Task Prediction Based Computation Offloading over Multi-UAV MEC Network

Xi Cheng[1], Zhenquan Qin[1], Ruixin Liu[1], Jiong Lu[2,3], and Jianbo Zheng[3,4](✉)

[1] Dalian University of Technology, Dalian 116024, Liaoning, China
[2] City University of Macau, Macau 999078, China
[3] Guangdong-Hong Kong-Macao Joint Laboratory for Emotional Intelligence and Pervasive Computing, Shenzhen MSU-BIT University, Shenzhen 518172, Guangdong, China
[4] Artificial Intelligence Research Institute, Shenzhen MSU-BIT University, Shenzhen 518172, Guangdong, China
`jianbo.zheng@smbu.edu.cn`

Abstract. In mobile edge computing (MEC), unmanned aerial vehicle (UAV) acts as a base station that can quickly provide communication and computing services for areas with a limited communication infrastructure. However, most of offloading studies neglect the deep learning algorithm to understand the dynamic changes of task in different time slots. Moreover, in large-scale computing offloading, the UAVs deployment only considers position optimization, not number optimization. Considering the limited energy of terminal device (TD) and UAV, this paper proposes a joint Task Prediction (TP) and Differential Evolution (DE) optimization framework, called TpDeRas, to reduce the total energy consumption over multi-UAV MEC system. To predict the future task set, we first use a TP algorithm based on distributed long short-term memory (LSTM) to achieve task prediction for different TD. Based on TP results, the optimization problem is divided into UAVs deployment subproblem and resource allocation subproblem. UAVs deployment optimization needs to consider number and position. We propose an adaptive DE algorithm to optimize the UAVs deployment. Based on TP results and UAVs deployment, we uses efficient greedy algorithm to optimize resource allocation. Experimental results show that TpDeRas approach greatly improves performance compared to other algorithms.

Keywords: Multi-UAV assisted MEC · Task prediction · UAVs deployment · Resource allocation

1 Introduction

With the development of mobile communication technology and the popularity of intelligent terminals, mobile users have higher requirements for network

---

This work was supported in part by the Shenzhen Sustainable Development Special Project under grant KCXFZ20201221173411032.

V. C. M. Leung et al. (Eds.): Qshine 2023, LNICST 574, pp. 438–452, 2024.
https://doi.org/10.1007/978-3-031-65123-6_33

performance, including network quality of service (QoS), delay, bandwidth and more. However, the rapid development of Internet technology produce the explosive growth of the Internet of things (IoT) devices, which generate lots of data and need more computing resources. The current mainstream computing mode is cloud computing architecture. Using the cloud computing mode, the massive data generated by terminal devices will create considerable pressure on both transmission networks and cloud computing centers. To solve these problems, the concept of mobile edge computing (MEC) [1] has been proposed. It moves the computing power of the terminal up, sinks the computing power of the cloud to the network edge, and constructs a 'cloud-edge-end' tripartite collaborative architecture. Because the computing and storage resources of edge base station are not as sufficient as the cloud computing architecture, it is necessary to use more effective algorithms to reasonably schedule the task offloading decision and node resource allocation in the process of computing offloading, and thus improve the users QoS and the utilization of computing resources [2,3]. Most of them are based on fixed edge computing nodes to study the resource allocation, so as to improve the performance of edge computing networks. For remote areas or emergency situations, the fixed edge nodes and their limited computing resources cannot meet the terminal equipment QoS, so it is necessary to consider using appropriate mobile relay equipment as MEC node. In recent years, due to low cost, high flexibility and universality, Unmanned Aerial Vehicle (UAV) has been widely concerned with wireless communication [4,5]. As in traffic monitoring, intelligent agriculture, emergency rescue and battlefield communication, UAV can be used as a mobile base station to provide communication and computing services for network edge both quickly and flexibly. Considering the mobility of UAV and limitation of battery, it is necessary to reasonably control trajectory of UAV and resource allocation to ensure that UAV cluster can serve as many IoT devices as possible.

At present, the research only has considered the single UAV-enabled MEC [6–8], and reduces the system cost by jointly optimizing the UAV position and resource allocation. For computing intensive and delay sensitive tasks, the coverage and resources provided by single UAV are limited, so multi-UAV are usually needed to assist MEC [9,10]. For multi-UAV assisted MEC, it is necessary to study the deployment of UAVs in different time slots, so as to improve the efficiency of computing offloading. Yang et al. [9] used one-dimensional search algorithm to find the optimal position of UAV, which is too complex, and only optimized the position of UAV cluster. At present, most of the research on multi-UAV assisted MEC does not consider that the number of UAVs required will change with the task chang of terminal device in different time slots.

