

Machine Learning Based Edge-Assisted UAV Computation Offloading for Data Analyzing

Kitae Kim, Yu Min Park, Choong Seon Hong
Department of Computer Science and Engineering
Kyung Hee University
Yongin, Republic of Korea
{glideslope, yumin0906, cshong}@khu.ac.kr

Abstract—Recently, Combining communication technology with Unmanned Aerial Vehicle (UAV) have been regarding as one of the promising techniques in the future network. Various services can be served through UAV, such as information collection in disaster area or traffic monitoring in smart city. However, on-board computation resource and the battery lifetime of UAV are limited. For that reason, there is a limit to perform analysis, such as, machine learning using collected data. In this situation, UAV can get help from an adjacent Mobile Edge Computing (MEC) server that has computing resources. However, the closest MEC server does not always guarantee optimal computing performance and communication efficiency. In this paper, to solve this problem, we propose the overall system of machine learning based UAV to MEC computation offloading for data analyzing. First, we define the problems into two. Firstly, we define two problems 1) matching UAV and task cluster in consideration of energy efficiency. 2) Finding optimal MEC server to offload the task to minimize the total processing time and energy consumption. Then, we apply a machine-learning algorithm to solve the two problems. In the simulation section, we analyze the proposed method and greedy method in terms of total energy consumption and total processing time. Simulation results demonstrate that our proposed mechanism is efficient in total energy consumption of UAV and total processing time of tasks.

Keywords—Mobile Edge Computing, Reinforcement Learning, Q-Learning

I. INTRODUCTION

Recently, the use of Unmanned Aerial Vehicles (UAVs) in various fields is attracting attention. In particular, it is expected to be able to provide various services by converging with communication technology such as 5G[1][2]. UAVs can offer a variety of services that were not previously available and can also be used on the battlefield. There are various types of UAVs depending on the purpose of use, the drone is one of the most widely known UAVs. These small UAVs are highly available, but they are constrained on-board resources such as battery and computing resources. Therefore, the problem of efficiently using the limited resources of UAV is important. UAV can perform disaster area monitoring, Internet of Things (IoT) sensor data collection, and surveillance, even package delivery. The data collected in this process must be analyzed somewhere to be utilized. Usually, data collected by UAVs are photographs or videos, and these data can be analyzed in the machine-learning technique. Analyzing the collected in a small UAV is not possible due to resource limitations of UAV. Therefore, UAVs transfer the collected data to the cloud. To provide faster service in these situations, UAVs can transfer data to nearby Mobile Edge Computing (MEC) servers which are computing resources near users such as base station or Access Point (AP) rather

than to the cloud. In this scenario, the closest MEC server does not always guarantee optimal performance when transferring data. Therefore, it is important to select the optimal MEC server to meet the requirements of the task. We already proposed the above mentioned problem in [3]. However, in previous work, we only consider when the number of UAVs and the number of tasks must be always the same because we apply the Hungarian algorithm that is one of the bipartite matching algorithm. Thus, it is not applicable if there are more tasks than the number of UAVs. In this paper, to overcome mentioned problem, we propose optimal offloading mechanism using K-means clustering algorithm and reinforcement learning while minimizing energy efficiency.

The rest of this paper is organized as follows. In section II, we introduce our overall system model consisting of a task model, communication model, energy consumption model, and processing delay model. In section III, we briefly review Q-Learning and present an optimal offloading mechanism using reinforcement learning. In the section IV, we verify the performance by comparing the proposed mechanism with the existing method. Finally, we conclude this paper in section V.

II. SYSTEM MODEL

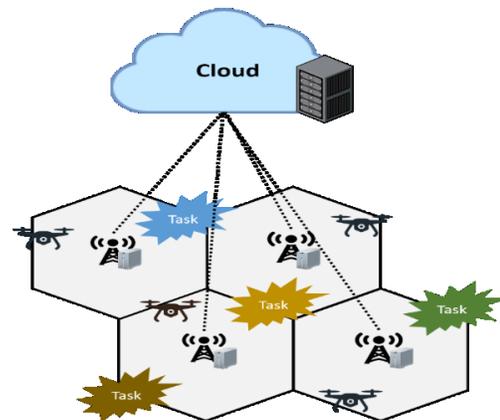


Figure 1. System Model

Fig1 shows overall system model of this paper. First we consider one cloud server that is connected MEC servers. Also we consider i number of UAVs $U = \{u_1, u_2, \dots, u_i\}$ and j number of MEC node $M = \{m_1, m_2, \dots, m_j\}$ that can be associated with UAV. We can define each location of UAV and MEC server in the 2-Dimensional coordinate system. Finally, we consider k number of tasks $T = \{t_1, t_2, \dots, t_k\}$ that occur at certain area and we express the

point of the task t_j in $P_j = \{x_j, y_j\}$ also on the 2-Dimension coordinate system.

