Federated Learning based Trajectory Optimization for UAV Enabled MEC

Anushka Nehra*, Prakhar Consul†, Ishan Budhiraja‡, Gagandeep Kaur §, Nidal Nasser¶, and Muhammad Imran‖
*Computer Science and Engineering, Thapar Institute of Engineering and Technology, Patiala, Punjab, India
†‡§School of Computer Science Engineering and Technology, Bennett University, Greater Noida, India
¶ College of Engineering, Alfaisal University, Riyadh, Saudi Arabia
‖ Institute of Innovation, Science and Sustainability, Federation University, Brisbane, Australia
Emails: (e21soep0004@bennett.edu.in, nehraanushka3@gmail.com, ishan.budhiraja@bennett.edu.in, gagandeep.kaur@bennett.edu.in, nnasser@alfaisal.edu, and dr.m.imran@ieee.org)

ICC 2023 - IEEE International Conference on Communications | 978-1-5386-7462-8/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICC45041.2023.10278857

***Abstract*—We present a moving mobile edge computing architecture in which unmanned aerial vehicles (UAV) serve as an equipment, providing computational power and allowing task offloading from mobile devices (MD). By improving user association, resource allocation, and UAV trajectory, we optimizing the energy consumption of all MDs. Towards that purpose, we provide a Trajectory optimization technique for making real-time choices while considering all the situation of the environment, followed by a DRL-based Trajectory control approach (RLCT). The RLCT approach may be adapted to any UAV takeoff point and can find the solution faster. The FL is introduced to address the Optimization problem in a Semi-distributed DRL technique to deal with UAV trajectory constraints. The proposed FRL approach enables devices to rapidly train the models locally while communicating with a local server to construct a network globally. The simulation results in the result section shows that the proposed technique RLCT and FRL in the paper outperforms the existing methods, while the FRL performs best among all.**

***Index Terms*—MEC, Unmanned aerial vehicle, DRL, Federated learning**

I. INTRODUCTION

The proliferation in mobile devices is growing gradually due to the anticipation by the consumers that it gives them a more pleasant life than ever before, due of the prominence of computationally intensive activities like smart navigation and augmented reality etc. But, due to the very small size and resource limitations, such as compute and energy, existing smart devices and mobile devices (MD) not able to deliver sufficient Quality of Service (QoS) and Quality of Experience (QoE) while performing such difficult activities [1, 2]. A novel computing approach known as mobile edge computing (MEC) allows MDs for task offloading to the MEC servers at network's edge. However, MEC improves MD computing capability by transferring resource-constrained tasks and services to the MEC server.

Moving MEC has recently introduced, which takes it a step further by assuming that the computational resources may be handled by UAVs. Moving MEC inherits UAV advantages and is supposed to deliver more simpler, and quicker computing services in place of standard stable-location MEC infrastructures. However the use of moving mobile edge computing also introduced various challenges like: 1) how to reduce energy utilization of all MDs by selecting correct user association; 2) how to manage trajectory of UAV's in real time (flying position and direction), particularly in a real - time scenario. Traditional techniques, such as exhaustive search, are difficult to address the mentioned issues for example, determining the ideal trajectory and resource allocation, is continuous rather than discrete [3, 4]. Real-time decision making is introduced by applying a DRL-based Trajectory control system. Two deep Q networks (DQNs) [5, 6], namely the actor and critic networks, are used in this, with the actor network controlling of direction and flying distance of the UAV has been done and the critic network used for analyzing the actor network's activities.

DRL combines DL with RL, which give quicker convergence time. Also, it is found to be more emerging for systems with large amount of state and action spaces. Deep Q-learning is a well-known DRL method. It employs deep Q networks (DQN) as function approximations to the Q-learning technique. However, the fundamental disadvantage of DQN is that in it there are discrete output decisions, resulting in quantization error for continuous action spaces. To handle this issue, DDPG approach is proposed. DDPG is an improved version of deterministic policy gradient, incorporates the benefits of experience replay and target network from DQN to boost learning stability. This makes it effective for continuous action tasks.

