RL-Based User Association and Resource Allocation for Multi-UAV enabled MEC

Liang Wang*, Peiqiu Huang†, Kezhi Wang*, Guopeng Zhang‡, Lei Zhang§, Nauman Aslam* and Kun Yang¶

*Department of Computer & Information Sciences, Northumbria University, Newcastle upon Tyne, UK
†School of Informacion Science & Engineering, Central South University, China
‡School of Computer Science & Technology, China University of Mining and Technology, China
§School of Engineering, University of Glasgow, UK
¶University of Electronic Science and Techology of China, Zhongshan Institute, China
¶School of Computer Science & Electronic Engineering, University of Essex, UK

Abstract—In this paper, multi-unmanned aerial vehicle (UAV) enabled mobile edge computing (MEC), i.e., UAVE is studied, where several UAVs are deployed as flying MEC platform to provide computing resource to ground user equipments (UEs). Compared to the traditional fixed location MEC, UAV enabled MEC (i.e., UAVE) is particular useful in case of temporary events, emergency situations and on-demand services, due to its high flexibility, low cost and easy deployment features. However, operation of UAVE faces several challenges, two of which are how to achieve both 1) the association between multiple UEs and UAVs and 2) the resource allocation from UAVs to UEs, while minimizing the energy consumption for all the UEs. To address this, we formulate the above problem into a mixed integer nonlinear programming (MINLP), which is difficult to be solved in general, especially in the large-scale scenario. We then propose a Reinforcement Learning (RL)-based user Association and resource Allocation (RLAA) algorithm to tackle this problem efficiently and effectively. Numerical results show that the proposed RLAA can achieve the optimal performance with comparison to the exhaustive search in small scale, and have considerable performance gain over other typical algorithms in large-scale cases.

Index Terms—Reinforcement Learning, Mobile Edge Computing, Unmanned Aerial Vehicle, User Association, Resource Allocation

I. INTRODUCTION

Nowadays, user equipments (UEs) such as smart phones, tablets, wearable devices and other Internet of smart things are becoming increasingly popular and bringing huge convenience to our daily life. Moreover, many emerging mobile applications (e.g., augmented reality, smart navigation and interactive service) are receiving more and more attention but most of those applications are resource intensive, which makes the UEs very difficult to execute them, due to limited battery and computation resource (e.g. CPU, storage or memory) in UEs.

Fortunately, mobile edge computing (MEC) has recently been proposed as a means to enable UEs with intensive computational tasks to offload them to the edge cloud, which can not only prolong the battery life of UEs, but also increase UEs' computational capacity. Offloading decision making and resource allocation have been studied in [1], [2], while MEC with Cloud Radio Access Network (C-RAN) has been investigated in [3], [4], [5]. The above works either consider there is only one MEC (e.g., [1], [7]), or consider the MECs have fixed location (e.g., [8], [3]), which may not be practical in some scenarios. For instance, the single MEC is normally resource-limited and may not be able to meet the requirement of all the UEs at the same time. Also, MEC with fixed location lacks flexibility and may not be suitable to the cases where the number and the requirement of UEs keep changing.

Unmanned aerial vehicle (UAV), due to the features of low cost, high flexibility and easy to deployment, have recently attracted much attention in wireless communication, e.g., serving as base station [9] or mobile relays [10]. UAV enabled MEC (e.g., [6]) have been proposed by integrating MEC server to UAVs (i.e., UAVE), to provide computing resource to ground UEs. Compared with the traditional fixed location MEC, UAVE is of particular interest to the scenario such as 1) temporary events (i.e., in case of a large number of people gathering in the ground celebrating a big event or watching football match); 2) emergency situations (i.e., in case of earthquake and the infrastructure may be destroyed or temporary unavailable) or other on-demand services. However, the operation of UAVE faces many challenges, two of which are how to achieve 1) the association between multiple UEs and UAVs and 2) the resource allocation from the UAVs to the UEs, while meeting the quality of service (QoS) and minimizing the whole energy consumption for all the UEs.

