under different parameters.

Some research problems will be left as future works. First, as NOMA is viewed as a promising technology in the future wireless networks, the NOMA technology can be adopted for the channel access; Second, we can consider both the partial task offloading and binary task offloading jointly; Third, we can study the scenario of task offloading by integrating MEC with ultra-dense network, which are studied separately in the previous works. We can also apply machine learning and other optimization methods, such as the dandelion algorithm (DA) [42] and deep reinforcement learning, to solve the task offloading optimization problems.

REFERENCES

[1] E. Dave, "The Internet of Things: How the next evolution of the internet is changing everything," CISCO White Paper. [Online]. Available: https://www.cisco.com/c/dam/en_us/about/ac79/docs/innov/IoT_IBSG_0411FINAL.pdf

[2] C. Yi, J. Cai, and Z. Su, "A multi-user mobile computation offloading and transmission scheduling mechanism for delay-sensitive applications," *IEEE Trans. Mobile Comput.*, vol. 19, no. 1, pp. 29–43, Jan. 2020.

[3] X. Li *et al.*, "Performance and power consumption tradeoff in multimedia cloud," *Multimedia Tools Appl.*, vol. 79, pp. 33381–33396, 2020.

[4] Z. Liu, C. Zhan, Y. Cui, C. wu, and H. Hu, "Robust edge computing in UAV systems via scalable computation and cooperative computation," *IEEE Wireless Commun.*, to be published, doi: 10.1109/MWC.121.2100041.

[5] S. Ranadheera, S. Maghsudi, and E. Hossain, "Computation offloading and activation of mobile edge computing servers: A minority game," *IEEE Wireless Commun. Lett.*, vol. 7, no. 5, pp. 688–691, Oct. 2018.

[6] Cisco, "Cisco visual networking index: Global mobile data traffic forecast update," 2017–2022, Feb. 2019.

[7] N. Lin *et al.*, "A novel improved bat algorithm in UAV path planning," *J. Comput., Mater. Contin*, vol. 61, no. 1, pp. 323–344, 2019.

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

[9] Z. Kuang *et al.*, "Cooperative computation offloading and resource allocation for delay minimization in mobile edge computing," *J. Syst. Architecture*, vol. 118, pp. 1–9, 2021.

[10] X. Li *et al.*, "A cooperative resource allocation model for IoT applications in mobile edge computing," *Comput. Commun.*, vol. 173, pp. 183–191, 2021.

[11] C. Zhan, H. Hu, Z. Liu, Z. Wang, and S. Mao, "Multi-UAV-enabled mobile edge computing for time-constrained IoT applications," *IEEE Internet Things*, vol. 8, no. 20, pp. 15553–15567, Oct. 2021.

[12] X. Chen, C. Wu, Z. Liu, N. Zhang, and Y. Ji, "Computation offloading in beyond 5G networks: A distributed learning framework and applications," *IEEE Wireless Commun.*, vol. 28, no. 2, pp. 56–62, Apr. 2021.

[13] F. Wang, J. Xu, and Z. Ding, "Multiantenna NOMA for computation offloading in multiuser mobile edge computing systems," *IEEE Trans. Commun.*, vol. 67, no. 3, pp. 2450–2463, Mar. 2019.

[14] O. Munoz, A. Pascual-Iserte, and J. Vidal, "Optimization of radio and computational resources for energy efficiency in latency constrained application offloading," *IEEE Trans. Veh. Technol.*, vol. 64, no. 10, pp. 4738–4755, Oct. 2015.

[15] W. Sun, J. Liu, Y. Yue, and H. Zhang, "Double auction-based resource allocation for mobile edge computing in industrial Internet of Things," *IEEE Trans. Ind. Inform.*, vol. 14, no. 10, pp. 4692–4701, Oct. 2018.

[16] B. Yang, X. Cao, X. Li, Q. Zhang, and L. Qian, "Mobile edge computing based hierarchical machine learning tasks distribution for IIoT," *IEEE Internet Things J.*, vol. 7, no. 3, pp. 2169–2180, Mar. 2020.

[17] Y. Jie, C. Guo, K. R. Choo, C. Z. Liu, and M. Li , "Game theoretic resource allocation for fog-based industrial Internet of Things environment," *IEEE Internet Things J.*, vol. 7, no. 4, pp. 3041–3052, Apr. 2020.

