2021 7th International Conference on Big Data Computing and Communications (BigCom) | 978-1-6654-4252-7/21/$31.00 ©2021 IEEE | DOI: 10.1109/BigCom53800.2021.00004

Energy Efficient Resource Allocation and Trajectory Optimization in UAV-Assisted Mobile Edge Computing System

*

Shuaibing Chen*†, Lei Shi*†, Xu Ding*†, Zengwei Lv*†, and Zhehao Li*†
*School of Computer Science and Information Engineering, HeFei University of Technology, Hefei 230009, China
†Engineering Research Center of Safety Critical Industrial Measurement and Control Technology, Ministry of Education, Hefei 230009, China
Email: dingxu@hfut.edu.cn

***Abstract*—Mobile edge computing(MEC) has been considered as a promising technology with the ever-increasing computation demands, which offloads computation-intensive tasks to MEC servers to meet the low latency and high bandwidth requirements of the tasks. But considering the dynamic UEs, nonuniform distribution of task requests and the limitation of the dynamic of the fixed deployment of the edge severs, we investigate a Unmanned Aerial Vehicles (UAVs)-assisted edge computing system in this paper, where each UAV is equipped with server to assist local edge servers. Utilizing the mobility of UAVs provides flexible edge computing services. In this model, tasks are executed on the local edge server first. When the computing resources of the local edge server cannot meet the computational requirements of the task, the system will dispatch the UAVs. Task will be offloaded to UAV for execution. Considering that the endurance of UAVs is a tough problem under the current technical level. Our aim is to minimize the energy consumption of UAVs under the premise of satisfying the UEs demands as much as possible to achieve a higher resource utilization rate. We propose Tasks Offloading Policy Algorithm(TOPA) and Online UAVs Dispatching Base on the Shortest Distant Algorithm(ODSH). Simulation results show the effectiveness of the proposed of algorithms.**

***Index Terms*—mobile edge computing, UAVs-assisted, energy consumption of UAVs, task allocation**

I. Introduction

In recent years, with the increasing popularity of smart mobile devices (such as mobile phones, wearable devices and smart cameras), the Internet of Things (IoT) technology has been booming. At the same time, in order to support a large number of mobile smart devices and process large amounts of data in time, MEC provides UEs with shorter response times, higher bandwidth and better reliability by placing computing resources close to the mobile devices [1]–[3]. However, the traditional edge server is fixed, and when the UE is not within the coverage of the edge server, the edge server will not be able to provide services for UEs. Because UEs are not evenly distributed, it is inevitable that somewhere the server will not be able to meet each UE's requirement [4]. Because 5G base station is very expensive, we should make reasonable use of the computing resources of the base station. But it is inevitable to encounter emergency situation where there is not enough computing resources and there is not even an edge server, such as forest fire fighting, emergency rescue and military training.

Fortunately, in recent years, in order to overcome this limitation, UAV-assisted multi-access edge computing has been proposed and conceived as a potential technology to overcome this challenge [5], [6]. Compared with cellular infrastructure-based edge computing, UAV-assisted edge computing possesses more reliable line-of-sight(LoS) links, and controllable mobility management [7], [8]. In [9], [10], the UAV was primarily used as a communication relay to enhance computational unload flexibility. There are more and more researches on UAVs as mobile servers providing edge computing services [6], [11]. In [12], author considers a UAV assisted multi-access edge computing system, a scheme based on game theory is proposed to optimize the weighted values of time delay and energy consumption uniformly. In [13], based on Hungarian algorithm, author proposed optimal task-UAV-edge server matching algorithm to minimize the energy consumption and processing time. In addition, the NOMA-based and UAV-assisted MEC system have also been studied in [14]. Due to the limited energy of UAVS, the energy efficiency of UAVs should also be considered in mission offloading [15]. Under the current technical conditions, UAVs generally use batteries as their energy source, but they are characterized by low power and high transmission rate, but there are shortcomings in transmission [16], [17]. In [4], author propose a novel UAVs assisted edge server scheme, which provides a flexible edge computing service and can achieve high resource utilization. In [18], The author proposes a dynamic computing offload strategy, which takes into account both task delay and execution failure as performance indicators, and presents an online algorithm with lower complexity. In [19], The author studies the user association, power control, computing resource allocation and joint planning of UAVs position in the multi-UAVs collaborative edge computing scenario. In [20], author studied how to use TDMA protocol to sovle the problem

Identify applicable funding agency here. If none, delete this.