Inspired by the above, taking intelligent traffic monitoring system as an example, this paper proposes a multi-UAV assisted MEC scenario that can serve large scale terminal devices. Due to the limited energy consumption of terminal device and UAV, the UAVs deployment and resource allocation are jointly optimized with the goal of minimizing the total energy consumption required to complete all tasks. The task prediction (TP) model based on long short-term memory

(LSTM) neural network can realize the “no perception" delay. The adaptive differential evolution algorithm can optimize the number and position of UAVs and greatly reduce the system energy consumption. The main contributions of this paper are as follows.

1. A multi-UAV assisted MEC system is constructed, in which UAV is used as relay and mobile edge base station to provide communication and computation services for large scale terminal devices. We propose an optimization method called TpDeRas, which uses the concept of TP to transform the energy consumption minimization problem into UAVs deployment subproblem and resource allocation subproblem.
2. First, the TP model in multi-UAV assisted MEC architecture is established. The historical dataset of an intelligent traffic monitoring system is used to train the TP model parameters through LSTM algorithm.
3. In this system, we propose an adaptive differential evolution algorithm to optimize the UAVs deployment. Based on the TP results and UAVs deployment scheme, this paper adopts binary offloading scheme and greedy algorithm to optimize offloading decision and resource allocation, so as to obtain the optimal resource allocation scheme with lower energy consumption.
4. The numerical results based on the real traffic monitoring system dataset show that our proposed TpDeRas algorithm can reduce the total energy consumption while minimizing the response time to complete a task.

The rest of the paper is organized as follows. The Sect. 2 introduces the system model, communication model and computation model, and then formulates the problem. The Sect. 3 describes the TpDeRas in detail. In Sect. 4, simulation studies are undertaken to verify the performance of the proposed TpDeRas algorithm. Section 5 offers conclusions.

2 System Model and Problem Formulation

This section briefly introduces the system model, communication model, computation model and problem formulation.

2.1 System Model

Figure 1 shows the computing offloading architecture of multi-UAV assisted MEC. In this system, $J = \{1, 2, \ldots, J\}$ represents the set of terminal devices (TDs), j represents the different TD. $M = \{1, 2, \ldots, M\}$ represents the set of UAVs, and m represents the different UAV. Using the three dimensional Cartesian coordinates, H represents the fixed flight height of the UAV, (X_m, Y_m, H) represents the position of the UAV m and $(x_j, y_j, 0)$ represents the position of the TD j. T represents the network time slot. In each network time slot, the task generated by TD j can be represented by $U_j = (x_j, y_j, F_j, D_j)$, where F_j and D_j represent the computing resources and data amount required by the task U_j.

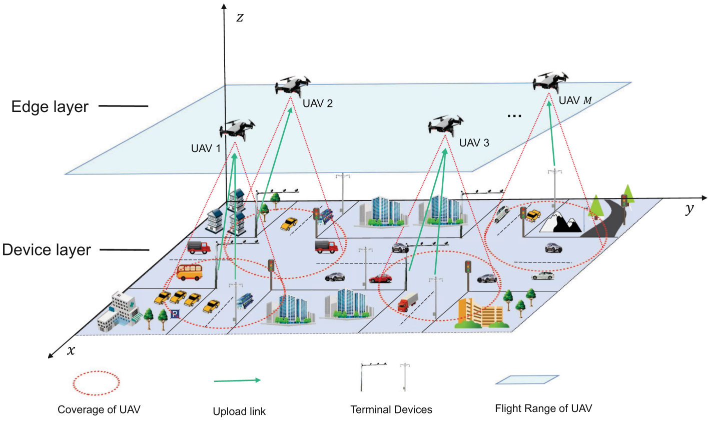

Fig. 1. The computing offloading architecture of multi-UAV assisted MEC.

In multi-UAV assisted MEC, UAVs are used as flight edge base station. We consider using binary offloading mode, the tasks generated by TDs can only be processed locally or offloaded to UAV, which cannot be separated. Thus, each task U_j has $(M + 1)$ execution patterns denoted as $K = \{0, 1, \ldots, M\}$. The binary array $a_{j,k}$ represents the task execution strategy. It can be expressed as follows.

$$\sum_{j \in J} \sum_{k \in K} a_{j,k} = \begin{cases} 1, & \text{task is executed on locally or UAV } m \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

In the computing offloading scheme, we consider the communication energy consumption, computation energy consumption and hovering energy consumption of UAV. Therefore, the task requests of TDs are more likely to be allocated to an edge base station with lower energy consumption within the delay constrained range.