A. Task Modeling

We define a task model in two variables. First, the size of task t_j as S_j and each task has different computing resource requirements. We only consider the CPU cycles required because it is related to processing time, which affects total processing time. We can describe CPU_j as a required CPU cycle for task j .

B. Network Model

For data analysis, each UAV is connected to one MEC server to transfer data. We only take into account the uplink data rate because the downlink data transfer is negligible. Thus, we can apply Shannon capacity for the data transfer rate between UAV and MEC server as follows. Therefore, data transfer rate (bits/sec) of task t_j between UAV and MEC server.

$$C_j = B \times \log_2(1 + SNR), \quad (1)$$

From equation (1), to calculate Signal to Noise Ratio (SNR) we consider Air-To-Ground Path Loss Model in urban environment as follows [4].

$$PL(\text{db}) = 20 \log\left(\frac{4\pi f_c d}{c}\right) + P(\text{LoS})\eta_{\text{LoS}} + P(\text{NLoS})\eta_{\text{NLoS}} \quad (2)$$

where d denotes the distance between UAV and base station, $P(\text{LoS})$ and $P(\text{NLoS})$ represents Line-of-Sight (LOS) Probability and non-Line-of-Sight(NLoS) probability, respectively. In addition, η_{LoS} and η_{NLoS} denote additional losses. Then we calculate transmission power P as follows :

$$P = P_{\text{UAV-MEC}} + G_t - PL, \quad (3)$$

where $P_{\text{UAV-MEC}}$ is transmitting power, G_t , represents antenna gain of transmitter losses. Thus, SNR for uplink can be written as follows:

$$SNR = \frac{P \cdot c}{\sigma^2}, \quad (4)$$

where σ^2 denotes noise power, p and c are transmit power and channel gain. Finally, we can calculate data transfer time between UAV and MEC server as follows:

$$T_{\text{transfer}} = \frac{S}{\text{DataRate}_{\text{UAV-Edge}}}, \quad (5)$$

C. Energy Consumption Model

We also consider energy consumption. To apply various type of UAV, we consider detailed features of UAV for energy consumption model.

The energy consumption of the UAV is closely related to the distance the UAV has moved. In [5] paper, modeled

UAV energy consumption in two cases (in joule/m) flying E_{flying}

$$E_{\text{flying}} = \frac{p_{\text{flying}}^{\text{min}}}{\eta} \cdot d, \quad (6)$$

where p_f^{min} represents the minimum power for going forward, d is distance that the UAV moved, η is power efficiency. p_f^{min} can be calculated in equation (8) below:

$$p_f^{\text{min}} = (v' + v \sin \beta)T \quad (8)$$

In equation (8) v and β are ground speed and pitch angle [7] of UAV propeller respectively. Lastly, T is thrust of UAV which has required speed v' :

$$T = mg + f_d, \quad (9)$$

where m denotes mass of UAV and g is gravity acceleration. f_d is drag force.

Also, we can get v' solving quadratic equation below:

$$v' = \frac{2T}{qr^2\pi\gamma\sqrt{(v \cos \beta)^2 + (v \sin \beta + v')^2}}, \quad (10)$$

where q, r , γ represents number and diameter of UAV propeller and density of air, respectively

For UAV current energy level, we consider actual commercial drones in current which is 6000mAh and they are operated under 7.4 voltage, then we convert this to Joule.

$$\text{Joule} = \text{mAh} \times \text{voltage} \times 3.6, \quad (11)$$

Finally, we can define the total distance traveled of UAV can be divided into two: (i) from initial location to task, (ii) distance should be traveled to return to the final location of UAV.

$$d = d_{\text{task}} + d_{\text{cluster}} + d_{\text{return}}, \quad (12)$$

We will explain about variable d_{cluster} in section III.

D. Queuing and Processing Delay

In order to consider the total processing time of the task in the MEC server, we calculate the total processing of the task using the estimated processing time of the tasks currently waiting in the queue. We assume that task arrival follows a Poisson distribution with λ .

$$T_{\text{exp}} = \sum_{q=0}^{Q_{\text{last}}} \frac{CPU_q}{E_{q,CPU}}, \quad (12)$$

where Q_{last} denotes current length of queue in the MEC server, and the CPU_q is needed CPU cycle to process the task. Finally, $E_{q,CPU}$ represents the allocated CPU cycles for the Task.

$$T_{\text{total}} = T_{\text{exp}} + \frac{CPU_q}{E_{cpu}} + T_{\text{transfer}}, \quad (13)$$

Thus, we can calculate the total performance time from the sum of the estimated queue delay and the predicted process time of the current task.