To address these challenges, we formulate above problem into a mixed integer non-linear programming (MINLP), which is very difficult to be addressed in general, especially in the large-scale scenario (e.g., when there is a large number of UEs in the ground waiting to be served). We then propose a Reinforcement Learning-based user Association and resource Allocation (RLAA) algorithm to deal with this problem efficiently and effectively. Numerical results show that the proposed RLAA can achieve the optimal performance compared to the exhaustive search in small scale, and have considerable performance gain over other typical algorithms in large-scale scenarios.

The rest of the paper is organized as follows. We show the system model and the optimization problem in Section II. Then, our proposed RLAA algorithm is introduced in Section
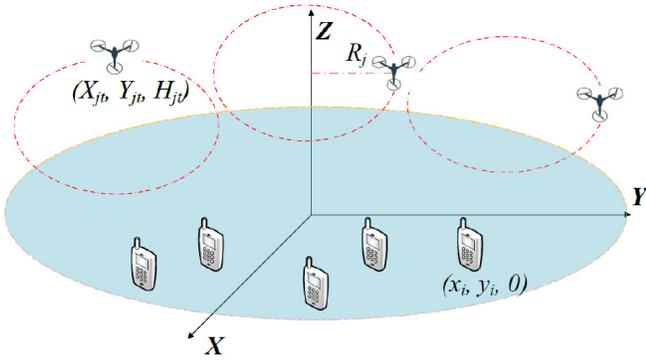
Fig. 1. A Multi-UAV enabled MEC system

III. The simulation result is given in Section IV, followed by the conclusion remarks in Section V.

II. SYSTEM MODEL

As shown in Fig. 1, we consider there are $i \in \mathcal{N} = \{1, 2, ..., N\}$ UEs, each of which has a computation-intensive task to be executed. Also, we consider there are $j \in \mathcal{M} = \{1, 2, ..., M\}$ UAVs deployed as the MEC platform, flying in a circle with radius R_j. Define a new vector $j \in \mathcal{M}' = \{0, \mathcal{M}\}$ to denote the possible place where the tasks from ground UEs can be executed at, in which $j = 0$ denotes that UE conducts task itself without offloading. Similar to [6], we assume that the j-th UAV's flight period can be discretized into $t \in \mathcal{T}_j = \{1, 2, ..., T_j\}$ time slots. Define a new vector $t \in \mathcal{T}'_j = \{0, \mathcal{T}_j\}$ to denote the possible time slots when the tasks from ground UEs can be executed at, in which $t = 0$ denotes that UE conducts task itself. Also we assume that the UAV's location change within each time slot can be ignored, compared to the distances from the UAV to all UEs.

Denote the coordinate of the j-th UAV at t-th time slot as $[X_{jt}, Y_{jt}, H_{jt}]$ and the coordinate of the i-th UE as $[x_i, y_i, 0]$.

Similar to [3], assume i-th UE has a computational intensive task I_i to be executed as

$$I_i = (D_i, F_i), \ \forall i \in \mathcal{N} \tag{1}$$

where F_i denotes the total number of CPU cycles required to complete this task and D_i denotes the amount of data needed to be transmitted to UAV if deciding to offload, in which D_i and F_i can be obtained by using the approaches provided in [11]. Assume that each UE can decide either to execute the task locally or choose to offload to one of the UAVs in one time slot and also assume that the task can be completed in this time slot. Similar to [1], we do not consider the time for returning the results back to UE from UAV. Thus, one can have

$$C1 : a_{ijt} = \{0, 1\}, \forall i \in \mathcal{N}, j \in \mathcal{M}', t \in \mathcal{T}'_j \tag{2}$$

where $a_{ijt} = 1$, $j \neq 0$, $t \neq 0$ denotes that the i-th UE choose the j-th UAV in the t-th time slot to offload, while $a_{ijt} = 1$, $j = 0$, $t = 0$ denotes that i-th UE execute the task itself and otherwise, $a_{ijt} = 0$. Note that $t = 0$, if and only if $j = 0$.

