Optimal Task-UAV-Edge Matching for Computation Offloading in UAV-Assisted Mobile Edge Computing

Kitae Kim  
Department of Computer Science and Engineering  
Kyung Hee University  
Yongin, Republic of Korea  
glideslope@khu.ac.kr

Choong Seon Hong  
Department of Computer Science and Engineering  
Kyung Hee University  
Yongin, Republic of Korea  
cshong@khu.ac.kr

Abstract—Unmanned Aerial Vehicle (UAV) is mobile and has the advantage of being equipped with cameras, sensors, computing resources and devices for communication. By combining the advantage of communication technology and UAV, rapid response can be made at disaster area and used as a mobile base station in place where traffic demand is high, such as concert venue and sports stadium. In addition, by using data that have been collected by mounted sensors and cameras, the UAV can provide data-based application. However, these services usually use big data processing and machine learning techniques which require high computing power and the UAV is not sufficiently capable of processing these applications due to lack of computing resources and battery limitation. To overcome this problem, the UAV can offload a task to near Mobile Edge Server that can provide computing resources. Mobile Edge Server can be cellular base station, Wi-Fi access point and so on. When tasks occur in a specific area, one or more UAV need to move the location to acquire data and process. If data processing is too heavy to process at local, the UAV can cooperate mobile edge server. In this situation, we can find out two problems. (i) When the tasks occur, which UAV can process occurred task. (ii) When UAV is assigned a task, then a mobile edge server can cooperate with UAV. In this paper, based on Hungarian algorithm which is one of matching algorithm, we propose optimal task-UAV-edge server matching algorithm which minimizes energy consumption and processing time.

Keywords—5th Generation Communication, Unmanned Aerial Vehicle, Mobile Edge Computing, Hungarian Algorithm

I. INTRODUCTION

Recently, the fifth-generation communication service has launched and various kind of mobile devices and Internet-of-Things sensors emerged which are produce a wide variety of data. Therefore, new applications have recently emerged, combining with the technologies such as big data processing and machine learning, VR(Virtual Reality), AR(Augmented Reality). However, processing these services on mobile device is not appropriate because mobile devices have lower computing power and battery.

To solve that problem, mobile edge computing has introduced which is network architecture concept that enables cloud computing capabilities and IT service environment at edge of the network such as base stations and Wi-Fi Access point[1]. However, when mobile device decide to offload their tasks to mobile edge server, the nearest mobile edge server is not always the optimal solution. In spite of the nearest distance, if there is congestion in communication environment and overhead in mobile edge server, It will not be able to meet the requirements of the task. Therefore, offloading proper mobile edge server consider communication and processing time and process requirements is very important.

Also, Unmanned Aerial Vehicles (UAV) have been attract attention in smart industry. Besides, In the smart city UAV can travel around city and perform given tasks such as observation, data acquisition, building map, disaster management, agriculture using on-board sensors[2]. But UAVs also has lower computing resources. Especially efficient use of battery is very important. In this situation, UAV can also utilize the resource of mobile edge server. UAV need move to location of task and select proper mobile edge server can cooperate with UAV to meet requirement of task while minimizing energy consumption.

In this paper, we propose optimal task-UAV-mobile edge server matching algorithm using Hungarian algorithm which consider energy consumption, processing and delay time in mobile edge server and the location of the UAVs, Tasks and mobile edge servers. To reduce energy consumption when UAVs move, we consider not only current location of UAVs but also return location of UAVs.

II. SYSTEM MODEL

Figure 1 shows considered system model. We consider four UAVs and four mobile edge servers and tasks are considered respectively. One ground station is considered which can give command to connected UAVs and these two are connected via
backhaul network. Tasks with different properties occur at random locations on two-dimensional coordinates. Task can be observation and data acquisition such as photo and video. In this situations, we aim at appropriate UAV-Task-Edge matching.

To achieve our goal, three important metrics are considered for optimal Task-UAV-edge server matching. The first one is Energy consumption. As mentioned in previous section, UAV has limited battery. The second and last is network and queuing delay in mobile edge server. In this section, we model the three metrics mentioned earlier and define the cost function to consider them all.

A. Network and Task Model

The UAV can be connected to one base station at a time through wireless channel. The uplink data rate takes into account to calculate the delay in the data transfer. Therefore, the data rate $\text{DataRate}_{\text{UAV-Edge}}$ between UAV and mobile edge server can be defined as follows:

$$\text{DataRate}_{\text{UAV-Edge}} = B \times \log_2(1 + SNR) \quad (1)$$

In equation 1, B represents the bandwidth between UAV and mobile edge server, and SNR represents signal to noise ratio that can be expressed below:

$$SNR = \frac{p \cdot c}{\sigma^2} \quad (2)$$

In here, $\sigma^2$ denotes noise power and p and c represents transmit power and channel gain respectively. In addition, the amount of data transfer required to perform task is defined S bits/s. Note that we did not consider downlink because the size of the result after processing task is relatively small.