A. Related Work

In this section, different type of problems and their given solutions by the researcher related to UAV, MEC and DRL are studied. The authors [7] described a protocol in a UAV-enabled multiple users communication system. They divided the terrestrial stations into separate clusters and used the UAV as a flying cell tower. They then enhanced the throughput in UAV-enabled downlink packet forwarding, downlink broadcasting, and uplink spread spectrum models by collaboratively improving the UAV altitude and antenna beam width. Furthermore, some recent literature has focused on mobile edge computing (MEC), which is seen as a potential solution for delivering computing resources to the edges of WCN [8], where MDs can improve offloading to the MEC servers. MEC with UAV studied in [9, 10]. A heterogeneous MEC (H-MEC) design with stationary ground stations and UAVs was presented out in [11]. The authors of [12, 13] evaluated UAV-enabled MEC, which uses WPT technique to operate IoT devices and gather data from them. DRL has now received much interest for its use in addressing optimization challenges. The authors of [14, 15] suggested an RL framework that use DQN as the function approximator. Although DRL has had considerable success in game-playing applications, it is still a study topic in UAV enabled MEC.

B. Motivation

Based on the challenges listed above, we identified that in [2]-[5], optimization theory is mostly used to produce optimum solutions, such as trajectory design and resource allocation. However, handling such optimization issues usually needs a large amount of computational resources and takes a very long time. This gives motivation to explore the energy usage for MEC across various groups of MDs underlying UAV by improving UAV trajectory. To solve this, we optimized the energy consumption and UAV trajectory for the MEC network using a DRL-based Trajectory control algorithm (RLCT) and FL in this study.

C. Contribution

In this research, we provide a trajectory optimization method for the MEC-UAV system based on federated reinforcement learning. The contributions in this paper are as follows:

- UAV enabled MEC system is introduced in this paper. In addition, an optimization problem of trajectory control and overall energy minimization is formulated with respect to task allocation.
- To transform the optimization problem into the DRL problem to control the trajectory (RLTC), MDP is used.
- Simulation results shows how trajectory and energy is optimized with the help of DRL and federated learning algorithm.

D. Organization

In the following manner this article is organized: System model and the concept of the formulation of problem is described in Section II. Section III has the solutions of the problem. Section IV contains numerical findings, while proposed study concludes in Section V.

II. System Model

Let us consider a scenario where $\mathcal{X}$ MDs represented as $\mathcal{X} = \{1, 2, ..., x, ..., X\}$ and $\mathcal{Y}$ UAVs represented as $\mathcal{Y} = \{1, 2, ..., y, ..., Y\}$, as shown in the architecture in Fig.1. Due to the QoS requirement, let assume that the k^{th} MD generates a task $K_k(s)$ in the s^{th} time slot, which

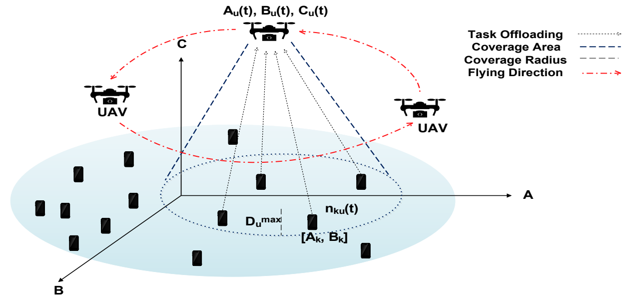

Fig. 1: System Model: Multiple UAV's Enabled MEC System

must be completed within a time duration S^{max}. We also assume that the complete procedure takes S time slots. As a result, for each MD, S tasks will be produced, and s ϵ $\mathcal{S} = \{1, 2, ..., S\}$ and

$$K_k(s) = \{W_k(s), C_k(s)\}, \forall k \ \varepsilon \ \mathcal{X}, \ s \ \varepsilon \ \mathcal{S} \tag{1}$$

where $W_k(s)$ is the amount of information that must be communicated to a UAV, and $C_k(s)$ represents required CPU cycles to complete the task. Let assume that each MD has the option of offloading the task to any of the UAV or doing it locally. After we have

$$n_{ku}(s) = \{0, \ 1\}, \forall k \ \varepsilon \ \mathcal{X}, \ u \ \varepsilon \ \mathcal{Y}, \ s \ \varepsilon \ \mathcal{S} \tag{2}$$