Also, assume that the j-th ($j \in \mathcal{M}$) UAV can serve more than one UE in each time slot and this task has to be completed either via offloading or local execution. Therefore, one can have

$$C2 : \sum_{j=0}^{M} \sum_{t=0}^{T_j} a_{ijt} = 1, \ \forall i \in \mathcal{N} \tag{3}$$

A. Task Offloading

In offloading scenario, we assume the horizontal distance between i-th UE and the j-th UAV in t-th time slot as

$$R_{ijt} = \sqrt{(X_{jt} - x_i)^2 + (Y_{jt} - y_i)^2} \tag{4}$$

Then, the offloading data rate can be given by

$$r_{ijt} = B \log_2(1 + \frac{\alpha P_i^{Tr}}{H_{jt}^2 + R_{ijt}^2}), \forall i \in \mathcal{N}, j \in \mathcal{M}, t \in \mathcal{T}_j \tag{5}$$

where B is denoted as the channel bandwidth, P_i^{Tr} as the transmission power of the i-th UE, $\alpha = \frac{g_0 G_0}{\sigma^2}$, $G_0 \approx 2.2846$, g_0 as the channel power gain at the reference distance $1 \ m$ and σ^2 as the noise power [12].

Also, one can see that the time to offload the data from i-th UE to the j-th UAV in t-th time interval can be given as

$$T_{ijt}^{Tr} = \frac{D_i}{r_{ijt}}, \ \forall i \in \mathcal{N}, j \in \mathcal{M}, t \in \mathcal{T}_j \tag{6}$$

Also, the time to execute the task can be expressed as

$$T_{ijt}^C = \frac{F_i}{f_{ijt}}, \ \forall i \in \mathcal{N}, j \in \mathcal{M}, t \in \mathcal{T}_j \tag{7}$$

where f_{ijt} is the computation resource that the jth UAV could provide to the i-th UE. Then, we can have the total time consumption as

$$T_{ijt} = T_{ijt}^{Tr} + T_{ijt}^C, \ \forall i \in \mathcal{N}, j \in \mathcal{M}, t \in \mathcal{T}_j \tag{8}$$

Moreover, the total energy consumption of the i-th UE to the j-th UAV in t-th time slot can be given as

$$E_{ijt}^{Tr} = P_i^{Tr} T_{ijt}^{Tr}, \ \forall i \in \mathcal{N}, j \in \mathcal{M}, t \in \mathcal{T}_j \tag{9}$$

Similar to [1], we assume each UAV in every time slot can only accept limited amount of offloaded task. Then, one has

$$C3 : \sum_{i=1}^{N} a_{ijt} \leq K, \ \forall j \in \mathcal{M}, t \in \mathcal{T}_j \tag{10}$$

where K is the maximal number of UEs that each UAV can accept in each time slot.

B. Local Execution

If the UE decides to execute the task locally, the power consumption for the i-th UE can be given by $k_i(f_{ijt})^{v_i}$, where $j = 0$, $t = 0$, and $k_i \geq 0$ is the effective switched capacitance and v_i can be normally to 3 [1]. Then, the local execution time can be given by $T_{ijt}^C = \frac{F_i}{f_{ijt}}$, ($j = 0$, $t = 0$) and then, the total energy consumption can be given as $k_i(f_{ijt})^{v_i} T_{ijt}^C$.

C. Problem Formulation

Then, one can have the energy consumption of each UE as

$$E_{ijt} = \begin{cases} k_i (f_{ijt})^{v_i} T_{ijt}^C, & \text{if } j = 0, \, t = 0 \\ P_i^{Tr} T_{ijt}^{Tr}, & \text{if } j \in \mathcal{M}, t \in \mathcal{T}_j \end{cases} \quad (11)$$

Also, the total time spent to complete each task can be expressed as

$$T_{ijt} = \begin{cases} T_{ijt}^C, & \text{if } j = 0, \, t = 0 \\ T_{ijt}^{Tr} + T_{ijt}^C, & \text{if } j \in \mathcal{M}, t \in \mathcal{T}_j \end{cases} \quad (12)$$

