Online preschool education optimization based on edge computing in the era of COVID-19

Xiaoxuan Duan

Beihua University Teacher's College, Jilin, China

Correspondence
Xiaoxuan Duan, Beihua University Teacher's College, Address: No. 22, Jilin Street, Fengman District, China.
Email: dxx1011@163.com

Under the influence of COVID-19, people's normal life and activities have been limited, such as the education of children, which leads to the emergence of online preschool education. Since online preschool education is large-scale and time-sensitive, the traditional network model cannot satisfy the needs of online education. In this paper, edge computing is adopted to optimize online preschool education, where a task unloading algorithm based on genetic algorithm (TUOGA) is designed to minimize the computing delay of terminal tasks. In order to verify the effectiveness and efficiency of the proposed algorithm, TUOGA is compared with two task offloading algorithms, and simulation results show that the proposed algorithm outperforms them in the aspect of time latency.

KEYWORDS
computing offloading, edge computing, genetic algorithm, online preschool education

1 | INTRODUCTION

Affected by COVID-19, many activities have been limited, including children's education. To provide educational resources to children, the online educational model has emerged to teach students through the network. However, since a large number of requests occur at the same time, if these requests are transmitted to the central cloud server, it will cause huge time latency related to computing and transmission, affecting the normal progress of online education. Aiming at efficient task computing in online education, the appropriate scheduling scheme for learning resources is extremely significant.

Recently, edge computing as an emerged technology has been investigated for resource scheduling, which can provide data computing services to users by using the computing resources stored in the edge cloud. In the edge computing system, the edge server is considered as a medium to handle tasks by using edge cloud computing resources closer to the local, which avoids transferring data to the remote central cloud. Computing offloading is an important strategy of edge computing to schedule tasks, which allocates the task to the edge cloud to store and process. Edge computing allocates complex computing tasks to local and edge servers by computing offloading algorithm, which reduces time latency of data processing.

This paper applies edge computing for online preschool education and designs a task offloading optimization algorithm based on genetic algorithm, namely TUOGA, to minimize the time latency of handling tasks. The contributions are shown as follows. (a) The online preschool education system based on edge computing is modeled. (b) A genetic algorithm is devised to solve the optimization model and obtain the optimal offloading computing scheme for online preschool education system.
SYSTEM MODEL AND PROBLEM FORMULATION

2.1 | System model

The architecture of online preschool education system based on edge computing is shown in Figure 1, which is composed of an edge server and a base station. In order to prevent the interference between terminal devices, the spectrum resources of base station are allocated to each terminal in an orthogonal way.9

In the system architecture, \( S \) represents the edge server to provide computing and communication services for \( N \) terminal devices, and the set of terminal devices is represented as \( D = \{d_1, d_2, \ldots, d_N\} \). Assume that each terminal device \( d_i \) only handles one task \( w_i \), and the triple is used to represent \( w_i \) as \( w_i = \{m_i, n_i, c_i\} \), where \( m_i \) represents the size of input task data, \( n_i \) represents the size of computing result data, and \( c_i \) represents the computing load of task. The total bandwidth allocated to each terminal device is set as \( B \). To show the scheduling scheme of tasks, a set is used to represent task computing offloading decisions, expressed as \( A = \{a_1, a_2, \ldots, a_N\} \), where \( a_i = \{0, 1\} \). If \( a_i = 1 \), \( d_i \) offloads \( w_i \) to edge server to handle; otherwise, \( d_i \) processes \( w_i \) locally. The objective is to obtain the set of task computing offloading decisions for online preschool education system to minimize the overall time latency of task computing and transmission.

2.2 | Communication model

The terminal device employs both uplink and downlink spectrum resources when processing tasks, and the uplink transmission rate of \( d_i \) is expressed as:

\[
r_{iu} = a_i B \log \left( 1 + \frac{|h_{i,B}|^2 P_{B} d^{-r}}{\sigma^2} \right)
\]  

(1)

where \( a_i \) is the ratio of uplink bandwidth occupied by terminal uploading tasks, \( h_{i,B} \) is the channel recession coefficient between base station and terminal, \( P_{B} \) is the transmission power of terminal, \( d \) is the distance between terminal and base station, \( r \) is the channel transmission loss, and \( \sigma^2 \) is the channel noise power.

In the same way, the downlink transmission rate of \( d_i \) is expressed as:

\[
r_{id} = \beta_i B \log \left( 1 + \frac{|h_{B,i}|^2 P_{B} d^{-r}}{\sigma^2} \right)
\]  

(2)

where \( \beta_i \) is the ratio of downlink bandwidth occupied by terminal receiving tasks, \( h_{B,i} \) is the channel recession coefficient between base station and terminal, and \( P_{B} \) is the transmission power of base station.
2.3 Computing model

In order to optimize online preschool education system based on edge computing, the objective is to obtain task computing offloading scheme with the lowest time latency, which includes two parts: computing time latency on the local and the edge server.