Whereas $n_{ku}(s)$ = 1, u $\neq$ 0 indicates that the k^{th} MD decide to offload the task to the u^{th} UAV in the s^{th} time slot, $n_{ku}(s)$ = 1, u = 0 indicates that the k^{th} MD performs the task by self in the s^{th} time slot, and $n_{ku}(s)$ = 0. Let a new set u ε $\mathcal{Y}$ = {1, 2, . . ., Y} define to represent the potential locations where MD operations might be conducted, where u = 0 indicates that the MD does its own activities locally without offloading.

Furthermore, let each MD served by a UAV or by itself, and that each operation can only be executed in a location. Then, it must follows:

$$\sum_{u=0}^{Y} n_{ku}(s) = 1, \ \forall k \ \varepsilon \ \mathcal{X}, \ s \ \varepsilon \ \mathcal{S} \tag{3}$$

A. Movement Model of UAV

Assume the u^{th} UAV moving at a height and has maximum coverage area horizontally, which fully depends on azimuth angle and flight altitude of the antenna. Also, suppose that the u^{th} UAV can fly horizontally in the s^{th} time slot.

$$0 \leq \phi_u^l(s) \leq 2\pi, \ \ \forall k \ \varepsilon \ \mathcal{X}, \ s \ \varepsilon \ \mathcal{S} \tag{4}$$

and distance as

$$0 \leq h_u(s) \leq h^{max}, \ \ \forall k \ \varepsilon \ \mathcal{X}, \ s \ \varepsilon \ \mathcal{S} \tag{5}$$

Because of the restricted power budget, let h^{max} is the maximum flying distance. In this research, we designate the coordinate of the u^{th} UAV in the s^{th} time slot as $[A_u(s), B_u(s), C_u]$, and the coordinates of the u^{th} UAV, $[A_u(0), B_u(0), C_u]$ representing the starting position.

Furthermore, each UAV may only travel inside a rectangle whose side lengths are designated by A^{max}, and B^{max}. Then, it has

$$0 \leq A_u(s) \leq A^{max}, \ \ \forall k \ \varepsilon \ \mathcal{X}, \ s \ \varepsilon \ \mathcal{S} \tag{6}$$

and

$$0 \leq B_u(s) \leq B^{max}, \ \ \forall k \ \varepsilon \ \mathcal{X}, \ s \ \varepsilon \ \mathcal{S} \tag{7}$$

The u^{th} UAV has a constant velocity $\mathbb{V}_u(s)$ that changes with respect to flying distance $h_u(s)$ in a time slot.

$$\mathbb{V}_u(s) = \frac{h_u(s)}{S^{max}}, \ \ \forall k \ \varepsilon \ \mathcal{X}, \ s \ \varepsilon \ \mathcal{S} \tag{8}$$

B. Execution of Task

If the k^{th} MD outsource the task to the u^{th} UAV during the s^{th} time slot, the $D_{ku}(s)$ (horizontal distance) may be calculated as:

$$D_{ku}(s) = \sqrt{(A_u(s) - A_k)^2 + (B_u(s) - B_k)^2} \tag{9}$$

where $[A_k, \ B_k]$ is the k^{th} MD's coordinate. Furthermore, we suppose that every UAV has a maximum azimuth angle ϕ^{max1}. Thus, the optimal horizontal coverage of the u^{th} UAV D^{max} in each time slot may be determined as follows

$$D^{\max} = C_u tan(\phi^{\max}) \tag{10}$$

As a result

$$n_{ku}(s) D_{ku}(s) \leq D^{\max}, \ \forall k \ \varepsilon \ \mathcal{X}, \ u \ \varepsilon \ \mathcal{Y}, \ s \ \varepsilon \ \mathcal{S} \tag{11}$$