One can assume that the maximal computation resource which the j-th UAV can provide is as f_j^{max}. Then, one can have

$$C4 : \sum_{i=1}^{N} a_{ijt} f_{ijt} \leq f_j^{max}, \; \forall j \in \mathcal{M}, t \in \mathcal{T}_j \quad (13)$$

Also, as the task normally has to be completed in certain amount of the time and thus without loss of generality, we assume the task must be completed in time T^{max} without loss of generality. In our paper, assume all the transmitting and computing process for each task must be completed within one time interval T^{max}, Then, we have

$$C5 : \sum_{j=0}^{M} a_{ijt} T_{ijt} \leq T^{max}, \; \forall i \in \mathcal{N}, t \in \mathcal{T}_j' \quad (14)$$

Denote $\boldsymbol{A} = \{a_{ijt}, \forall i \in \mathcal{N}, j \in \mathcal{M}', t \in \mathcal{T}_j'\}$, $\boldsymbol{F} = \{f_{ijt}, \forall i \in \mathcal{N}, j \in \mathcal{M}', t \in \mathcal{T}_j'\}$. Then, one can have

$$\mathcal{P} : \min_{\boldsymbol{A}, \boldsymbol{F}} \sum_{i=1}^{N} \sum_{j=0}^{M} \sum_{t=1}^{T_j} a_{ijt} E_{ijt} \quad (15a)$$

$$\text{subject to} \quad (15b)$$

$$C1 : a_{ijt} \in \{0, 1\}, \; \forall i \in \mathcal{N}, j \in \mathcal{M}', t \in \mathcal{T}_j' \quad (15c)$$

$$C2 : \sum_{j=0}^{M} \sum_{t=0}^{T_j} a_{ijt} = 1, \forall i \in \mathcal{N} \quad (15d)$$

$$C3 : \sum_{i=1}^{N} a_{ijt} \leq K, \; \forall j \in \mathcal{M}, t \in \mathcal{T}_j \quad (15e)$$

$$C4 : \sum_{i=1}^{N} a_{ij} f_{ijt} \leq f_j^{max}, \; \forall j \in \mathcal{M}, t \in \mathcal{T}_j \quad (15f)$$

$$C5 : \sum_{j=0}^{M} a_{ijt} T_{ijt} \leq T^{max}, \forall i \in \mathcal{N}, t \in \mathcal{T}_j' \quad (15g)$$

Note that the above problem $\mathcal{P}$ is a MINLP problem, which is difficult to be solved optimally in general. Some existing algorithms like exhaustive search or branch and bound algorithm may solve this problem, but with prohibitive complexity. Therefore, in this paper, we aim to obtain an efficient solution to solve this problem. To this end, we propose the RLAA algorithm to deal with $\mathcal{P}$ effectively and efficiently.

III. Proposed Algorithm

In this section, we show our proposed RLAA algorithm. First, we introduce three important elements in RLAA (i.e., actions, states, and reward functions).

- **Actions:** At each episode eps, each UE takes an action. If the UE decides to offload the task to the j-th UAV in t-th time interval, the action is denoted as ρ_{jt}, $\forall j \in \mathcal{M}, t \in \mathcal{T}_j$. If UE decides to execute the task locally, the action is as ρ_{00}. Then, one can define the collection of actions as follows:

$$\mathcal{C} = \{\rho_{00}, \rho_{11}, ..., \rho_{1T_1}, ..., \rho_{M1}, ..., \rho_{MT_M}\}. \quad (16)$$

For above offloading action ρ_{jt}, $\forall j \in \mathcal{M}, t \in \mathcal{T}_j$, the minimal computation resources of the i-th UE can be given by

$$f_{ijt}^{min} = \frac{F_i}{T^{max} - \frac{D_i}{r_{ijt}}}, \; \forall j \in \mathcal{M}, t \in \mathcal{T}_j \quad (17)$$

For local execution action ρ_{00}, the minimal computation resources of the i-th UE is given as

$$f_{ijt}^{min} = \frac{F_i}{T^{max}}, \; j = 0, t = 0 \quad (18)$$

Note that not all actions can guarantee that the task can be completed within one time interval, as the available computation resources may be less than the minimal computation resources (i.e., f_{ijt}^{min} in (17) and (18)). Similarly, the communication resource can also not be guaranteed (i.e., $\mathcal{C}3$ in (15e)). Therefore we may remove some actions in $\mathcal{C}$, resulting in the collection of feasible actions for the i-th UE as $\mathcal{C}_i$.