2.2 Communication Model

Because UAV has a certain flight height, UAV base station is easier to use when establishing a Los link with ground users than ground base station. The Los communication channel model is used to model the channel between UAV base station and end users. In the case of Los, the Euclidean distance $L_{j,m}$ between the UAV base station m and the TD j can be expressed as follows.

$$L_{j,m} = \sqrt{(X_m - x_j)^2 + (Y_m - y_j)^2 + H^2} \tag{2}$$

Based on the path loss, the channel gain $g_{j,m}$ between the UAV m and the TD j is expressed as follows.

$$g_{j,m} = g_0 * L_{j,m}^{-2} = \frac{g_0}{(X_m - x_j)^2 + (Y_m - y_j)^2 + H^2} \tag{3}$$

The g_0 represents the channel power gain in the reference distance $d_0 = 1\text{m}$. According to the channel gain, the data transmission rate $r_{j,m}$ between UAV m and TD j can be expressed as follows.

$$r_{j,m} = B * log_2(1 + \frac{P_{j,m} * g_{j,m}}{\sigma^2}) \tag{4}$$

where, B represents the channel bandwidth, σ represents the additive Gaussian white noise, and $P_{j,m}$ represents the transmission power of the TD j. The transmission delay required for the TD j task offloading to the UAV m can be expressed follows.

$$t_{j,m}^{tran} = \frac{D_j}{r_{j,m}} \tag{5}$$

Thus, the transmission energy consumption required for the TD j task offloading to the UAV m can be expressed follows.

$$E_{j,m}^{tran} = P_{j,m} * \frac{D_j}{r_{j,m}} \tag{6}$$

2.3 Computation Model

In the time slot T, the TD j generates a task to be executed, which can be processed locally or offloaded to the UAV base station m processing.

When $a_{j,0} = 1$, the TD j task selects local execution, the computing delay to complete task U_j is defined as follows.

$$t_{j,0}^{local} = \frac{F_j}{f_{j,0}} \tag{7}$$

In addition, the computation energy consumption can be expressed as follows.

$$E_{j,0}^{local} = k_1 * f_{j,0}^{v-1} * F_j \tag{8}$$

where, k_1 represents the effective switched capacitor, where $f_{j,0}$ represents the calculation strength assigned by the TD j, v represents the positive constraint.

When $a_{j,m} = 1$, the task of TD j is offloaded to UAV base station m for execution. UAV m will be allocated the corresponding calculation strength $f_{j,m}$, namely CPU cycle. Before processing the task, it is necessary to load the virtual machine into the UAV base station m. Assuming that the service loading time has been determined for different known tasks, that time can be expressed as t_j^{load}. The computing delay required for UAV base station m to process the TD j tasks can be expressed as follows.

$$t_{j,m}^{UAV} = t_j^{load} + \frac{F_j}{f_{j,m}} \tag{9}$$

Thus, the UAV computation energy consumption can be expressed as follows.

$$E_{j,m}^{UAV} = P_0 * t_j^{load} + k_2 * f_{j,m}^{v-1} * F_j \tag{10}$$

where, k_2 represents the effective switched capacitor, P_0 represents the loading power of UAV when the service is loaded.

The total delay required for UAV base station m to complete the TD j tasks can be expressed as follows.

$$t_{j,m}^{MEC} = t_{j,m}^{tran} + t_j^{load} + \frac{F_j}{f_{j,m}} \tag{11}$$

In addition, the total energy consumption required for the UAV base station m to complete the TD j tasks can be expressed as follows.

$$E_{j,m}^{MEC} = P_{j,m} * \frac{D_j}{r_{j,m}} + P_0 * t_j^{load} + k_2 * f_{j,m}^{v-1} * F_j \tag{12}$$

2.4 Problem Formulation

The total energy consumption also includes the hovering energy consumption of UAV. Assuming that P^H is the hovering power, the hovering energy consumption of UAV can be expressed as follows.

$$E^H = P^H * T \tag{13}$$

For all computing tasks, there is a minimum energy consumption as determined by the optimal UAVs deployment and optimal resource allocation. In T, the total energy consumption required to complete all tasks can be expressed as P1.