978-1-6654-4252-7/21/$31.00 ©2021 IEEE
DOI 10.1109/BigCom53800.2021.00004

of joint deployment and task offloading of UAVs in UAV-assisted edge computing network. In [21], author studied the problem of energy consumption minimization of devices and UAVs in the system through joint optimization of task unloading, resource allocation and UAVs trajectory. In [22], author propose a network structure of UAVs' collaborative MEC, in which UAVs can assist each other to perform computing tasks, and a scheme for offloading the collaborative tasks. In [23], author proposed an unmanned aerial vehicle (UAV)-aided mobile edge computing (MEC) framework, and proposed a multi-agent deep reinforcement learning based trajectory control algorithm. In multi-UAV assisted system, UAV can help users offload tasks to the cloud for faster execution. In [24], author investigated the UAVs' intermediate relay scheme with dual constrains of QoE and battery limitation. In [25], in this paper, author investigated the minimized average weighted sum problem of joint planning computing offloading, resource allocation and trajectory scheduling in UAV-assisted MEC systems. In other UAV assisted MEC system studies, [26] proposed UAV on-demand coverage deployment. In [27], author proposed an algorithm to control the regular topology of multiple mobile nodes. [28] studied the UAV-assisted MEC system, and proposed the problem of minimizing tthe sum of the maximum delay among all the users in each time slot by jointly the unloading ratio, user scheduling and UAV trajectory. [29] studied the problem of the user offloading bits to the UAV to the maximum. In [30], the UAVs can help calculate the latency-critical task bits for TDs offloading, and they can also act as a relay to help computation offloading.

Previous work have made some contributions in MEC. However, most of them only consider fixed edge servers. Existing strategy of adding UAVs to edge computing also mostly consider UAVs alone. However, in real life, the cost of the UAV with the server is relatively high. If the UAV is only considered, it will cause great cost pressure. In this paper, we consider a novel hybrid solution for UAV-assisted local edge servers. The local edge server process the tasks under normal circumstances. When there is an emergency situation that the local server cannot meet the user's service requirements, the UAV is dispatched to assist the local edge server. In the existing researches on the UAV-assisted edge server, most of them also study the task delay correlation, while ignoring the UAVs themselves. Due to the limitations of the current technical conditions, the endurance of the UAVs makes us unable to ignore the problem. In this paper, we consider to minimize the energy consumption of the system UAVs under the premise of satisfying the UEs' demands as much as possible. Firstly, the mathematical model is established, because it is difficult to find the optimal solution directly. We designed TOPA algorithm and ODSH algorithm based on greedy strategy to find the approximate optimal solution.

The rest of the article is organized as follows. The section 2 introduces the system model and calculation model. In section 3, the TOPA algorithm and ODSH algorithm are proposed. In section 4, we give the simulation results and analyze them. In section 5, we summarize this paper.

II. System Model and Problem Formulation

A. System Model

We first describe the system model(see Figure 1). Consider a UAV-Assisted network consists of N edge servers, M UAVs and several user devices (UEs) in a regular region. The region has N equal areas. And the edge server are uniformly deployed and each server has its own coverage, i.e., each small square area is configured with an edge server. UEs will generate computing tasks. The tasks can be offloaded to edge servers or UAVs. And these tasks are often typically delay sensitive and require completion before a deadline. Consider the dynamic UEs, nonuniform distribution of task requests and the limitation of the dynamic of the fixed deployment of the edge severs, if UE is not covered by the available edge servers, the task will require multiple hops to complete. Therefore, in order to avoid the occurrence of multiple hops, UAVs that are equipped with edge servers is introduced to assist the edge servers to provide flexible edge computing services. When the local edge server cannot meet all the task requirements in the area, the area will makes a service request to the UAV, the UAV will fly to the area to provide computing service. We consider the servers equipped on UAVs have the same specifications as edge servers. And all UEs devices have the same specifications. The sets of the edge servers, UAVs and the small square areas are denoted by $\mathcal{N}$ (with $|\mathcal{N}| = N$), $\mathcal{M}$ (with $|\mathcal{M}| = M$) and $\mathcal{Z}$ (with $|\mathcal{Z}| = Z$), respectively. Denote a_n as an edge server, $a_n \in \mathcal{N}$. Denote b_m as an UAV, $b_m \in \mathcal{M}$. Denote c_z as a small area, $c_z \in \mathcal{Z}$. We define area c_z to be configured with edge server a_z. Denote T seconds as the computing cycle. Suppose the whole computing cycle T is divided into K time slots equally, and the sets of time slots are denoted $\mathcal{K}$, where $\mathcal{K}(1, ..., K)$ as these time slots. Thus, the time length for a slot is T/K, which is denote by τ. Each task starts at a certain time slot. Denote $\mathcal{U}_z$ (with $|\mathcal{U}_z| = U_z$) as the set of the tasks of c_z. Denote $s_{z,k,u}$ as the task u of c_z at time slot k, $u \in \mathcal{U}_z$. We want to minimize the energy consumption of the UAVs in the system under the premise of maximizing the satisfaction of UEs needs. The energy consumption of the UAV includes the flight energy consumption, hovering energy consumption and computing energy consumption.

B. Network Layer Model

Define a new set $\mathcal{M}' = \{0, 1, ..., M\}$ as the places where the tasks might offload. When $m = 0$, the task will be offloaded to the local server. Define $c^m_{z,k,u}$ as the offloading indicator variables of task $s_{z,k,u}$,we have

$$c^m_{z,k,u} \in \{0, 1\}, z \in \mathcal{Z}, k \in \mathcal{K}, u \in \mathcal{U}_z, m \in \mathcal{M}', \quad (1)$$

where $c^m_{z,k,u} = 1$, $m \neq 0$ means that task $s_{z,k,u}$ is offloaded to UAV b_m at the time slot k, $c^m_{z,k,u} = 1$, $m = 0$ means that $s_{z,k,u}$ is offloaded to local server a_z, and otherwise $c^m_{z,k,u} = 0$. Note that

$$\sum_{m=0}^{M} c^m_{z,k,u} = 1. \quad (2)$$

TABLE I
MAIN NOTATIONS.