Using equation 1 and 2, we can calculate the data transfer time for task processing when UAV and mobile edge server are matched as shown in below

$$T_{transfer} = \frac{S}{\text{DataRate}_{\text{UAV-Edge}}} \quad (3)$$

B. Energy Consumption

It is important to efficiently use the energy of UAV with a limited battery. When a task occurs, the UAV move to the task to obtain data for processing, and then move to connect to a matched mobile edge server. Also, energy for hovering should be considered while the UAV is transferring data to the Edge server. In [3] paper modeled UAV energy consumption while flying and hovering.

$$E_{flying} = \frac{p^{f \cdot h}}{\eta} \cdot d \quad (4)$$

$$E_{hovering} = \frac{p^{m \cdot h}}{\eta} \quad (5)$$

From equation (4) and (5) $p^{f \cdot h}$ and $p^{m \cdot h}$ denotes the minimum power for going forward and hovering respectively, d is distance that the UAV traveled, $\eta$ denotes power efficiency.

$$p^{f \cdot h} = (\nu' + v \sin \beta)T \quad (6)$$

$$d = d_{task} + d_{edge} + d_{return} \quad (7)$$

$$p^{m \cdot h}$$ can be calculated as equation (6) where $\nu$ is ground speed and $\beta$ is pitch angle[4]. $\nu'$ is required speed for thrust $T$ can be expressed as equation (8)

$$T = mg + f_d \quad (8)$$

m is mass of UAV and g is gravity acceleration. $f_d$ is drag force. In addition, we can get $\nu'$ by solving quadratic equation (9)

$$\nu' = \frac{2T}{qr^2 \pi r^2 (\nu \sin \beta)^2 + (\nu + \nu')^2} \quad (9)$$

Where q and r, $\gamma$ represents number and diameter of UAV propeller and density of air. Thus, we can consider heterogeneous UAV by using this energy consumption model.

C. Queuing and Processing Delay

When offload a task to mobile edge server, it is important to consider the number of jobs currently queued. Each mobile edge server has queue of size $Q_{que}$. This tasks arrive at mobile edge server following poisson process with rate $\lambda$. Each edge server can calculate expected standby time by equation (10)

$$T_{exp} = \sum_{q=0}^{Q_{current}} \frac{CPU_q}{E_{q, CPU}} \quad (10)$$

where $Q_{current}$ represents the last queue index number of the current mobile edge server and $CPU_q$,denotes required CPU cycle of q-th task in the queue. $E_{q, CPU}$ is allocated CPU cycle to process q-th task. Finally, we can easily derive total delay of task completion

$$T_{total} = T_{exp} + \frac{S}{E_{CPU}} \quad (11)$$

D. Cost Function and Other Considerations

Based on equation (3) (4) (5) (11), we can calculate total cost for each task-UAV-mobile edge server.

$$C_{total} = a \cdot E_{flying} + b \cdot E_{hovering} + c \cdot T_{transfer} + d \cdot T_{total} \quad (12)$$

Each variable in $C_{total}$ has weight value that can be adjusted according to properties of task.

$$a + b + c + d = 1 \quad (13)$$

Where weight variable a,b,c,d are in [0,1]
### TABLE I. Considered Variables

<table>
<thead>
<tr>
<th>Elements</th>
<th>Considerations</th>
<th>Related Variables</th>
</tr>
</thead>
<tbody>
<tr>
<td>UAV</td>
<td>Current Position</td>
<td>$p(x, y)$</td>
</tr>
<tr>
<td></td>
<td>Return Position</td>
<td>$p_r(x, y)$</td>
</tr>
<tr>
<td></td>
<td>Energy Consumption</td>
<td>$E_f(x) + E_{hoven}$</td>
</tr>
<tr>
<td>Task</td>
<td>Size of Task</td>
<td>$S$ bits</td>
</tr>
<tr>
<td></td>
<td>Needed CPU cycle</td>
<td>$CPU_u$</td>
</tr>
<tr>
<td>Edge Server</td>
<td>Queue Length</td>
<td>$Q_{sw} + Q_{current}$</td>
</tr>
<tr>
<td></td>
<td>Queue Delay</td>
<td>$T_{exp}$</td>
</tr>
<tr>
<td></td>
<td>Processing Delay</td>
<td>$T_{return} + \frac{S}{E_{cpu}}$</td>
</tr>
</tbody>
</table>

Note that when match task-UAV-mobile edge server, we should consider not only current location of UAV but also return position for next task matching from energy efficiency aspect.

### III. PROPOSED ALGORITHM

In this section, we propose matching algorithm based optimal task-UAV-mobile edge server while minimizing energy consumption and task completion time. We can get cost for every matching through equation (12) from the previous section. In this point, we can express relationship between task-UAV-mobile edge server using bipartite graph which is consist of two disjoint and independent sets $U, V$ to apply Hungarian Algorithm. Then we can represent all the tasks, UAVs and mobile edge servers as vertices of the graph.