If task $w_i$ is not offloaded to edge server, it is computed on the terminal locally. The time latency of handling tasks locally is expressed as:

$$t_l^{i} = \frac{c_i}{f_l^i}$$

(3)

where $f_l^i$ represents the computing power of terminal $d_i$ to handle tasks locally. Therefore, the total time latency of terminal $d_i$ handling tasks locally is expressed as:

$$T_l^i = \sum_{l \in D} (1 - a_i) t_l^{i}$$

(4)

If task $w_i$ is offloaded to edge server, it is computed on the edge server. The time latency of handling tasks on edge server includes uplink transmission time, downlink transmission time, server computing time, and backhaul link time. Because base station and server are connected by wires, the backhaul link time is ignored. The uplink transmission delay is related to the size of uploading data and uplink transmission rate, which is expressed as:

$$t_u^{i} = \frac{m_i}{r_u^i}$$

(5)

The downlink transmission delay is related to the size of receiving data and downlink transmission rate, which is expressed as:

$$t_d^{i} = \frac{n_i}{r_d^i}$$

(6)

The computing time of server is related to the size of task load and the computing power of server, which is expressed as:

$$t_c^{i} = \frac{c_i}{f_i}$$

(7)

Therefore, the time delay of offloading task $w_i$ to edge server is expressed as:

$$t_s^{i} = t_u^{i} + t_d^{i} + t_c^{i}$$

(8)

To sum up, the overall time delay of offloading $w_i$ to edge server to handle is expressed as follows:

$$T_s^{i} = a_i t_s^{i}$$

(9)

2.4 Problem formulation

In this paper, the computing offloading scheme is required for learning resources scheduling to minimize time latency of handling tasks in online preschool education system based on edge computing. The system problem is formulated as an optimization problem, considering time latency as the metric. The optimization model is expressed as:

$$\begin{align*}
\min T &= \sum_{i=1}^{N} (T_c^{i} + T_s^{i}) \\
\text{s.t.} & \quad \sum_{d_i \in D} f_i \leq f_m \\
& \quad \sum_{d_i \in D} a_i \leq 1 \\
& \quad \sum_{d_i \in D} \beta_i \leq 1 \\
& \quad f_l^i \geq 0, \forall i \in D
\end{align*}$$

(10)
where $C_1$ represents that the total amount of tasks offloaded to the server cannot exceed the maximum computing power of server; $C_2$ and $C_3$ represent the ratio of uplink and downlink spectrum resources allocated to all terminals respectively; $C_4$ represents that the local computing power of terminal cannot be less than 0.

3 | GENETIC ALGORITHM FOR TASK COMPUTING OFFLOADING OPTIMIZATION

In order to solve the optimization problem related to task computing offloading, this section designs a genetic algorithm to obtain the optimal task computing offloading scheme for online preschool education system.

Firstly, the chromosome is employed to represent each individual, and each chromosome is coded by binary coding, which is expressed as $X = \{x_1, x_2, \ldots, x_N\}$, where $x_i$ corresponds to $a_i$ and represents the task computing offloading decision of $w_i$.

Then, the fitness value of each individual is calculated, which is used as the measure to evaluate the individual performance. The objective is to minimize the time latency of overall system, so fitness is defined related to time latency, and the lower time latency corresponds to the higher fitness. The fitness value is calculated as follows:

$$F_i = \frac{1}{T_i}$$ (11)

Next, the algorithm selects individual by a certain strategy for the next generation evolution process. The roulette method is used to select individuals, and the probability of the individual being selected is proportional to its fitness value, which is expressed as:

$$p_i = \frac{F_i}{\sum_{i \in D} F_i}$$ (12)

To generate offspring, the selected individuals are employed as parents to cross, which mates and recombines some genes of the chromosomes of the selected parent individuals in a certain way. The single crossover method is used for crossover process. The crossover point is set randomly in the coded chromosomes of two parents, and then the genes behind the crossover point of the two parents are exchanged, and two new individuals are generated.

The offspring generated by crossover will mutate with a certain probability to generate new individuals. Because the individual is coded by binary coding method, the binary mutation method is employed for mutating of the individual, which is described as: firstly, select a mutation position randomly, and then change the gene value on the mutation position from “0” to “1”, or from “1” to “0”, to generate a new individual.

Algorithm 1. Generic algorithm for task offloading

| Input: $I_{\text{max}}, f_m$;                                      |
| Output: $X_{\text{best}} = \{x_1, x_2, x_3, \ldots, x_n\}$ |
| For $i=1$: $I_{\text{max}}$                                    |
| Initialize the population, $X_{\text{best}}$, and $F_{\text{best}}$ |
| While $f < f_m$                                                |
| Calculate fitness value for each individual                   |
| Calculate selection probability of each individual            |
| Select parents by roulette                                    |
| Cross and mutate to generate offspring                       |
| End while                                                     |
| Obtain optimal task offloading solution $X_i$ and optimal fitness value $F_i$ |
| If $F_i > F_{\text{best}}$                                    |
| $X_{\text{best}} = X_i$                                       |
| End if                                                        |

End for

$X_{\text{best}}$ is the task offloading scheme for online childhood education system based on edge computing.
SIMULATION AND ANALYSIS

4.1 Setup

The simulations of genetic algorithm for online preschool education based on edge computing include two parts: the first one is to model online preschool education system based on edge computing and construct task computing offloading optimization model related to time latency; the other is conducting genetic algorithm for optimization model to obtain the optimal task computing offloading scheme. To ensure the reality of simulations, the real learning resources are used to deploy the system, and simulations are implemented on MATLAB with a computer configured with Intel(R) Core (TM)i5-9400F, CPU 2.90 GHz, 8GB RAM.