$$\textbf{P1} \quad min \sum_{j=1}^{J} (a_{j,0} E_{j,0}^{local} + \sum_{m=1}^{M} a_{j,m} E_{j,m}^{MEC}) + ME^H. \tag{14}$$

$$\textbf{s.t.} \quad \sum_{k \in K} a_{j,k} = 1, \forall j \in J \tag{15}$$

$$M \geq M_{min} \tag{16}$$

$$\sqrt{(X_{m1} - X_{m2})^2 + (Y_{m1} - Y_{m2})^2} \geq d_{min}, \forall m1 \neq m2 \tag{17}$$

$$a_{j,m} * \sqrt{(X_m - x_j)^2 + (Y_m - y_j)^2} \leq H \tan\theta, \forall a_{j,m} = 1 \tag{18}$$

$$\sum_{j=1}^{J} a_{j,m} \leq n_{max}, \forall a_{j,m} = 1, m \in M \tag{19}$$

$$0 < f_{j,0} \leq f_{max}^{local}, \forall a_{j,0} = 1 \tag{20}$$

$$\frac{F_j}{f_{j,0}} \leq T_{max}, \forall a_{j,0} = 1 \tag{21}$$

$$f_{j,m} > 0, \forall a_{j,m} = 1 \tag{22}$$

$$\frac{D_j}{r_{j,m}} + t_j^{load} + \frac{F_j}{f_{j,m}} \leq T_{max}, \forall a_{j,m} = 1 \tag{23}$$

$$f_{j,k} = 0, \forall a_{j,k} = 0, j \in J, k \in K \tag{24}$$

Here, constraint (16) and constraint (17), respectively, indicate that the number of UAVs in the multi-UAV assisted MEC system has a certain constraint, and also that any two UAVs must keep a certain distance to avoid collision. constraint (24) indicates that when the task cannot execute $(M + 1)$ pattern, the allocated computing resource is equal to 0. P1 is a energy efficient optimization problem that combines UAVs deployment with resource allocation. The problem is non-convex and nonlinear, so it is difficult to achieve the optimal solution.

3 Proposed Solution

From the problem formulation in the Sect. 2, problem P1 is a NP-hard problem of non-convex and nonlinear. In order to solve P1 effectively, we decompose the original problem P1 into UAVs deployment subproblem and resource allocation subproblem. In the past, the solutions will increase the response time and energy consumption to complete the task. Therefore, We propose the TpDeRas method to solve the optimization problem, whose framework flow is shown in Fig. 2, where g represents the number of incompleted tasks. First, we use the task prediction (TP) algorithm to make real-time dynamic prediction of the generated task flow of the terminal devices. Then, based on the predicted task set, we optimize the UAVs deployment subproblem using the adaptive differential evolution algorithm and the resource allocation subproblem using efficient greedy algorithm in advance.

The computing offloading framework of a multi-UAV assisted MEC system based on our proposed TpDeRas in this paper is presented in Algorithm 1. First, the TP algorithm is used to predict task of the access device in T_{n+1} according to the current TD and the generated task, so as to get the number of TDs $J^{'} = \left\{1, 2, \cdots, j^{'}\right\}$ and task set $U^{'} = \left\{U_1, U_2, \cdots, U_{j^{'}}\right\}$ in advance. Afterward, according to the predicted $\left\{J^{'}, U^{'}\right\}$, the number of UAVs M and UAVs position P are initialized. During the initial UAVs deployment, because the flight height of UAV is fixed, the two-dimensional coordinate position of each UAV is coded as an individual one to simplify the complexity of the algorithm. Then, the resource allocation scheme $\{a, f\}$ is obtained based on the predicted $\left\{J^{'}, U^{'}\right\}$ and UAVs deployment $\{M, P\}$. According to $\left\{J^{'}, U^{'}, M, P, a, f\right\}$, the solution of the objective function (14) is obtained.

When the number of evaluations are fewer than the maximum number of evaluation (ENs_{max}) optimization, problem enters the joint optimization steps (9–38). The UAVs deployment optimization includes two aspects: the number of UAVs adaptive optimization steps (10–15) and the position optimization step

Algorithm 1. Computing Offloading Framework of TpDeRas

```
Input: the number of TDs $J$ and task set $U$ in current time slot $T_n$;
Output: UAVs deployment scheme $\{M, P\}$, resource allocation scheme $\{a, f\}$ in the next slot $T_{n+1}$;
Generate $J^{'}$ and $U^{'}$ in advance through Algorithm 2;
Initialize $M$ and randomly generate an initial $P$ under constraints (17);
Generate $\{a, f\}$ according to $\left\{J^{'}, U^{'}, M, P\right\}$ by Algorithm 4;
Evaluate the total energy consumption according to $\left\{J^{'}, U^{'}, M, P, a, f\right\}$, get number of incompleted tasks $g$;
If $g = 0$, then $IC = 0$; Otherwise $IC = 1$ $//IC$ is the times of consecutive incompleted tasks in the number of evaluations
Initialize $ENs = 0, Status = 0$;
while $ENs < ENs_{max}$ do
    if $g = 0$ and $M > M_{min}$ and $Status = 0$ then
        Select two individuals with the minimum Euclidean distance, calculate the second smallest individual, delete the position, and get the updated $P$;
        $M = M - 1$;
        Generate resource allocation $\{a, f\}$ by using Algorithm 4;
        Evaluate the performance of this operation£»
        $ENs = ENs + 1$;
    else
        Generate an offspring UAVs deployment $P^{'}$ by using Algorithm 3;
        for $i \in M$ do
            Replace the position of a random individual in $P$ with the $i$th individual in $P^{'}$, and get a UAVs deployment $S$.
            if $S$ satisfies constraints (17) then
                Generate resource allocation $\{a, f\}$ by using Algorithm 4;
                Evaluate the performance of this operation; if the performance of $S$ is better than $P$ , $P = S$;
                $ENs = ENs + 1$;
                if $g \neq 0$ then
                    $IC = IC + 1$;
                    while $IC = IC_{max}$ do
                        $Status = 1$ and return the solution set to its last state in which all tasks can be completely completed;
                        break;
                    end while
                end if
                if $g = 0$ and $Status = 0$ then
                    $IC = 0$;
                    break;
                end if
            end if
        end for
    end if
end while
```