- **States:** Then, we define the states as follows:

$$\mathbf{s} = \{\omega_1, \ldots, \omega_i, \ldots, \omega_N\} \quad (19)$$

where ω_i represents the decision of the i-th UE. Specifically, if the i-th UE ($i \in \mathcal{N}$) offloads the task to the j-th UAV in t-th time interval, we assign action ρ_{jt} $\forall j \in \mathcal{M}, t \in \mathcal{T}_j$ to state ω_i. It is worth mentioning that if the i-th UE decides to execute the task locally, we assign action ρ_{00} to state ω_i.

- **Reward Functions:** We define the reward function as

$$Z(\mathbf{s}, \rho_{jt}) = \frac{1}{E_{ijt}} \quad (20)$$

The above proposed reward function can keep reducing the energy consumption of each UE and may finally achieve the minimization of the energy consumption of all UEs.

Then, we present RLAA in Algorithm 1. In the beginning, states $\mathbf{s}$ is initialized. The Q-table is also initialized, which is used to record every state and action (i.e., line 1 in Algorithm 1). At each episode, we obtain the collection of the actions $\mathcal{C}_i$ for the i-th UE. Then, according to the ϵ-greedy policy [13], the i-th UE either chooses a random action with

probability ϵ or follows the greedy policy with probability $1 - \epsilon$, which is expresses as

$$\rho_{jt} = \begin{cases} \rho, & \text{if } \text{rand}(0,1) < \epsilon \\ \underset{\rho_{jt} \in \mathcal{C}_i}{\text{argmax}} Q(\mathbf{s}, \rho_{jt}), & \text{otherwise} \end{cases} \quad (21)$$

where ρ is an action randomly selected from $\mathcal{C}_i$, rand(0,1) denotes a random number uniformly distributed over the interval [0,1] (i.e., line 4 - line 8 in Algorithm 1).

Then, the resource allocation is conducted for the i-th UE (i.e., line 9 in Algorithm 1). If the i-th UE offload the task to the j-th UAV in t-th time slot, the minimal computation resource f_{ijt}^{min} in (17) is allocated. If the i-th UE execute task locally, the minimal computation resource f_{ijt}^{min} in (18) is allocated. Based on the proposed reward function in (20), the i-th UE can then obtain a reward (i.e., line 10 in Algorithm 1).

Next, we update the Q-table (line 11), where the updating rule of Q-table is given as

$$\begin{aligned} Q(\mathbf{s}, c) \leftarrow & Q(\mathbf{s}, c) + \beta \{ Z(\mathbf{s}, c) \\ & + \gamma \max_{c \in \mathcal{C}_i} Q(\mathbf{s}', c) - Q(\mathbf{s}, c) \}, \end{aligned} \quad (22)$$

where γ is the reward decay over the interval [0,1], β is the learning rate over the interval [0,1], and $\mathbf{s}'$ is the next state. Also, states $\mathbf{s}$ is updated based on action ρ_{jt}. Specifically, we assign action ρ_{jt}, $\forall j \in \mathcal{M}', t \in \mathcal{T}_j'$ to state ω_i.

The above process will be repeated until the maximum episode (eps^{max}) is reached. Finally, each UE selects an action according to Q-table (line 16). Specifically, for the i-th UE, the action in C_i corresponding to the largest value of Q-table is selected.