Symbol	Description
N	Number of edge servers
M	Number of UAVs
Z	Number of square areas
U	Number of UEs tasks
$\mathcal{N}$	Set of edge servers, and $N = \|\mathcal{N}\|$
$\mathcal{M}$	Set of UAVs, and $M = \|\mathcal{M}\|$
$\mathcal{Z}$	Set of square areas, and $U = \|\mathcal{U}\|$
$\mathcal{U}$	Set of UEs tasks, and $U = \|\mathcal{U}\|$
a_n	Edge server n
b_m	UAV m
c_z	Square area z
$\mathcal{U}$	The set of the tasks of square area z
$s_{z,k,u}$	The task u of c_z at the time slot k
$c^m_{z,k,u}$	The offloading indicator variable of task $s_{z,k,u}$
$w_{m,k}$	The status indicator variable of UAV b_m
$T^m_{z,k,u}$	The latency requirement of task $s_{z,k,u}$
$f^m_{z,k,u}$	The CPU-cycle allocated to task $s_{z,k,u}$
$F_{z,k,u}$	Number of CPU cycles of task $s_{z,k,u}$
$D_{z,k,u}$	Data size of task $s_{z,k,u}$
$T_{z,k,u}$	Latency requirement of task $s_{z,k,u}$
e_m	The energy consumption of UAVs b_m
$e^f_{m,k}$	The flight energy consumption of the UAV b_m in the time slot k
$e^c_{m,k}$	The computing energy consumption of the UAV b_m in the time slot k
$e^h_{m,k}$	The hovering energy consumption of the UAV b_m in the time slot k

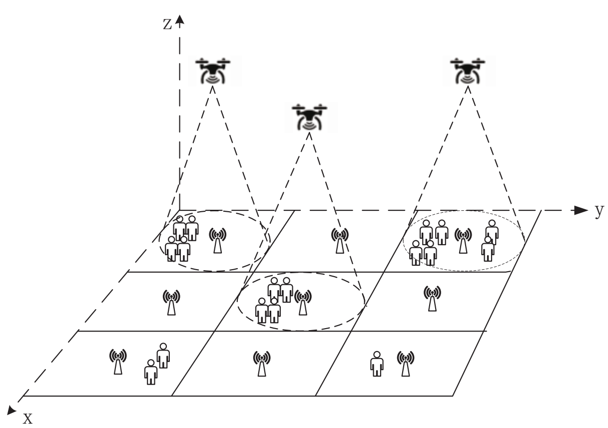

Fig. 1. System modle

This means that each task can only be executed at one place.

The quality of communication between UAVs and UEs depends on their distance. To represent their distance, we introduce a three-dimensional Cartesian coordinates. The scheduling algorithm schedules the UAV according to the task requirements. Denote $q_{m,k} = (x_{m,k}, y_{m,k})$ as the horizontal coordinate of b_m at the time slot k. The trajectory of the UAVs is composed of the position of the UAV in each time slot, i.e., $Q_m = [q_{m,1}; ...; q_{m,K}]$. We assume that all UAVs are at a fixed altitude H. Denote (x_z, y_z) as the central coordinates of c_z. For convenience, we define the coordinates of the c_z central point as the coordinates of the local edge server a_n and the horizontal projection of the UAVs pre-designated parking location. Since the acceleration period is short and for the convenience of calculation, we assume that the UAVs fly at a constant speed V, ignoring the acceleration period of the UAV. Define $w_{m,k}$ as the UAV b_m status indicator variable, we have

$$w_{m,k} = \begin{cases} 1: & q_{m,k} \neq q_{m,k-1}, \text{the UAV } b_m \text{ is flying;} \\ 0: & q_{m,k} = q_{m,k-1}, \text{the UAV } b_m \text{ is hovering.} \end{cases} \tag{3}$$

1) Communication between UEs and edge server: $c^m_{z,k,u} = 1$ and $m = 0$ denotes task $s_{z,k,u}$ is offloaded to edge servers a_z. Given that the quality of communication between the edge server and the task depends on their distance. Denote $(x_{z,k,u}, y_{z,k,u})$ as the coordinates of task $s_{z,k,u}$, B represents the channel bandwidth between UEs to edge servers. The distance between task $s_{z,k,u}$ and edge server a_z can be calculated as

$$d^m_{z,k,u} = \sqrt{(x_z - x_{z,k,u})^2 + (y_z - y_{z,k,u})^2}. \tag{4}$$

The channel gain between task $s_{z,k,u}$ to edge server a_z at t is denoted by

$$h^m_{z,k,u} = \frac{g_0}{(d^m_{z,k,u})^2}, \tag{5}$$

where g_0 is the channel power gain at the reference distance $1m$. The data transmission rate between task $s_{z,k,u}$ to edge server a_z at time slot k is denoted by

$$r^m_{z,k,u} = Blog_2\left(1 + \frac{G_0 P h^m_{z,k,u}}{\sigma^2}\right), \tag{6}$$

where B represents the channel bandwidth.$G_0 \approx 2.2846$. P is the transmitting power. σ^2 is the white Gaussian noise power.