TABLE I. CONSIDERED VARIABLES

Elements	Description	Variables
UAV	Current Position	$P_{UAV}(x, y)$
	Last Position	$P_L(x, y)$
	Energy Consumption	$E_{flying}, E_{hovering}$
	Current Energy Level	$E_{current}$
Task	Size of Task	$S_j(\text{bits})$
	Position	$P_j(x_j, y_j)$
	Needed CPU cycle	CPU_j
	Deadline	DL_j
Edge Server	Queue Length	$Q_{size}, Q_{current}$
	Queue Delay	T_{exp}
	Processing Delay	$T_{total}, \frac{CPU_q}{E_{cpu}}$

III. PROPOSED METHOD

In this section, we introduce the method of matching UAVs and tasks based on energy-efficient reinforcement learning, and the method of optimal MEC server selection problem, which can perform all the tasks most quickly after UAV-task matching.

A. Machine Learning UAV Placement on Task

We consider two situations in task and UAV matching as follows:

- 1. The number of tasks is greater than the number of UAVs**
 - In this case, one UAV can perform several tasks according to the matching result.
- 2. The number of tasks and the number of UAVs are the same**
 - We can match UAVs and task clusters on 1to1 matching.

To cover those situations, we use the K-Means clustering algorithm to cluster the tasks consider their distance. For applying, we set the K to the number of UAVs currently available.

This allows 1 to 1 matching between UAVs and clusters. Then we can calculate the total distance should be traveled in the cluster to visit all the task using the Traveling Salesman Problem. As a result, of the Traveling Salesman algorithm, we can have a primal node to visit first. To apply Q-Learning model of this problem, we defined state and action as following table III.

TABLE II. Q-LEARNING MODEL FOR MATCHING

Description	Variables
State	Current UAV Position $P_{UAV}(x, y)$ First visit task in Clusters $P_j = \{x_c, y_c\}$ Last Position of UAV $P_L = \{x_l, y_l\}$ Current Energy Level $E_{current}$
Action	Place the UAVs on the tasks
Reward	Total Distance need to be traveled = $\frac{1000}{(d_{task} + d_{cluster} + d_{return})}$

In Q-Learning, the most important factor in achieving a goal is the reward function. Calculate the reward function using the variables that must be maximized to achieve the

goal. At this stage, the energy required for the movement was considered. Since the energy efficiency changes according to the moving distance, the reward function can be represented by using the total moving distance that the UAV should move. The Agent learns an optimal policy to maximize the reward function. Thus, the reward function is the total sum of inverses of each travel distance. Thus, agent gets bigger reward as travel distance gets shorter. We multiply 1000 on the reward function. If not, the reward function is less than 1 and there is no big difference between reward values.

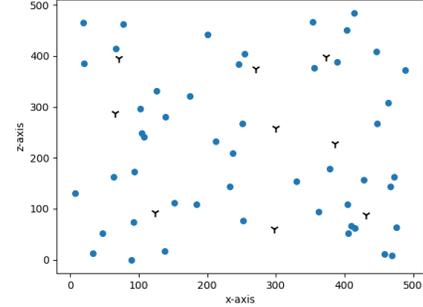


Figure 3. Before Clustering

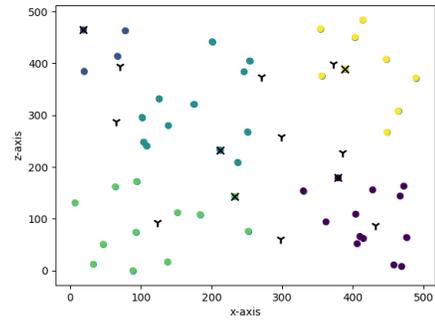


Figure 4. After Clustering

Fig. 3 and Fig. 4 shows the example of K-Means Clustering with 50 tasks in 4 cluster and, 9 edge server. On fig. 5, Y denotes edge server, and X denotes UAV. Each time the UAV moves to the next task, connectable depending on the coverage of AP or base station.

B. Optimal MEC server selection for collaboration

To find the MEC server that can process the task fastest, we consider the queueing delay and the processing delay in MEC server. We also take into account the amount of time the UAV sends data to the MEC server $T_{transfer}$. This selection process must be performed each time the UAV arrives at a task. This problem also solved in Q-Learning. Table IV.

TABLE III. Q-LEARNING MODEL FOR MEC SERVER SELECTION

Description	Variables
State	Requirement of Task Network Status of MEC Server Queue Status of MEC Server
Action	Select the MEC server
Reward	Transfer Delay, Expected Delay $\frac{1000}{T_{exp} + \frac{CPU_q}{E_{cpu}}} + T_{transfer}$

A. Simulation Environment

We simulate using python programming and Keras for Q-Learning process. Parameter values for simulation are defined following table V.