[18] P. Zhao, H. Tian, C. Qin, and G. Nie, "Energy saving offloading by jointly allocating radio and computational resources for mobile edge computing," *IEEE Access*, vol. 5, pp. 11255–11268, Jul. 2017, doi: 10.1109/ACCESS.2017.2710056.

[19] M. T. Kabir and C. Masouros, "A scalable energy vs. latency trade-off in full-duplex mobile edge computing systems," *IEEE Trans. Commun.*, vol. 67, no. 8, pp. 5848–5861, Aug. 2019.

[20] T. Q. Dinh, J. Tang, Q. D. La, and T. Q. S. Quek, "Offloading in mobile edge computing: Task allocation and computational frequency scaling," *IEEE Trans. Commun.*, vol. 65, no. 8, pp. 3571–3584, Aug. 2017.

[21] Y. Gao, Y. Cui, X. Wang, and Z. Liu, "Optimal resource allocation for scalable mobile edge computing," *IEEE Commun. Lett.*, vol. 23, no. 7, pp. 1211–1214, Jul. 2019.

[22] H. Chen *et al.*, "Knowledge distillation for mobile edge computation offloading," *ZTE Commun.*, vol. 18, no. 2, pp. 40–48, Jun. 2020.

[23] Z. Yang, Y. Liu, Y. Chen, and N. Al-Dhahir, "Cache aided NOMA mobile edge computing: A reinforcement learning approach," *IEEE Trans. Wireless Commun.*, vol. 19, no. 10, pp. 6899–6915, Oct. 2020.

[24] J. Yan, S. Bi, and Y. J. Zhang, "Offloading and resource allocation with general task graph in mobile edge computing: A deep reinforcement learning approach," *IEEE Trans. Wireless Commun.*, vol. 19, no. 8, pp. 5404–5419, Aug. 2020.

[25] H. Li, C. Zhou, and X. Lu, "Optimization of radio and computational resources for energy efficiency in latency constraine application offloading," *IEEE Trans. Veh. Technol.*, vol. 64, no. 10, pp. 4738–4755, Oct. 2015.

[26] L. Zhang and N. Ansari, "Latency-aware IoT service provisioning in UAV-aided mobile edge computing networks," *IEEE Internet Things J.*, vol. 7, no. 10, pp. 10573–10580, Oct. 2020.

[27] Z. Kuang, L. Li, J. Gao, L. Zhao, and A. Liu, "Partial offloading scheduling and power allocation for mobile edge computing systems," *IEEE Internet Things J.*, vol. 6, no. 4, pp. 6774–6785, Aug. 2019.

[28] X. Chen, H. Zhang, C. Wu, S. Mao, Y. Ji, and M. Bennis, "Optimized computation offloading performance in virtual edge computing systems via deep reinforcement learning," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4005–4018, Jun. 2019.

[29] Y. Wu *et al.*, "Secrecy based delay-aware computation offloading via mobile edge computing for Internet of Things," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4201–4213, Jun. 2019.

[30] Y. Huang, Y. Liu, and F. Chen, "NOMA-aided mobile edge computing via user cooperation," *IEEE Trans. Commun.*, vol. 68, no. 4, pp. 2221–2235, Apr. 2020.

[31] Z. Yang, C. Pan, J. Hou, and M. Shikh-Bahaei, "Efficient resource allocation for mobile edge computing networks with NOMA: Completion time and energy minimization," *IEEE Trans. Commun.*, vol. 67, no. 11, pp. 7771–7784, Nov. 2019.

[32] Y. Yu, X. Bu, K. Yang, Z. Wu, and Z. Han, "Green large-scale fog computing resource allocation using joint benders decomposition, Dinkelbach algorithm, ADMM, and Branch-and-Bound," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4106–4117, Jun.. 2019.

[33] J. Ren, G. Yu, Y. Cai, and Y. He, "Latency optimization for resource allocation in mobile edge computation offloading," *IEEE Trans. Wireless Commun.*, vol. 17, no. 8, pp. 5506–5519, Aug. 2018.

[34] W. Wen, Y. Cui, T. Q. S. Quek, F. Zheng, and S. Jin, "Joint optimal software caching, computation offloading and communications resource allocation for mobile edge computing," *IEEE Trans. Veh. Technol.*, vol. 69, no. 7, pp. 7879–7894, Jul. 2020.

[35] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.