2) Communication between UEs and UAVs: $c^m_{z,k,u} = 1$ and $m \neq 0$ denotes task $s_{z,k,u}$ is offloaded to UAV b_m. We assume that the UAV will fly to the c_z central point to provide computing services. The distance between task $s_{z,k,u}$ and edge server a_z can be calculated as

$$d^m_{z,k,u} = \sqrt{H^2 + (x_{m,k} - x_{z,k,u})^2 + (y_{m,k} - y_{z,k,u})^2}. \tag{7}$$

The channel gain between task $s_{z,k,u}$ to UAV b_m at t is denoted by

$$h^m_{z,k,u} = \frac{g_0}{(d^m_{z,k,u})^2}, \tag{8}$$

where g_0 is the channel power gain at the reference distance $1m$. The data transmission rate between task $s_{z,k,u}$ to edge server a_m at time slot k is denoted by

$$r^m_{z,k,u} = Blog_2\left(1 + \frac{G_0 P h^m_{z,k,u}}{\sigma^2}\right). \tag{9}$$

C. Computation Model

There are two options for offloading a task: 1) when $c_{z,k,u}^m = 1$ and $m = 0$, task $s_{z,k,u}$ is offloaded to the edge server a_z at time slot k. 2) when $c_{z,k,u}^m = 1$ and $m \neq 0$, task $s_{z,k,u}$ is offloaded to UAV b_m at time slot k. And the task can be formulated as $s_{z,k,u} = (F_{z,k,u}, D_{z,k,u}, T_{z,k,u})$, where $F_{z,k,u}$ indicates the number of CPU cycles of task $s_{z,k,u}$ required. $D_{z,k,u}$ indicates the data size of task $s_{z,k,u}$. $T_{z,k,u}$ indicates the latency requirement of task $s_{z,k,u}$.

$$T_{z,k,u}^m = T_{z,k,u}^{tr,m} + T_{z,k,u}^{co,m} = \frac{D_{z,k,u}}{r_{z,k,u}^m} + \frac{F_{z,k,u}}{f_{z,k,u}^m}, \quad (10)$$

where $T_{z,k,u}^m$ indicates the total duration of task $s_{z,k,u}$ from its produce to completion. $T_{z,k,u}^{tr,m}$ indicates the transmission time of task $s_{z,k,u}$. $f_{z,k,u}^m$ indicates the computation capability that the server allocated to task $s_{z,k,u}$.

$$c_{z,k,u}^m T_{z,k,u}^m \leq T_{z,k,u}. \quad (11)$$

This means that each task should be completed within the deadline.
f indicates the computation capacity of server. We assume that the UAVs and the edge servers have the same computation capacity. Then we can have

$$\sum_{u=1}^{U_z} f_{z,k,u}^m \leq f. \quad (12)$$

This means that the computing capacity assigned to the tasks by the server cannot exceed the computing capacity of the server.

D. Energy Consumption Model

For the energy consumption of UAV b_m can be expressed as following:

$$e_m = \sum_{k=1}^{K} \left(e_{m,k}^f + e_{m,k}^c + e_{m,k}^h \right), m \in \mathcal{M}, k \in \mathcal{K}, \quad (13)$$

where e_m indicates the energy consumption of UAVs b_m in the system, $e_{m,k}^f$ indicates the flight energy consumption of the UAV b_m in the time slot k, $e_{m,k}^c$ indicates the computing energy consumption of the UAV b_m in the time slot k, $e_{m,k}^h$ indicates the The hover energy consumption of the the UAV b_m in the time slot k. Then the first item $e_{m,k}^f$ can be expressed as

$$e_{m,k}^f = w_{m,k} \times P_b^f \times \tau, \quad (14)$$

where p_b^f indicates the power of UAV propulsion. For the second item $e_{m,k}^c$, it can be calculated by

$$e_{m,k}^c = \sum_{z=1}^{Z} \sum_{u=1}^{U_z} c_{z,k,u}^m k_1 \left(f_{z,k,u}^m \right)^2 D_{z,k,u}^m, \quad (15)$$

where k_1 depends on the effective switching capacitance of the UAV, which depends on the chip structure. $D_{z,k,u}^m$ indicates the UAV b_m processes the amount of data from task $s_{z,k,u}$ in the time slot k. For the last item $e_{m,k}^h$, it can be calculated by

$$e_{m.k}^h = (1 - w_{m,k}) \times P_b^h \times \tau, \quad (16)$$

where P_b^h indicates the power of UAV hovering.

Considering that the endurance of UAVs is a tough problem under the current technical level, our aim is to minimize the energy consumption of UAVs on the basis of maximizing the satisfaction of user needs. Based on Sec. 2.1, discussions, we can formulate the sum of energy consumption of all UAVs in the system minimization problem as follows

$$\begin{aligned} \min \quad & \sum_{m=1}^{M} e_m \\ \text{s.t.} \quad & (1) - (16), \end{aligned} \quad (17)$$

$$f \leq f_{max}, \quad (18)$$

$$V \leq V_{max}, \quad (19)$$

where (18) and (19) indicates the maximum CPU-cycle frequency of the server and the maximum flying speed of the UAVs constraints, respectively.