To testify the effectiveness and efficiency of the proposed algorithm, it is compared with local offloading algorithm (LOA) and random offloading algorithm (ROA) in terms of time latency. The parameter settings involved in simulations are shown as follows. \( B = 40 \text{ MHz}, P_B = 50 \text{ dBm}, P_{L,B} = 50 \text{ dBm}, r = 4, \sigma^2 = -185 \text{ dBm} \) and \( f_m = 40\,000 \text{ Megacycles/s} \). Furthermore, the number of simulations is 10.

4.2 Results analysis

Comparison simulations are implemented under different numbers of terminals, in terms of time latency, and the simulation results are shown in Figure 2. Figure 2 shows that TUOGA performs better than LOA and ROA. With the increasing of terminals, TUOGA can appropriately offload tasks from terminals to edge server to optimize the process of handling tasks.

Comparison simulations are implemented under different sizes of tasks, in terms of time latency, and the simulation results are shown in Figure 3. Figure 3 shows that TUOGA performs better than LOA and ROA. With the increasing of task size, TUOGA can better allocate the optimal learning resources for different terminals to minimize the time latency of task handling.

5 CONCLUSION

Under the influence of COVID-19, children’s education cannot be carried out normally, which leads to the emergence of online preschool education mode. In this paper, edge computing is used for online preschool education, and the algorithm called TUOGA is designed to optimize the computing latency for the system. Firstly, the online preschool education system based on edge computing is modeled, and the optimization model is constructed in terms of computing latency.
Next, TUOGA is proposed to solve the optimization model. Finally, TUOGA is compared with LOA and ROA, and simulation results show that TUOGA is efficient. However, in the actual online education system, tasks on the terminal can be multiple, which puts forward more challenges for task unloading. Therefore, the problem of multiple tasks will be considered. In addition, during the period of COVID-19, the further optimization on online preschool education will also be continued.

DATA AVAILABILITY STATEMENT
The data that support the findings of this study are available from the corresponding author upon reasonable request.

ORCID
Xiaoxuan Duan https://orcid.org/0000-0002-7606-1040

REFERENCES
1. Alsamhi SH, Lee B, Guizani M, Kumar N, Qiao Y, Liu X. Blockchain for decentralized multi-drone to combat COVID-19 and future pandemics: framework and proposed solutions. *Trans Emerg Telecommun Technol*. 2021. https://doi.org/10.1002/ett.4255.
2. Khan WZ, Ahmed E, Hakak S, Yaqoob I, Ahmed A. Edge computing: a survey. *Futur Gener Comput Syst*. 2019;97:219-235.
3. Ren J, Pan Y, Gosinski A, Beyah RA. Edge computing for the internet of things. *IEEE Netw*. 2018;32(1):6-7.
4. Lin H, Zeadally S, Chen Z, Labiod H, Wang L. A survey on computation offloading modeling for edge computing. *J Netw Comput Appl*. 2020;169:1-25.
5. Farahbakhsh F, Shahidinejad A, Ghobaei-Arani M. Multiuser context-aware computation offloading in mobile edge computing based on Bayesian learning automata. *Trans Emerg Telecommun Technol*. 2021;32(1):1-26.
6. Lin L, Liao X, Jin H, Li P. Computation offloading toward edge computing. *Proc IEEE*. 2019;107(8):1584-1607.
7. Li X, Dang Y, Aazam M, Peng X, Chen T, Chen C. Energy-efficient computation offloading in vehicular edge cloud computing. *IEEE Access*. 2020;8:37632-37644.
8. Jang J, Jung J, Hong J. K-LZF: an efficient and fair scheduling for edge computing servers. *Futur Gener Comput Syst*. 2019;98:44-53.
9. Luo J, Deng X, Zhang H, Qi H. QoE-driven computation offloading for edge computing. *J Syst Archit*. 2019;97:34-39.
10. Wang Z, Sobey A. A comparative review between genetic algorithm use in composite optimisation and the state-of-the-art in evolutionary computation. *Compos Struct*. 2020;233:1-20.
11. Zhang Y, Niyato D, Wang P. Offloading in mobile cloud-let systems with intermittent connectivity. *IEEE Trans Mob Comput*. 2015;14(12):2516-2529.

How to cite this article: Duan X. Online preschool education optimization based on edge computing in the era of COVID-19. *Internet Technology Letters*. 2021;e292. https://doi.org/10.1002/itl2.292