Free space model is used in this research. As a result, the data rate at uplink is

$$d_{ku}(s) = B \ log_2\left(1 + \frac{\gamma P^S}{C_u^2 + D_{ku}^2(s)}\right), \ \ \forall k \ \varepsilon \ \mathcal{X}, \ u \ \varepsilon \ \mathcal{Y}, \ s \ \varepsilon \ \mathcal{S} \tag{12}$$

where bandwidth denoted by B and transmission power of the k^{th} MD denoted by P^S.

If the k^{th} MD transfer its function to the u^{th} UAV in the s^{th} time slot, the overall task execution time is provided by

$$S_{ku}^{Total}(s) = S_{ku}^{Off.}(s) + S_{ku}^{Compt.}(s), \ \forall s \ \varepsilon \ \mathcal{S} \tag{13}$$

The total energy consumed during task $E_{ku}(s)$ is provided by

$$E_{ku}(s) = \begin{cases} E_{ku}^{Loc}(s), & \text{E}xecution \ Locally, \\ E_{ku}^{Off.}(s), & \text{O}ffloading, \end{cases} \tag{14}$$

and to execute the task $T_{ku}(s)$ time will be

$$S_{ku}(s) = \begin{cases} S_{ku}^{Loc}(s), & \text{E}xecution \ Locally, \\ S_{ku}^{Off.}(s), & \text{O}ffloading, \end{cases} \tag{15}$$

Without compromising generality, let assume that all task must be done within the maximum time length S^{max}, which corresponds to the maximum flight time in each time slot.

$$S_{ku}(s) \leq S^{max}, \forall k \ \varepsilon \ \mathcal{X}, \ u \ \varepsilon \ \mathcal{Y}', \ s \ \varepsilon \ \mathcal{S} \tag{16}$$

C. Formulation of Problem

The goal of the article is optimization of the UAV trajectory. A network is mathematically represented as follows:

$$\begin{aligned} P.F. \quad &: \quad \min_{k,\phi,U} \sum_{k=1}^{X} \sum_{u=0}^{Y} \sum_{s=1}^{S} n_{k,u}(s) E_{k,u}(s) \quad (17) \\ s.t. \ \ C_1 \quad &: \quad n_{k,u}(s) = \{0, 1\}, \forall k \ \varepsilon \ \mathcal{X}, \ u \ \varepsilon \ \mathcal{Y}', \ s \ \varepsilon \ \mathcal{S} \\ C_2 \quad &: \quad \sum_{u=0}^{Y} n_{k,u}(s) = 1, \forall k \ \varepsilon \ \mathcal{X}, \ s \ \varepsilon \ \mathcal{S} \\ C_3 \quad &: \quad \sum_{k=1}^{X} n_{k,u}(s) \leq Q^{max}, \forall u \ \varepsilon \ \mathcal{Y}, \ s \ \varepsilon \ \mathcal{S} \end{aligned}$$

The given problem is a MINLP problem, as it contains integer and continuous variables, and is extremely hard to tackle. We first proposed DRL based RLCT to enable smart decision in a fast-changing environment. To do this, we first need to optimize the user association ϕ and resource allocation U in reference to the trajectory of UAV G. The UAV trajectory coordinates G is then optimized in context of the user association ϕ and resource allocation U. Constraint C1 representing the time taken by the MDs to offload the data and C2 take care of that the local computation for all MD pairs should be finished on time. Constraint C3 describe the present location of the UAV in its trajectory.

III. Proposed Solution

To calculate the distinguished offloading policy by reducing UAV's drained energy, and the area of the abandon assignment during the whole decision incident is our objective of this work.We review the standard FL (federated learning) framework and construct the UAV-assisted edge evaluating offloading managing and resource-distributive problem as a Markov decision process (MDP).