III. Algorithms

In the last section, we introduce the original problem model and find it difficult to solve it directly. In this section, we consider the issue how to reasonably allocate the computation resources of server to the tasks and to optimize the UAVs path under the premise of meeting the tasks requirements in the system. Based on these discussions, we will try to propose the heuristic algorithms to solve this problem. The main idea of our algorithms are based on iterative steps. In the following, we first introduce Tasks Offloading Policy Algorithm(TOPA) in sub section A. Then in sub section B, we introduce Online UAVs Dispatching Base On the Shortest Distant(ODSH).

A. Tasks Offloading Policy Algorithm(TOPA)

In our model, the local task will take the edge server as the first choice to unload. If the edge server does not have enough computing capacity, the region where the tasks are located will request the UAVs to assist the edge server to jointly provide computing services for the users. Tasks will be offloaded to UAV to perform. As we can see the variable $c_{z,k,u}^m$ in (1). It's the offloaded variable for task $s_{z,k,u}$. The TOPA will determine the value of the Offloading variable $s_{z,k,u}$ according to the task data index and the load capacity of the local server. In the following we will give the main steps of the algorithm.

Step 1: Place all generated task categories in their own set of tasks. Because the tasks generated in the region are not fixed in different time slices, we first judge whether the task set of each region is empty. If it is not empty, proceed to the second step.

Step 2: We will calculate the minimum allocated CPU cycles of each task in turn according to the task delay requirement and equation (11).

Algorithm 1 Tasks Offloading Policy Algorithm

Require:
$\mathcal{S}_k$: Set of tasks for all areas generated at the time slot k; $\mathcal{S}_{z,k}$: Set of tasks for area c_z generated at the time slot k; $\mathcal{M}$: Set of UAVs;$\mathcal{Z}_{requir}$: Set of areas requiring UAV-assisted;

Ensure: $\mathcal{Z}_{requir}$

while $\mathcal{S}_k$!=Empty **do**
  $\mathcal{S}_{z,k} \leftarrow \mathcal{S}_k.pop()$;
  while $\mathcal{S}_{z,k}! = NULL$ **do**
    for $s_{z,k,u} \in \mathcal{S}_{z,k}$ **do**
      Calculate $f^m_{z,k,u} = \frac{F_{z,k,u}}{T^m_{z,k,u} - \frac{D_{z,k,u}}{r^m_{z,k,u}}}$;
      if $f^m_{z,k,u} <= f$ **then**
        $c^m_{z,k,u} \longleftarrow 1, m \longleftarrow 0$;
        $f \longleftarrow f - f^m_{z,k,u}$;
      else
        $c^m_{z,k,u} \longleftarrow 1, m \neq 0$;
        $\mathcal{Z}_{requir}$.add(c_z)
      end if
    end for
  end while
end while

Step 3: The first step is to determine whether the local server has enough CPU cycles to allocate to the task based on the minimum CPU required by the task calculated in step 2. If the local server has more CPU cycles than the task requires, the task will be offloaded to the local server, otherwise the task will be unloaded to no one for execution.

B. Online UAVs Dispatching Base On the Shortest Distant

As time changes, the areas requiring UAV assistance will change. Therefore, it is an important issue to plan a scheduling scheme for UAVs to make the flying distance of UAVs shorter. In this paper, the problem we studied is to minimize the energy consumption of the UAV under the condition of satisfying the tasks requirement as far as possible. The energy consumption of UAV mainly comes from three aspects: flying, hovering and computing. Since the flying energy consumption of the UAV is greater than that of hovering, if the energy consumption of the UAVs are minimized without considering the computing, it is equivalent to minimizing the flying distance of the UAVs. Based on the above discussion, we propose the Online UAVs Dispatching Base On the Shortest Distant. In the following we will give the main steps of the algorithm.

Step 1: Firstly, a two-dimensional array is initialized to store the distance of the UAVs to each demand area, and then the areas that need UAVs assistance are obtained according to the Tasks Offloading Policy Algorithm.

Step 2: Calculate the distances of each UAVs to the requirement points, and store the distance into a two-dimensional array in turn. Then, the shortest path is obtained by iterating the two-dimensional array, and the state of the UAVs are updated.