Algorithm 1 Our proposed RLAA

1: Initialize $\mathbf{s}$ and Q-table;
2: **while** $eps \leq eps^{max}$
3:   **for** $i = 1{:}N$
4:     **if** rand(0,1)$< \epsilon$
5:       Select an action ρ_{jt} randomly from $\mathcal{C}_i$ for the i-th UE;
6:     **else**
7:       Select an action $\rho_{jt} = \underset{\rho_{jt} \in \mathcal{C}_i}{\text{argmax}} Q(\mathbf{s}, \rho_{jt})$ for the i-th UE;
8:     **end if**
9:     Allocate computation resource f_{ijt}^{min} from (17) and (18) for the i-th UE;
10:     Obtain a reward $Z(\mathbf{s}, \rho_{jt})$ according to (20);
11:     Update Q-table according to (22);
12:     Update $\mathbf{s}$;
13:   **end for**
14:   $eps \leftarrow eps + 1$;
15: **end while**
16: Select an action for each UE.

IV. SIMULATION RESULTS

In this section, the simulation for the proposed multi-UAV enabled MEC system is conducted, where the parameters of

Parameters	Settings
Radius R_j for all UAVs	800 m
Flying height H_{jt} for all UAVs	350 m
Bandwidth B	1 MHz
Transmission power P_i^{Tr}	1 W
Noise variance σ^2	-90 dbm/Hz
G_0	2.2846
Channel power gain g_0	1.42×10^{-4}
Data Size D_i	$[100, 1000]$ KB
Execution task F_i	$[10^8, 10^9]$ cycles
Time duration T^{max}	1 s
Location of UEs	$[-2000, 2000] \times [-1000, 1000]$ m
f_j^{max}	150 GHz
ϵ-greedy policy probability	0.9
Reward decay γ	0.9
Learning rate β	0.2
k_i for all UEs	10^{-27}
v_i for all UEs	3
eps^{max}	10000
T_j for all UAVs	12

the tests are shown in Table. I, in which the channel bandwidth is set to $B = 1$ MHz, the noise variance is set to $\sigma^2 = -90$ dbm/Hz, the channel power gain at the reference distance $1\ m$ is set to $g_0 = 1.42 \times 10^{-4}$ [12], the transmission power P_i^{Tr} is set to 1 W, the time interval T^{max} is set to 1 s, the k_i is set to 10^{-27} for all the UEs. Also, we assume each UAV can support $K = 150$ UEs in one time slot. All UEs are assumed to be randomly distributed in a rectangle area of coordinates $[-2000, 2000] \times [-1000, 1000]$ m. We randomly select the data size D_i of each task from the interval of $[100, 1000]$ KB and select F_i from the interval of $[10^8, 10^9]$ cycles.

In order to evaluate the performance of our proposed RLAA, the following four algorithms are used as comparison algorithms.

- **Exhaustive search (ES):** We examine all the possibilities, with the objective of minimizing the overall energy consumption for all the UEs.
- **Local execution (LE):** We assume all tasks are executed locally and there are no offloading.
- **Random offloading (RO):** Each UE randomly selects the UAV and the time slot to offload its task.
- **Greedy offloading (GO):** Each UE selects the nearest UAV to offload its task. If the UAV is overloaded (i.e., $C3$ is violated), then selects the second nearest UAV to offload and so on.