#### A. Task-Mobile Edge Server Matching

First of all, we match task-edge. Because the UAV consider both the location of task and mobile edge server. As we mentioned earlier, we can construct graph with different two independent set of two vertices. In the graph, the weight of edges between tasks and mobile edge servers can be cost when task-mobile edge server are matched. For the cost calculation, we first measure distance between task and mobile edge server.

$$C_{task} - m es = T_{total} + \text{distance task} - m es \quad (14)$$

#### B. UAV-Mobile Edge Server with Task

After matching task and mobile edge server, we can create a graph for matching UAVs with the results. Figure 3 represents graph for UAV-mobile edge server with task matching.

![Figure 2. UAV-Mobile Edge Server Matching](image)

Vertices of graph on figure 2 represent weight when the UAV and mobile edge server are matched. For weight calculating, we consider Current position of UAVs and their original position that they need to go back for energy consumption of the UAV.

$$C_{uav-m es} = E_{f(x)} + E_{hoven} + T_{transfer} \quad (15)$$

The objective this two stages of matching can be expressed below:

Minimize $\sum_{i=1}^{m} \sum_{j=1}^{n} \sum_{k=1}^{o} C_{total} \ u_{yk}$

Subject to $\sum_{j=1}^{n} u_{yk} = 1, \quad i = 1, 2, \ldots n$

$\sum_{i=1}^{m} u_{yk} = 1, \quad j = 1, 2, \ldots m$

$\sum_{i=1}^{o} u_{yk} = 1, \quad k = 1, 2, \ldots o$

$u_{yk} \in \{0, 1\}$

$m, n, o$ denotes the number of tasks, UAVs and mobile edge servers respectively. $C_{total}$ is the total cost when task $i$, UAV $j$, mobile edge server $k$ are matched. Lastly, $u_{yk}$ is binary variable which is represent matching of task-UAV-mobile edge server. Therefore, we propose optimal matching algorithm in following:

Algorithm 1. Optimal task-UAV-mobile edge matching

1: if offloading_request = true
2: Construct bipartite graph task-mobile edge
3: for each vertexes between task i and MES k
4: $ast_k = \text{calculate } C_{task} - m es$
5: task_edge[i][k] = $ast_k$
6: end for
7: Hungarian_Algorithm(task_edge[i][k])
8: Subtract the minimum value from each row and column from task_edge[i][k]
9: Erase rows and columns contains 0 with a minimum lines
10: while(lines=k)
11: Subtract the minimum value of the remaining values from the remaining values
12: Find independent 0 that is only one 0 in one low and column
13: end while
14: return task-mobile edge_matching_result
15: Construct bipartite graph uav-mobile edge
16: for each vertexes between UAV j and MES k
17: $ast_k = \text{calculate } C_{uav-m es}$
18: uav_edge[j][k] = $ast_k$
19: end for
20: UAV-mes_matching_result = Hungarian_Algorithm(uav_edge[j][k])
21: for each UAV
22: send_command(UAVj)
23: end for
task-mobile edge graph for Hungarian method. Then calculate each weight for matching. From this each weight between task and mobile edge server can expressed \( i \times j \) matrix. Then from line 7 to 12 is Hungarian method process.

IV. SIMULATION

A. Simulation Environment

<table>
<thead>
<tr>
<th>Variables</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>( B )</td>
<td>[1,10]Mhz</td>
</tr>
<tr>
<td>( \rho )</td>
<td>[0.5,1]W</td>
</tr>
<tr>
<td>( c )</td>
<td>127+30 \times \log d</td>
</tr>
</tbody>
</table>
| \( \sigma^2 \) | \( 2 \times 10^{-11}W \)
| \( q \)    | 4[6] |
| \( \gamma \) | 0.254m[6] |
| \( f_d \)  | 9.9698N[6] |
| \( \eta \) | 70%/[6] |
| \( \gamma \) | 12.25kg/m\(^3\)[6] |
| \( g \)    | 9.8m/s\(^2\) |
| \( a \)    | 0.4 |
| \( b \)    | 0.4 |
| \( c \)    | 0.1 |
| \( d \)    | 0.1 |
| \( \lambda \) | 0.4 |

For the simulation, we construct simulation environment using python programming. Table II shows simulation variables.

Figure 3 shows measurement of processing time when task-UAV-mobile edge server are matched using distance based algorithm compared to our proposed algorithm. At some point, processing time of distance based matching algorithm are lower than proposed algorithm. Because we more consider energy consumption by setting higher value than processing time related value. At that time processing time is influenced by random state of queue at mobile edge server and CPU allocation. Despite of that point, overall processing time of proposed method is lower than distance based method. Figure 47 shows total cost of proposed algorithm compared to distance based algorithm. We can see the proposed algorithm shows better performance in terms of total cost that we defined in section III clearly.

V. CONCLUSION

In this paper, we propose optimal task-UAV-mobile edge server method using Hungarian algorithm while minimizes energy and processing time. For this, we modeled communication model and queuing and processing delay to define cost function with weight. Weight adjustment allows flexibility in handling in dynamic situations such as different Quality of Service (QoS) requirement from users. For simulation we consider each 4 tasks, UAVs, mobile edge servers. We compared our proposed method to distance based matching method and see the proposed one shows better performance. For the further research we will consider situation that assigns the sequential task for UAV and find optimal position for task transfer to mobile edge server when UAV matches with mobile edge server.

REFERENCES