A. Markov Decision Process (MDP)

To apply the DRL, we need to first introduce the MDP and define the State ($\mathbb{S}$), Action ($\mathbb{A}$) and Reward ($\mathbb{R}$). It showcase the required variables in the DRL framework:

1) State ($\mathbb{S} = [A_u(s), B_u(s), C_u]$, $\forall u \, \varepsilon \, \mathcal{Y}$): It is basically group of the coordinates of all the UAVs in the system model.

2) Actions ($\mathbb{A}$: The action space includes the group of possible offloaded UAVs. It is the collection of each UAV's individual tasks, including the horizontal direction $\phi_u^l(s)$ and distance $h_u s$. Then, the set of action can be expressed as $\mathbb{A} = [\phi_u^l(s), \ h_u s], \ \forall u \ \varepsilon \ \mathcal{Y})$.

3) Reward ($\mathbb{R}$): It is considered as the negative of the total energy consumed by all MDs in each time slot.

$$\mathbb{R} = -\sum_{k=1}^{X} \sum_{u=0}^{Y} n_{k,u}(s) E_{k,u}(s) - f \tag{18}$$

where f is the fault cost if any of the UAVs fly outside of the core zone.

B. Reinforcement Learning based Trajectory Control (RLTC)

DRL-based Trajectory Control algorithm (RLTC) is introduced, which combines DNN and matching algorithms. According to the algorithm architecture employed in this article, where an agent, which might be placed at the base station's control centre, is supposed to interact with the environment.

The MD's will be chosen by each UAV's based on the following criteria:

- MD's must be within its coverage area;
- MD's can save more energy

In Algorithm, we will go through the fundamentals of the presented algorithm. Reward can be acquired with the following algorithm. The RLTC technique is an offline learning algorithm based on DRL that uses the fundamentals of replay mechanism, and the mini-batch sampled several random variables episodes for training networks at time step. Furthermore, the training approach performed in a simulator, and the RLTC can immediately deployed in practice once convergence is fulfilled, which results lowering the operational cost. Furthermore, after the entire network has converged, solutions may be obtained extremely quickly using only a few basic algebraic computations rather than solving the initial MINLP. This is because, during the training phases, random takeoff spots for all UAVs are created.

C. DDPG

1) Actor Process: The actor calculates the gradient of the POF in order to improve policy. The DPG is calculated as follows:

$$\nabla_{\phi}\mathbb{F}(\phi_\alpha) = \int_{\mathbb{S}} D_{\phi_\alpha}(s)\nabla_\alpha \phi_\alpha(s)\nabla_\alpha \mathbb{Q}\big((s,a)|_{a=\phi_\alpha} ds$$
$$= \mathbb{E}_{\phi_\alpha}\big[\nabla_\alpha \phi_\alpha(s)\nabla_\alpha \mathbb{Q}\big(s,a\big)|_{a=\phi_\alpha} ds\big]. \quad (19)$$

When employing neural networks to estimate the state value function, the training set is considered to be self-reliant and uniformly distributed. But, the data gathered by the agent frequently contains a high degree of correlation, making the RL model unstable if such data is considered for training. The experience replay technique has the ability to break the correlation between the collected data.

Let experience at time slot s is represented by $\big(S^r_{v,s}, a^r_{v,s}, \mathcal{F}^r_{v,s}, S^r_{v,s+1}\big)$ and it is recorded in the experience replay buffer using a bounded storage capacity of $\mathbb{C}$. The buffer updates the experience on a regular basis by gathering new samples and removing old ones. The Monte-carlo approach is used to compute the prediction in the replay buffer by randomly sampling mini-batch of capacity Z. Therefore, Eq. (19) is reformulated as:

$$\nabla_\alpha \mathbb{F}(\phi_\alpha) \approx$$
$$\frac{1}{Z}\sum_{z=1}^{Z}\big(\nabla_\alpha \mathbb{Q}(S)\big)|_{S=S^m_{i,s}, a=\phi(S^m_{i,s})}\nabla_\alpha \phi_\alpha(S)|_{S=S^m_{i,s}}, \quad (20)$$

where $\mathbb{Q}(S,a)$ represents the state value function created by critic network.