Algorithm 2 Online UAVs Dispatching Base On the Shortest Distant

Require:
$\mathcal{M}$:Set of UAVs;$\mathcal{Z}_{requir}(Algorithm1)$; array1:Storing the distance of the UAVs to the areas;array2:Storing the status of the UAVs;
initial i=0,j=0,k=0,sum=0,array2:every values of the array2 are 0, and the length is $\mathcal{M}.size()$;

Ensure: Optimal scheduling strategy for the shortest flight distance of UAV

while $i < \mathcal{Z}_{requir}$.size() **do**
  while $j < \mathcal{M}$.size() **do**
    Calculate $d_{z,m,k} = \sqrt{(x_z - x_{m,k})^2 + (y_z - y_{m,k})^2}$;
    $array1[i][j] \longleftarrow d_{z,m,k}$;
    $i \longleftarrow i + 1$;
    $j \longleftarrow j + 1$;
  end while
end while
MINDISTANT(Array1, Array2, k,sum)
Get the dispatch strategy of the UAVs
Update $\mathcal{M}$

function MINDISTANT($Array1, Array2, k, sum$)
  if $i < \mathcal{Z}_{requir}$ **then**
    if $sum < min$ **then** min=sum;
    end if
    return min;
  end if
  while $k = 0$ to $\mathcal{M}.size()$ **do**
    if $array[k] == 1$ **then**
      Continue;
    else
      $array[k] == 1$
      MINDISTANT($Array1, Array2, i + 1, sum + array1[i][k]$)
      $array[k] == 0$
    end if
  end while
  return min;
end function

IV. SIMULATION AND RESULTS

In this section, we will present simulation results to evaluate the performance of our algorithm. We consider the square ares(600m×600m) with a UAV-assisted edge computing network with $N = 9$ edge servers, $M = 4$ UAVs and several UEs. In simulations, the channel bandwidth is $B =$ 1MHz. For each UAV, we set the the altitude as $H =$ 20m. The propulsion power and hovering power are respectively set as $P^f_b =$ 200W and $P^h_b =$ 100W. The transmission power is $P =$ 16dBm, and the computation capacity of server is $f = 10^9$ cycles/s. We set the channel power as $g_0 = -40$dB and $\sigma^2 = -160$dBm/Hz. Then we set $T =$ 1000ms and $\tau =$ 1ms.

In our simulation, we take the ratio of tasks completion and energy cost of the the system UAVs as performance indicators. To make a fair comparison, we introduce the following server placement scenarios.

Fixed deployment(Only_fixed): The edge server is configured within the zone and is used to compute user-generated tasks within the zone.

UAV dispatching(Only_UAV): The UAVs that are equipped with edge servers provide computing services for UEs, and when there is the task request in a certain area, the UAVs will is dispatched to provide services for it.

Hybrid dispatching(Hybrid): A hybrid solution is considered here, in which the UAVs assist local edge servers to provide services to the UEs. the task that the local edge server fails to complete will be offloaded to the UAVs for execution.

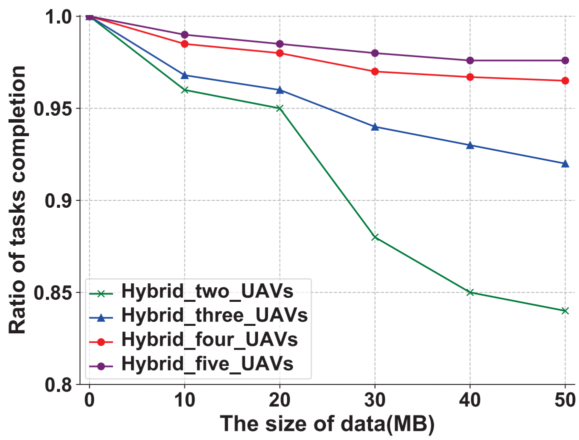

Fig. 2. Ratio of tasks completion

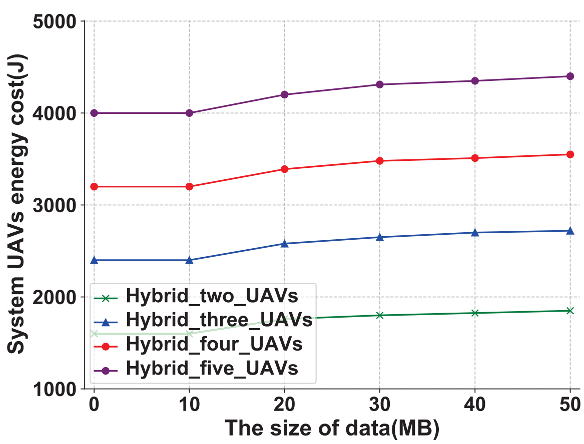

Fig. 3. System UAVs energy cost with different number of UAVs

Firstly, Fig. 2 and Fig. 3 show the performance of different numbers of UAVs in a mixed scheme Hybrid scheme in