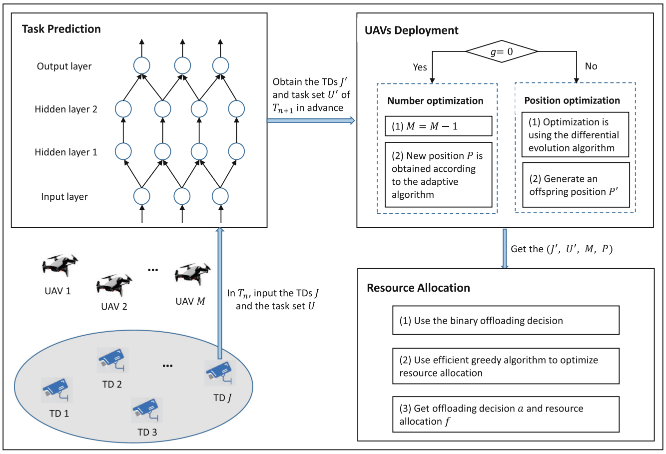

Fig. 2. Illustration of the TpDeRas algorithm framework.

(17–36). According to the UAVs deployment scheme, optimizing the resource allocation. The objective function (14) performance of each iteration is evaluated, and the optimal solution and optimal solution set for the evaluation process are recorded. When the number of evaluations reaches the maximum, then the optimal UAVs deployment scheme and resource allocation scheme are outputs.

3.1 Task Prediction Algorithm

In this paper, we propose a multi-UAV assisted MEC system based on traffic monitoring system that can predict the dynamic changes of network resources by using artificial intelligence (AI) algorithm. Thus, we propose a TP mechanism based on LSTM. Because of the limited accuracy of single layer LSTM model, we considers multi-layer LSTM model. In this paper, explained variance score is used to evaluate the effect of the model. Finally, the value of explained variance score of the two-layer LSTM is more than 0.95, which is infinitely close to 1. Therefore, increasing the number of layers of the LSTM will only greatly increase the model complexity, and the improvement of explained variance score is limited.

The processing of a single LSTM cell consists of forgetting gate, input gate and output gate. The principle of LSTM method is stated as follows.

$$f_t = \sigma\left(W_f[h_{t-1}, x_t] + b_f\right). \tag{25}$$

$$i_t = \sigma\left(W_i[h_{t-1}, x_t] + b_i\right). \tag{26}$$

$$C_t^{'} = tanh\left(W_C[h_{t-1}, x_t] + b_C\right). \tag{27}$$

$$C_t = f_t * C_{t-1} + i_t * C_t^{'}. \tag{28}$$

$$o_t = \sigma\left(W_o[h_{t-1}, x_t] + b_o\right). \tag{29}$$

$$h_t = o_t * tanh\left(C_t\right). \tag{30}$$

where, f_t, C_t, h_t represent forgetting gate, input gate and output gate. W_f, W_o and b_f, b_o represent the weights and bias vector of forgetting gate and output gate. W_i and b_i represent the weight vector and bias vector of the input gate. W_C and b_C is the weight vector and bias vector of input gate output state. Where, σ and $tanh$ are activation functions. h_t is the traffic feature of the output gate, used to calculate the next cell.