TABLE IV. SIMULATION PARAMETERS

Parameters	Value
B	10Mhz
\hat{G}_t	8dB
c	$127+30 \times \log d$
σ^2	$2 \times 10^{-13}W$
$E_{current}$	[50000, 159840]
q	4[9]
γ	0.254m[6]
f_d	9.9698N[6]
η	70%[6]
γ	1.225kg/m ³ [6]
g	9.8m/s ²
$\alpha(\text{Learning rate})$	10^{-2}
Discount Factor γ	0.95

B. Simulation Results

In simulation results, we can see that our proposed mechanism shows better performance in two cases. Fig. 5 represents total energy consumption. It is similar when there are few tasks, but as the number of tasks increases, the energy consumption gap increases. Fig. 6 describes the total processing time comparison of the two methods. It does not make much difference, but we can see that the proposed method performs tasks faster.

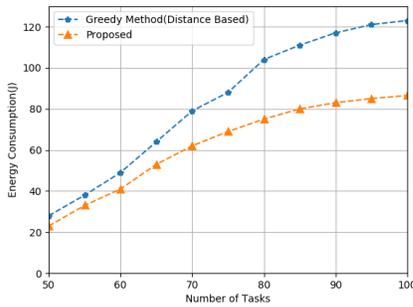


Figure 5. Total Energy Consumption (Proposed method versus Greedy Method)

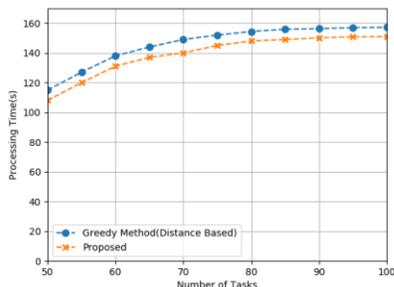


Figure 6. Total Processing Time (Proposed method versus Greedy Method)

In this paper, we propose a reinforcement learning-based edge-assisted UAV computation offloading method that considers energy efficiency and task processing time. In order to consider energy efficiency, a distance-based energy consumption model is used and the last position of the UAV is considered. Comparing with the greedy method, the proposed method shows better performance. However, two Q-learning models were used to solve the problem. Therefore, for the future work, we will consider the way to solve both problems in one Q-learning model. In addition, we plan to specify UAV and ground communication model and communication model between UAV and MEC server.

ACKNOWLEDGEMENT

This work was partially supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No.2019-0-01287, Evolvable Deep Learning Model Generation Platform for Edge Computing), and by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIT) (No.2015-0-00567, Development of Access Technology Agnostic Next-Generation Networking Technology for Wired-Wireless Converged Networks) *Dr. CS Hong is the corresponding author

REFERENCES

- [1] Long Zhang, Hui Zhao, Shuai Hou, Zhen Zhao, Haitao Xu, Xiaobo Wu, Qiwu wu and Rongui Zhang, "A survey on 5G Millimeter Wave Communications for UAV-Assisted Wireless Networks," IEEE Access, vol. 7, pp117460-117504, July. 2019
- [2] N. Hosseini, H. Jamal, J. Haque, T. Magesacher and D. W. Matolak, "UAV Command and Control, Navigation and Surveillance: A Review of Potential 5G and Satellite Systems," 2019 IEEE Aerospace Conference, Big Sky, MT, USA, March. 2019, pp. 1-10.
- [3] Kitae Kim and Choong Seon Hong, "Optimal Task-UAV-Edge Matching for Computation Offloading in UAV-Assisted Mobile Edge Computing," The 20th Asia-Pacific Network Operations and Management Symposium, Matsue, Japan, September. 2019.
- [4] S. ur Rahman, G. Kim, Y. Cho and A. Khan, "Positioning of UAVs for throughput maximization in software-defined disaster area UAV communication networks," in *Journal of Communications and Networks*, vol. 20, no. 5, pp. 452-463, Oct. 2018.
- [5] Joshua K. Stolaroff, Constantine Samaras, Emma R. O'Neill, Alia Lubers, Alexandra S. Mitchell & Daniel Ceperley, "Energy use and life cycle greenhouse gas emissions of drones for commercial package delivery", *Nature Communications*9, article number 409, 13 Feburary. 2018.
- [6] Momena Monwar, Omid Semiari, Walid Saad, "Optimized Path Planning for Inspection by Unmanned Aerial Vehicles Swarm with Energy Constraints". arXiv1808.06018, Aug. 2018.