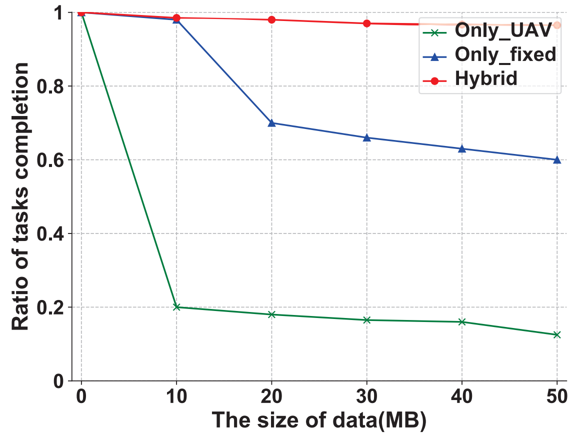

Fig. 4. Ratio of tasks completion

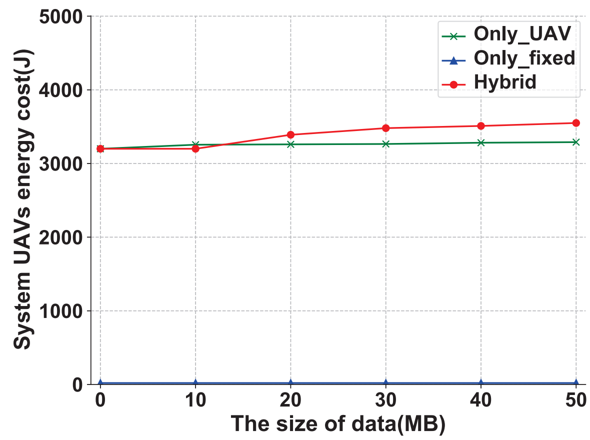

Fig. 5. System UAVs energy cost under different scenarios

different data volumes. Form Fig. 2, we can see that the completion rate of the task decreases with the increase of size of data. However, the completion rate of Hybrid_four_UAVs and Hybrid_five_UAVs is always above 95%. Fig. 3 shows the energy consumption performance of the Hybrid scheme under different size of data of the system UAVs. Since the the computational energy consumption of the system UAVs cannot be reduced, we only consider the energy consumption of the UAVs in flying and hovering. As expected, the energy consumption of the UAVs increases with the increase of the number of UAVs. From the above, we can know that Hybrid_four_UAVs has an excellent performance in our scenario. So in Fig. 4 and Fig. 5, Hybrid_four_UAV are selected as the Hybrid scheme.

Then, in Fig. 4 and Fig. 5 show the performance of Hybrid, Only_UAV, and Only_fixed. As can be seen from Fig. 4, the ratio of tasks completion of the Hybrid scheme does not fluctuate significantly with the increase of the size of data, and it always approaches 1. However, the Only_UAV, and Only_fixed performed poorly. the Only_UAV performed the worst with the ratio of tasks completion of less than 20%. In Fig. 5 shows the energy consumption of system UAVs. Since there is no UAV in the Only_fixed, the energy consumption is zero. Although the energy consumption of the Hybrid scheme is higher than the Only_UAV's, the difference is very small. Therefore, the comprehensive the ratio of tasks completion and energy consumption of the system UAVs, the Hybrid scheme we proposed is the most outstanding performance.

V. Conclusion

In this paper, we propose a hybrid scheme of the UAVs-assisted edge computing system. We study the problem of the ratio of tasks completion and the energy consumption of system UAVs. To solve this problem, we propose the TOPA and ODSH algorithms. In the simulation experiment, we compared the Hybrid scheme with the traditional Only_fixed and Only_UAV scheme. The simulation results shows that the ratio of tasks completion of the Hybrid scheme is 35% and 75% higher than the Only_fixed's and the Only_UAV's, respectively. In the future work, We can use artificial intelligence(AI) to predict the trajectory of the UAVs, so as to better dispatch the UAVs. And UAVs can also be used as the relays of tasks offloading and delivery.

References

[1] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu. Edge computing: Vision and challenges. *IEEE Internet of Things Journal*, 3(5):637–646, 2016.

[2] H. Li, G. Shou, Y. Hu, and Z. Guo. Mobile edge computing: Progress and challenges. In *2016 4th IEEE International Conference on Mobile Cloud Computing, Services, and Engineering (MobileCloud)*, pages 83–84, 2016.

[3] P. Hao, L. Hu, J. Jiang, J. Hu, and X. Che. Mobile edge provision with flexible deployment. *IEEE Transactions on Services Computing*, 12(5):750–761, 2019.

[4] J. Wang, K. Liu, and J. Pan. Online uav-mounted edge server dispatching for mobile-to-mobile edge computing. *IEEE Internet of Things Journal*, 7(2):1375–1386, 2020.

[5] F. Zhou, Y. Wu, H. Sun, and Z. Chu. Uav-enabled mobile edge computing: Offloading optimization and trajectory design. In *2018 IEEE International Conference on Communications (ICC)*, pages 1–6, 2018.

[6] S. Jeong, O. Simeone, and J. Kang. Mobile edge computing via a uav-mounted cloudlet: Optimization of bit allocation and path planning. *IEEE Transactions on Vehicular Technology*, 67(3):2049–2063, 2018.

[7] Y. Zeng, R. Zhang, and T. J. Lim. Wireless communications with unmanned aerial vehicles: opportunities and challenges. *IEEE Communications Magazine*, 54(5):36–42, 2016.

[8] R. Fan, J. Cui, S. Jin, K. Yang, and J. An. Optimal node placement and resource allocation for uav relaying network. *IEEE Communications Letters*, 22(4):808–811, 2018.

[9] S. Garg, A. Singh, S. Batra, N. Kumar, and L. T. Yang. Uav-empowered edge computing environment for cyber-threat detection in smart vehicles. *IEEE Network*, 32(3):42–51, 2018.