2) Updation Process: $\mathbb{Q}^{\beta'}\big(S(s+1), \phi_{\alpha'}\big(S(s+1)\big)\big)$ is the target $\mathbb{Q}$ value and $\mathbb{Q}^{\beta}\big(S(s) - a(s)\big)$ is the estimated on-line $\mathbb{Q}$ value. Updated parameters are defined as follows:

$$\beta(s+1) = \beta(s) + \tau_\beta \delta(s)\nabla_\alpha \mathbb{Q}^\beta\big(S(s), a(s)\big). \quad (21)$$

3) Critic Process: The actor process's performance is determined by the critic process. It has two DNN models same as the actor network: the critic Q process network and the target Q process. DNN is used in the critic network to estimate the value function and is caculated as $\mathbb{Q}_\beta(S,a) \approx \mathbb{Q}(S,a)$.Let $\beta^Q = \big[\beta_1, \beta_2, \ldots, \beta_n\big]$ denotes critic Q network's parameters, $\beta^{Q'} = \big[\beta^{'}_1, \beta^{'}_2, \ldots, \beta^{'}_n\big]$ denotes target Q network's parameters. Now mini batch of k^{th} transition samples, i.e.$S(k)$ and $a(k)$, is extracted from the replay buffer in critic Q network . Then these samples are given to the DNN to evaluate the Q value $Q\big(S(k), a(k)|_{\beta Q}\big)$. Concurrently in the target Q' network, the mini-batch sample $\mathcal{F}(k)$ and $S(k+1)$ are given to DNN to create the target Q value $y(k)$, which is determined as follows:

$$y(k) = \mathcal{F}(k)\Upsilon Q'\big(S(k+1), \phi'\big(S(k+1)|\beta^{Q'}\big)\big), \quad (22)$$

where $\phi'\big(S(k+1)|\beta^{Q'}$ indicates the estimation of $a(s+1)$ and is generated by the target actor network ϕ'.

The loss function must be minimized for each learning step in order to modify the critic network and is defined as follows:

$$\mathbb{L}_f \approx \frac{1}{Z}\sum_{z=1}^{Z}\big(y(k) - \mathbb{Q}\big(s(k), a(k)\big)|_{\alpha Q}\big)^2 \quad (23)$$

D. Federated Reinforcement Learning (FRL)

To deal with UAV trajectory restrictions, we use FL to solve the optimization issue in a semi-distributed DRL manner. The proposed FRL approach enables devices to speedily train their local models while communicating with a parameter server to construct a global network. Such interactions need simply the upload of metadata and the receiving of regulations, significantly lowering communication traffic. Furthermore, these devices are unable to acquire information from the variables of others, ensuring the privacy of the system. Rather of combining state data, each UAV trains the pre - trained model by giving its state data and altering server parameters. The resource server then gets authenticated variables from each UAV and begins training the global Q-network. The downloaded parameters are used to federated averaging in order to fit the global Q-network. The parameter host then assigns global Q-network parameters to each UAV. Each UAV enhances its network and bases its decisions on the best possible policy.

Furthermore, all UAVs in the FRL approach share the same network structure. Their models are aggregated using the federated averaging approach. The preferred FRL algorithm technique is presented in Algorithm 1. The