Firstly, we compare the performance of RLAA with its four compared algorithms on a set of small scale instances (i.e., the number of UEs ranges from 3 to 7). We assume that there are two UAVs flying in circles with the same radius and the center coordinates of two UAVs are set to $[1200, 1200, 350]$ and $[-1200, -1200, 350]$, respectively. From Fig. 2, one can see that RLAA has the same performance as ES, both of which can achieve the minimal enery consumption. Also, one can see that GO achieves better performance than RO, whereas LE achieve the worst performance for all examined values. This
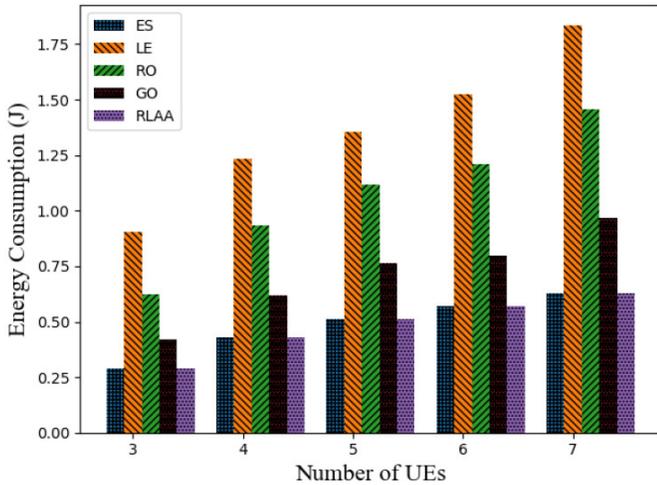
Fig. 2. The overall energy consumption of ES, LE, RO, GO and RLAA versus the number of UEs.
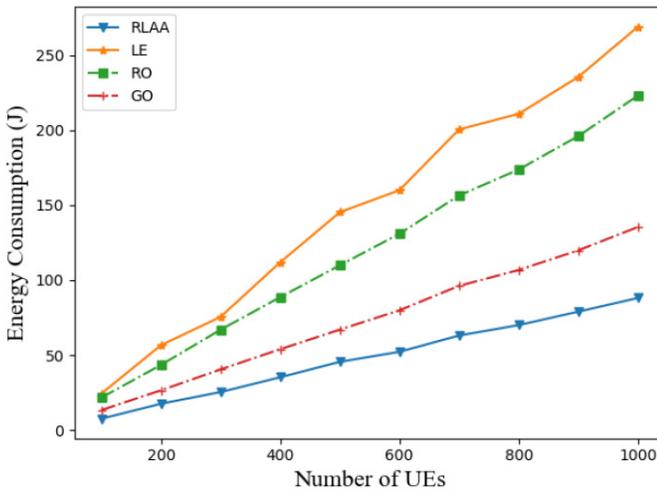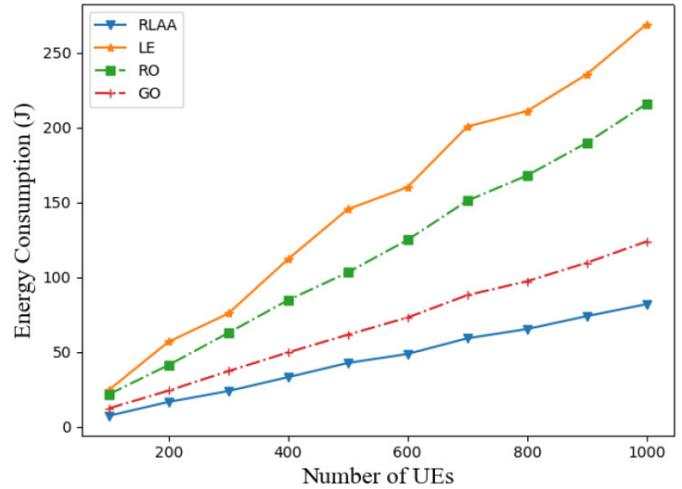

Fig. 4. The overall energy consumption of RLAA, LE, RO and GO versus the number of UEs with 5 UAVs.

and $[0, 0, 350]$, respectively. One sees that our proposed RLAA still outperforms other compared algorithms, with significant amount of energy being saved for all the UEs.

V. CONCLUSION

In this paper, we studied a multi-UAV enabled MEC system, in which the UAVs are assumed to fly in circles over the ground UEs to provide the computation services. The proposed problem is formulated as a MINLP, which is hard to deal with in general. We propose a RLAA algorithm to address it effectively. Simulation results show that RLAA can achieve the same performance as the exhaustive search in small scale cases, whereas in large case scenario, RLAA still have considerate performance gain over other traditional approaches.

VI. ACKNOWLEDGEMENTS

Fig. 3. The overall energy consumption of RLAA, LE, RO and GO versus the number of UEs with 3 UAVs.

is because that our proposed RLAA can choose most energy-efficient action for all the UEs according to computation and communication requirement, while others either make UE to execute all the task locally (i.e., LE), or randomly offload the tasks (i.e., RO), or just find the nearest UAV (i.e., GO), resulting in worse performance.