The cloud computing platform has the historical task record of the TDs, and obtains the record as the dataset R. On the one hand, the dataset is transformed into two-dimensional dataset by using a dimension reduction operation on cloud platform. The time information for dataset is used to process the data, and the dataset of each TD is sorted from time information data R_j into time series data $R_j^{'}$. The characteristics of each line include the device ID, time, and amount of data. The time information is extracted and the dataset is normalized. The dataset is divided into a training set and a test set according to the ratio 9:1. RMSprop algorithm is used to optimize the TP model parameters, and Mean Squared Error (MSE) is used as the loss function to calculate the loss. The parameters of the LSTM model are obtained through training and sent to the TD. On the other hand, according to the LSTM parameters as trained by the cloud computing platform, the TD uses the current task to obtain the task of the next slot through the TP model.

The TP algorithm can be used to predict the amount of data generated by the TD. According to the predicted task set $U^{'}$ and the characteristics of UAV, the UAVs deployment scheme and resource allocation scheme can be obtained in advance. Before the next slot task arrives, the UAV node loads the service virtual machine in advance, which greatly reduces the service loading time.

3.2 UAVs Deployment Subproblem

The UAVs deployment subproblem mainly considers two aspects: The number of UAVs optimization and the UAVs position optimization. In this paper, we use the Differential Evolution (DE) algorithm to optimize UAVs deployment scheme. Among them, the number of UAVs is equivalent to the number of populations. Because the mechanical energy consumption (hovering energy consumption) of UAVs accounts for a large proportion of the total energy consumption, the number of UAVs is optimized by adaptive adjustment method (step 10–15) in algorithm 1, so as to minimize the number of UAVs when all the tasks can be completed. In addition, the UAVs position optimization is based on the UAV two dimensional coordinate. However, the search space is too large when using DE algorithm for position optimization. Inspired by Wang et al. [11], we use coding mechanism to encode each UAV position (X, Y) into an individual one, which can greatly reduce the complexity of the algorithm. $P(x, g)$ represents the UAVs position in the gth generation, and it can be expressed as follows.

$$P(x, g) = \left\{x_1(g), x_2(g), \cdots, x_m(g), \cdots, x_M(g)\right\}. \tag{31}$$

where, $x_m(g), m \in M$ represents the two dimensional coordinate position of gth generation UAV m. DE algorithm is an optimization algorithm based on group intelligence, and its evolution process includes mutation, crossover and selection. First, the mutation operation proceeds as follows.

$$h_m(g) = x_{m1}(g) + F * (x_{m2}(g) - x_{m3}(g)). \tag{32}$$

where, F is the scaling factor, and $x_m \neq x_{m1} \neq x_{m2} \neq x_{m3}$.

Secondly, the crossover operation proceeds as follows.

$$v_m = \begin{cases} h_m(g), & rand(0,1) \leq CR, \\ x_m(g), & else. \end{cases} \tag{33}$$

where, CR is a cross control factor.

Finally, the resource allocation algorithm is used to evaluate the performance of the new and old UAVs deployment, and the best UAVs deployment scheme is selected as the next generation using the selection operation. The selection operation is as follows.

$$P_m(g+1) = \begin{cases} V_m(g), & if E(V_m(g)) \leq E(P_m(g)), \\ P_m(g), & else. \end{cases} \tag{34}$$

$E(.)$ represents the total energy consumption from completing all the tasks after executing the resource allocation algorithm according to the current UAVs deployment. Thus, the UAVs position optimization algorithm is shown in Algorithm 2 below.

Algorithm 2. UAVs Deployment Subproblem

```
Input: the number of UAVs $M$, UAVs position $P$;
Output: offspring UAVs deployment $P'$;
Define scaling factor $F$ and crossover control factor $CR$;
Coding the two dimensional position of UAV $m$ into an individual $x_m$;
for $m \in M$ do
    Randomly select three different individuals to execute formula (32) to generate a new mutation vector in population $P$;
    Check whether the mutation vector is within the range of UAVs deployment;
    Perform formula (33) to get the new individual $v_m$;
end for
Return offspring UAVs deployment $P'$.
```

3.3 Resource Allocation Subproblem

Using the TP model in T_n can predict the number of TDs J' and task set U' in T_{n+1}, so that the resource can be pre-allocated in T_n.The task completion time for the offloading decision can be expressed as follows.

$$t_{j,m}^{MEC} = t_{j,m}^{tran} + \frac{F_j}{f_{j,m}} \tag{35}$$

According to the delay constraints and UAVs performance constraints, the next slot offloading decision and resource allocation scheme can be obtained in T_n. In P1, $\left\{J^{'}, U^{'}, M, P\right\}$ can be determined based on the TP algorithm and UAVs deployment scheme. Thus, the resource allocation subproblem of T_{n+1} can be expressed as P2 in T_n.