[10] M. A. Messous, H. Sedjelmaci, N. Houari, and S. M. Senouci. Computation offloading game for an uav network in mobile edge computing. In *2017 IEEE International Conference on Communications (ICC)*, pages 1–6, 2017.

[11] F. Tang, Z. M. Fadlullah, B. Mao, N. Kato, F. Ono, and R. Miura. On a novel adaptive uav-mounted cloudlet-aided recommendation system for lbsns. *IEEE Transactions on Emerging Topics in Computing*, 7(4):565–577, 2019.

[12] K. Zhang, X. Gui, D. Ren, and D. Li. Energy-latency tradeoff for computation offloading in uav-assisted multi-access edge computing system. *IEEE Internet of Things Journal*, pages 1–1, 2020.

[13] Kitae Kim and Choong Seon Hong. Optimal task-uav-edge matching for computation offloading in uav assisted mobile edge computing. In *2019 20th Asia-Pacific Network Operations and Management Symposium (APNOMS)*, pages 1–4, 2019.

[14] X. Diao, J. Zheng, Y. Wu, Y. Cai, and A. Anpalagan. Joint trajectory design, task data, and computing resource allocations for noma-based and uav-assisted mobile edge computing. *IEEE Access*, 7:117448–117459, 2019.

[15] D. Yang, Q. Wu, Y. Zeng, and R. Zhang. Energy tradeoff in ground-to-uav communication via trajectory design. *IEEE Transactions on Vehicular Technology*, 67(7):6721–6726, 2018.

[16] Y. Zeng, R. Zhang, and T. J. Lim. Throughput maximization for uav-enabled mobile relaying systems. *IEEE Transactions on Communications*, 64(12):4983–4996, 2016.

[17] X. Chen. Decentralized computation offloading game for mobile cloud computing. *IEEE Transactions on Parallel and Distributed Systems*, 26(4):974–983, 2015.

[18] Y. Mao, J. Zhang, and K. B. Letaief. Dynamic computation offloading for mobile-edge computing with energy harvesting devices. *IEEE Journal on Selected Areas in Communications*, 34(12):3590–3605, 2016.

[19] Z. Yang, C. Pan, K. Wang, and M. Shikh-Bahaei. Energy efficient resource allocation in uav-enabled mobile edge computing networks. *IEEE Transactions on Wireless Communications*, 18(9):4576–4589, 2019.

[20] Yuqing Chen and Zhaohui Zheng. Joint deployment and task computation of uavs in uav-assisted edge computing network. In *2020 21st Asia-Pacific Network Operations and Management Symposium (APNOMS)*, pages 413–416, 2020.

[21] Yan Kyaw Tun, Yu Min Park, Nguyen H. Tran, Walid Saad, Shashi Raj Pandey, and Choong Seon Hong. Energy-efficient resource management in uav-assisted mobile edge computing. *IEEE Communications Letters*, 25(1):249–253, 2021.

[22] Yi Liu, Shengli Xie, and Yan Zhang. Cooperative offloading and resource management for uav-enabled mobile edge computing in power iot system. *IEEE Transactions on Vehicular Technology*, 69(10):12229–12239, 2020.

[23] L. Wang, K. Wang, C. Pan, W. Xu, N. Aslam, and L. Hanzo. Multi-agent deep reinforcement learning-based trajectory planning for multi-uav assisted mobile edge computing. *IEEE Transactions on Cognitive Communications and Networking*, 7(1):73–84, 2021.

[24] A. Gao, Q. Wang, Y. Hu, and W. Duan. An offloading optimization scheme for multi-uav aided network in mobile computing. In *2020 International Wireless Communications and Mobile Computing (IWCMC)*, pages 1468–1473, 2020.

[25] J. Zhang, L. Zhou, Q. Tang, E. C. . Ngai, X. Hu, H. Zhao, and J. Wei. Stochastic computation offloading and trajectory scheduling for uav-assisted mobile edge computing. *IEEE Internet of Things Journal*, 6(2):3688–3699, 2019.

[26] H. Zhao, H. Wang, W. Wu, and J. Wei. Deployment algorithms for uav airborne networks toward on-demand coverage. *IEEE Journal on Selected Areas in Communications*, 36(9):2015–2031, 2018.

[27] H. Zhao, J. Wei, S. Huang, L. Zhou, and Q. Tang. Regular topology formation based on artificial forces for distributed mobile robotic networks. *IEEE Transactions on Mobile Computing*, 18(10):2415–2429, 2019.

[28] Q. Hu, Y. Cai, G. Yu, Z. Qin, M. Zhao, and G. Y. Li. Joint offloading and trajectory design for uav-enabled mobile edge computing systems. *IEEE Internet of Things Journal*, 6(2):1879–1892, 2019.

[29] Y. Qian, F. Wang, J. Li, L. Shi, K. Cai, and F. Shu. User association and path planning for uav-aided mobile edge computing with energy restriction. *IEEE Wireless Communications Letters*, 8(5):1312–1315, 2019.

[30] T. Zhang, Y. Xu, J. Loo, D. Yang, and L. Xiao. Joint computation and communication design for uav-assisted mobile edge computing in iot. *IEEE Transactions on Industrial Informatics*, 16(8):5505–5516, 2020.