Next, we compare the performance of RLAA with LE, RO and GO on a set of large scale instances, where the number of UEs is increased to 100∼1000. The number of the UAVs is set to 3, where the center coordinates are $[1200, 1200, 350]$, $[-1200, -1200, 350]$ and $[-1200, 1200, 350]$, respectively. Note that we do not examine ES here, due to its prohibitive complexity. From Fig. 3, one can see that our proposed RLAA still performs best, followed by GO, RO and LE, as expected.

In Fig. 4, we further increase the number of UAVs to 5, where the center coordinates are set to as $[1200, 1200, 350]$, $[-1200, -1200, 350]$, $[-1200, 1200, 350]$, $[1200, -1200, 350]$

REFERENCES

[1] X. Lyu, H. Tian, W. Ni, Y. Zhang, P. Zhang, and R. P. Liu, "Energy-Efficient Admission of Delay-Sensitive Tasks for Mobile Edge Computing," *IEEE Transactions on Communications*, vol. 66, no. 6, pp. 2603–2616, June 2018.

[2] K. Yang, S. Ou, and H. Chen, "On effective offloading services for resource-constrained mobile devices running heavier mobile internet applications," *IEEE Communications Magazine*, vol. 46, no. 1, pp. 56–63, January 2008.

[3] L. Zhang, K. Wang, D. Xuan, and K. Yang, "Optimal Task Allocation in Near-Far Computing Enhanced C-RAN for Wireless Big Data Processing," *IEEE Wireless Communications*, vol. 25, no. 1, pp. 50–55, February 2018.

[4] H. Mei, K. Wang, and K. Yang, "Multi-layer cloud-ran with cooperative resource allocations for low-latency computing and communication services," *IEEE Access*, vol. 5, pp. 19 023–19 032, 2017.

[5] X. Wang, K. Wang, S. Wu, S. Di, K. Yang, and H. Jin, "Dynamic resource scheduling in cloud radio access network with mobile cloud computing," in *2016 IEEE/ACM 24th International Symposium on Quality of Service (IWQoS)*, June 2016, pp. 1–6.

[6] Q. Wu and R. Zhang, "Common throughput maximization in uav-enabled ofdma systems with delay consideration," *IEEE Transactions on Communications*, vol. 66, no. 12, pp. 6614–6627, Dec 2018.

[7] X. Chen, "Decentralized Computation Offloading Game for Mobile Cloud Computing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 4, pp. 974–983, April 2015.

[8] X. Wang, K. Wang, S. Wu, S. Di, H. Jin, K. Yang, and S. Ou, "Dynamic Resource Scheduling in Mobile Edge Cloud with Cloud Radio Access Network," *IEEE Transactions on Parallel and Distributed Systems*, vol. 29, no. 11, pp. 2429–2445, Nov 2018.

[9] J. Lyu, Y. Zeng, R. Zhang, and T. J. Lim, "Placement Optimization of UAV-Mounted Mobile Base Stations," *IEEE Communications Letters*, vol. 21, no. 3, pp. 604–607, March 2017.

[10] Y. Chen, N. Zhao, Z. Ding, and M. Alouini, "Multiple UAVs as Relays: Multi-Hop Single Link Versus Multiple Dual-Hop Links," *IEEE Transactions on Wireless Communications*, vol. 17, no. 9, pp. 6348–6359, Sept 2018.

[11] L. Yang, J. Cao, S. Tang, T. Li, and A. T. S. Chan, "A Framework for Partitioning and Execution of Data Stream Applications in Mobile Cloud Computing," in *2012 IEEE Fifth International Conference on Cloud Computing*, June 2012, pp. 794–802.

[12] H. He, S. Zhang, Y. Zeng, and R. Zhang, "Joint Altitude and Beamwidth Optimization for UAV-Enabled Multiuser Communications," *IEEE Communications Letters*, vol. 22, no. 2, pp. 344–347, Feb 2018.

[13] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.