$$\mathbf{P2} \quad min \sum_{j=1}^{J} (a_{j,0} E_{j,0}^{local} + \sum_{m=1}^{M} a_{j,m} E_{j,m}^{MEC}). \tag{36}$$

$$\mathbf{s.t.} \quad \sum_{k \in K} a_{j,k} = 1, \forall j \in J \tag{37}$$

$$a_{j,m} * \sqrt{(X_m - x_j)^2 + (Y_m - y_j)} \leq H \tan\theta, \forall a_{j,m} = 1 \tag{38}$$

$$\sum_{j=1}^{J} a_{j,m} \leq n_{max}, \forall a_{j,m} = 1, m \in M \tag{39}$$

$$0 < f_{j,0} \leq f_{max}^{local}, \forall a_{j,0} = 1 \tag{40}$$

$$\frac{F_j}{f_{j,0}} \leq T_{max}, \forall a_{j,0} = 1 \tag{41}$$

$$f_{j,m} > 0, \forall a_{j,m} = 1 \tag{42}$$

$$\frac{D_i}{r_{j,m}} + \frac{F_j}{f_{j,m}} \leq T_{max}, \forall a_{j,m} = 1 \tag{43}$$

$$f_{j,k} = 0, \forall a_{j,k} = 0, j \in J, k \in K \tag{44}$$

In the process of solving P2 problem, two variables need to be optimized, namely $a_{j,k}, f_{j,k}$. In paper, we use a double greedy algorithm to solve resource allocation subproblem.

Therefore, when the computing task needs to be offloaded to the UAV node for processing, the candidate sets of each computing task are arranged in descending order, the TD with the minimum candidate set is selected, and then the UAV with the minimum energy consumption is selected according to the candidate set for offloading.

4 Performance Evaluation

In this section, we use Python simulation platform to simulate our proposed TpDeRas based on the real traffic monitoring scene. In the parameter setting, 50–100 TDs are randomly distributed in the square area with a side length of 500m. According to the real traffic monitoring dataset, the amount of each task is within 60M. The virtual service loading time for tasks generated by different

TDs is set at 20–30ms. In this paper, we provides 4–20 UAVs as edge base station to provide communication and computation services for ground users. Because the mechanical energy consumption of UAV is much greater than the calculated energy consumption when completing the task, the hovering power of UAV is set as $p^H = 1500$.

Based on the above conditions, we use three benchmark comparison algorithms for a comparison with our proposed TpDeRas algorithm, which are ToDeTas algorithm [11], shortest job first (SJF) algorithm and Full Offloading algorithm.

4.1 Performance Comparison of System Energy Consumption

In parameter setting, in order to compare the system performance of different approach better, the flight height of UAV is set at $H = 120$m, and the maximum number of iterations of joint optimization is set at 10000. We evaluate the total energy consumption under different approaches by changing the number of terminal equipment and the processing capacity of UAV.

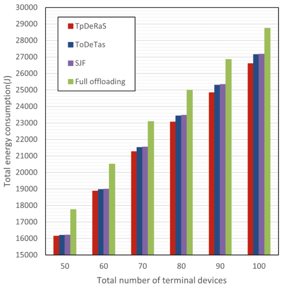

Fig. 3. Comparison of total energy consumption for different terminal devices.

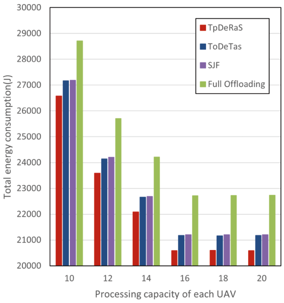

Fig. 4. Comparison of total energy consumption under different processing capacity.

Figure 3 shows the total energy consumption for completing all tasks with different terminal devices. It can be seen that the total energy consumption is proportional to the number of terminal devices, and the total energy consumption in our proposed TpDeRas is lower than the other three algorithms. This is because when the number of terminal devices decreases, the number of UAVs required decreases due to the reduction of task set, and the number of UAVs is positively related to the total energy consumption when completing tasks.

Figure 4 shows the total energy consumption for completing all tasks under different processing capacity. With the enhancement of UAV processing capacity, the tasks completed by each UAV increase, which can dynamically reduce the number of UAVs required, and the mechanical energy consumption of UAV accounts for a large proportion in completing all tasks. Therefore, as the processing capacity increases, the energy consumption required to complete all tasks decreases.

4.2 Performance Comparison of Average Response Time

The average response delay mainly includes task transmission delay and service loading delay. We evaluate the average response time of different approaches by changing the processing capacity of UAV and amount of data.

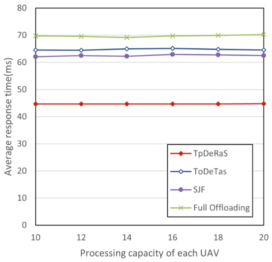

Fig. 5. Comparison of average response time with different processing capacity.