[36] W. Fang *et al.*, "A distributed ADMM approach for energy-efficient resource allocation in mobile edge computing," *Turkish J. Elect. Eng. Comput. Sci.*, vol. 26, no. 6, pp. 3335–3344, 2018.

[37] X. Chen *et al.*, "Multi-tenant cross-slice resource orchestration: A deep reinforcement learning approach," *IEEE J. Sel. Area Commun.*, vol. 37, no. 10, pp. 2377–2392, Oct. 2019.

[38] K. Cheng, Y. Teng, W. Sun, A. Liu, and X. Wang, "Energy-efficient joint offloading and wireless resource allocation strategy in multi-MEC server systems," in *Proc. IEEE Int. Conf. Commun.*, 2018, pp. 1–6.

[39] X. Lyu, H. Tian, W. Ni, Y. Zhang, P. Zhang, and R. P. Liu, "Energy-efficient admission of delay-sensitive tasks for mobile edge computing," *IEEE Trans. Commun.*, vol. 66, no. 6, pp. 2603–2616, Jun. 2018.

[40] S. Misra and S. Bera, "Soft-VAN: Mobility aware task offloading in software defined vehicular network," *IEEE Trans. Veh. Technol.*, vol. 69, no. 2, pp. 2071–2078, Feb. 2020.

[41] Z. Zhou, P. Liu, J. Feng, Y. Zhang, S. Mumtaz, and J. Rodriguez, "Computation resource allocation and task assignment optimization in vehicular fog computing: A contract-matching approach," *IEEE Trans. Veh. Technol.*, vol. 68. no. 4, pp. 3113–3125, Apr. 2019.

[42] X. Li *et al.*, "New dandelion algorithm optimizes extreme learning machine for biomedical classification problems," *Comput. Intell. Neurosci.*, vol. 2017, pp. 1–14, 2017.

[43] Y. Wu *et al.*, "Efficient task scheduling for servers with dynamic states in vehicular edge computing," *Comput. Commun.*, vol. 150, pp. 245–253, 2020.

[44] X. Chen, Y. Cai, L. Li, M. Zhao, B. Champagne, and L. Hanzo, "Energy-efficient resource allocation for latency sensitive mobile edge computing," *IEEE Trans. Veh. Technol.*, vol. 69, no. 2, pp. 2246–2262, Feb. 2020.

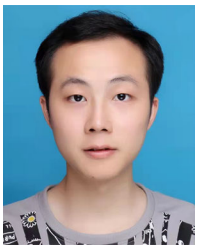
[45] C. You, K. Huang, H. Chae, and B. Kim, "Energy-efficient resource allocation for mobile edge computation offloading," *IEEE Wireless Commun.*, vol. 16, no. 3, pp. 1397–1411, Mar. 2017.

GUOLONG CHEN received the master's and Doctoral degrees in 1993 and 1998, respectively. He is currently a Professor with the School of Computer and Information Engineering, Bengbu University, Bengbu, China. He is also a Professor with the School of Information Engineering, Suzhou University, Suzhou, China. His research interests include Internet of Things and Big Data.

LIANG ZHAO (Member, IEEE) received the Ph.D. degree from the School of Computing, Edinburgh Napier University, Edinburgh, U.K., in 2011. He is currently an Associate Professor with Shenyang Aerospace University, Shenyang, China. Before joining Shenyang Aerospace University, he was an Associate Senior Researcher with Hitachi, China, Research and Development Corporation from 2012 to 2014. He has authored or coauthored more than 70 articles. His research interests include VANETs, SDVN, FANETs, and WMNs.

XIANWEI LI received the master's degree from Hunan University, Changsha, China, in 2010 and the Doctoral degree from Waseda University, Tokyo, Japan, in 2019. He is currently a Lecturer with the School of Computer and Information Engineering, Bengbu University, Bengbu, China. His research interests include Internet of Things, machine learning, and Big Data.

FUQI ZHAO received the M.S. degree from Northeastern University, Shengyang, China, in 2014. He is currently a Lecturer with the School of Computer and Information Engineering, Bengbu University, Bengbu, China. His research interests include computer vision and image recognition.



XIAOJIAN ZENG received the M.S. degree in computer technology from Yangtze University, Jingzhou, China, in 2021. He used to serve Huawei Cloud, as an AI-enabling Engineer, and is currently an AI Algorithm Engineer with Sky Limit Entertainment Company. He has translated and authored or coauthored several books on AI-related topics. His main research interests include recommender systems, autonomous driving, and other AI-related topics.