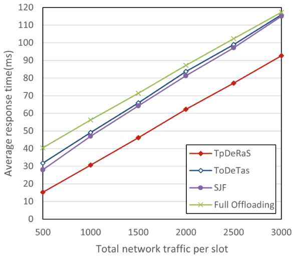

Fig. 6. Comparison of average response time under different network traffic.

In this paper, each UAV provides a certain number of channels. If the number of channels becomes exhausted, the UAV cannot receive the computing task of ground users in the current time slot. Therefore, the channel number of the UAV is equivalent to the processing capacity of a UAV. Figure 5 shows the average response time to complete the task with a different processing capacity. It can be seen that the average response time of our proposed TpDeRas is much less than that of the SJF algorithm and ToDeTas algorithm. The main reason is that our proposed TpDeRas can predict the task set generated by the TDs in advance before the next time slot task arrives, and the UAV can load the virtual machine in advance before the real task arrives, so as to reduce the service loading delay. Therefore, the response delay in our proposed TpDeRas is mainly related to the task transmission delay and the accuracy of the TP model.

Figure 6 shows the average response time to complete tasks under different network traffic flows. Because the transmission delay is mainly related to the UAVs deployment and the amount of task data, the average response time will change with the increase of network traffic. It can be seen that the average response time increases with the increase of network traffic, but the average response time of our proposed TpDeRas is always far lower than that of other algorithms that have been in use previously. Compared with the best SJF algorithm and ToDeTas algorithm, the proposed TpDeRas can basically achieve “no perception" delay.

5 Conclusion

In this paper, we have proposed a multi-UAV assisted MEC system. To minimize the total energy consumption of the system, both UAVs deployment and resource allocation are jointly optimized. We combine deep learning with an evolutionary algorithm to propose an optimization method called TpDeRas. Simulation results show that our proposed TpDeRas is superior to other algorithms and can effectively reduce the system energy consumption and greatly reduce the average response time to complete the task.

References

1. Ning, Z., et al.: Mobile edge computing enabled 5G health monitoring for internet of medical things: a decentralized game theoretic approach. IEEE J. Sel. Areas Commun. **39**(2), 463–478 (2020)
2. Wang, X., Ning, Z., Guo, L., Guo, S., Gao, X., Wang, G.: Mean-field learning for edge computing in mobile blockchain networks. IEEE Trans. Mob. Comput. **22**, 5978–5994 (2022)
3. Ning, Z.: Dynamic computation offloading and server deployment for UAV-enabled multi-access edge computing. IEEE Trans. Mob. Comput. **22**(5), 2628–2644 (2021)
4. Mohsan, S.A.H., Khan, M.A., Noor, F., Ullah, I., Alsharif, M.H.: Towards the unmanned aerial vehicles (UAVs): a comprehensive review. Drones **6**(6), 147 (2022)
5. Wu, W., Zhou, F., Wang, B., Wu, Q., Dong, C., Hu, R.Q.: Unmanned aerial vehicle swarm-enabled edge computing: potentials, promising technologies, and challenges. IEEE Wirel. Commun. **29**(4), 78–85 (2022)
6. Yu, Z., Gong, Y., Gong, S., Guo, Y.: Joint task offloading and resource allocation in UAV-enabled mobile edge computing. IEEE Internet Things J. **7**(4), 3147–3159 (2020)
7. Hu, X., Wong, K.K., Yang, K., Zheng, Z.: UAV-assisted relaying and edge computing: scheduling and trajectory optimization. IEEE Trans. Wireless Commun. **18**(10), 4738–4752 (2019)
8. Li, W.T., et al.: Collaborative offloading for UAV-enabled time-sensitive MEC networks. EURASIP J. Wirel. Commun. Netw. **2021**, 1–17 (2021)
9. Wang, X., Ning, Z., Guo, S., Wen, M., Guo, L., Poor, V.: Dynamic UAV deployment for differentiated services: a multi-agent imitation learning based approach. IEEE Trans. Mob. Comput. **22**(4), 2131–2146 (2021)
10. Lin, W., Huang, T., Li, X., Shi, F., Wang, X., Hsu, C.H.: Energy-efficient computation offloading for UAV-assisted MEC: a two-stage optimization scheme. ACM Trans. Internet Technology (TOIT) **22**(1), 1–23 (2021)
11. Wang, Y., Ru, Z.Y., Wang, K., Huang, P.Q.: Joint deployment and task scheduling optimization for large-scale mobile users in multi-UAV-enabled mobile edge computing. IEEE Trans. Cybern. **50**(9), 3984–3997 (2019)