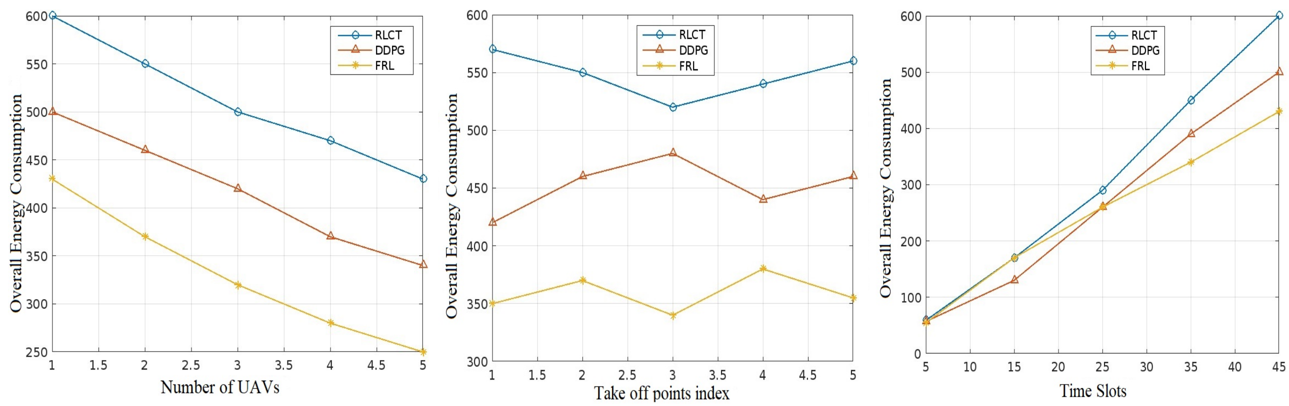

Fig. 2: Comparative Analysis (a) Overall energy consumption v/s Number of UAVs (b) Overall energy consumption v/s Take off points index (c) Overall energy consumption v/s Time Slots

Algorithm 1 FRL Algorithm for UAV-MEC System.

Requirement:
- Memory (Replay) of size Y,
- Greedy prob. is ϵ,
- Learning rate is ϕ

for u = 1,$\ldots$,X **do**
  Memory size Y initialization;
  Q-Network with variable G_u initialized;
  Target Q-Network with variable $G^- = G_u$ initialized;
  for ep. = 1,$\ldots$,X **do**
    for y = 1,$\ldots$, $\mathcal{Y}$ **do**
      u^{th} UAV chooses a Random actions a_k;
      The u^{th} UAV gets the reward r_k and move to next state s_{k+1};
      Transition matrix (s_k, a_k, r_k, s_{k+1}) stored in Relay memory;
      Randomized mini-batch sample (s_u, a_u, r_u, s_{u+1}) in memory Y;
      if Final State is s_{u+1} **then**
        $B_u = r_u$;
      else
        $B_u = r_u + \phi \max_{a'} \widehat{Q}(s_{k+1}, a'; G^-)$;
      end if
      Evaluate Global model $\mathcal{X}_w$ via Federated Averaging;
      for G_w in $\mathcal{X}_w$ **do** |StateUpdate $G_k = G_w$;
      end for
      Reset the Value $\widehat{Q} = Q$;
    end for
  end for
end for
return Trajectory of each UAV..

system creates the global network first, and then each agent creates its own network at the UAV site. The agents then begin training their own networks by observing the states around them and making their own decisions. To reduce communication overhead, FRL only asks agents to send network parameters. Furthermore, data security is not adequately protected for UAVs that use RLTC to make centralized decisions. However, in order to obtain the benefits indicated above, this design compromises some performance. The numerical statistics in the next section demonstrate the detailed system performance discrepancies between the FRL technique, the RLTC methodology, and other benchmarks.

IV. Results and Discussion

The evaluation and discussion of the performance analysis of the proposed FRL, RLTC schemes with other existing schemes has been discussed in this section.

TABLE I: Simulation Parameter Values table

Parameters	Values
Height of UAV	110 m
Greedy Probability	1
Size of Memory	35000
Learning Rate	0.001
Time Slot	0.2 s
Maximum Velocity	25 m/s
Weight of UAV	1.2 Kg
UAV Initial and Final Value	[-25, -25] m
Data Size	[100, 600] kbits
Number of Mobile Devices	250
Training Episodes	1500
Reward Discount	0.9
Minimum of Greedy Probability	0.02
Testing steps	130
Time duration for each round of FL	0.0207 s

A. Numerical Settings

In the numerical settings, we have assumed that there are 80 user equipment dispersed randomly in an area of 400×400 m. Also, three UAVs are deployed to facilitate the user equipment in the area. Other major simulation parameters are shown in Table I. Also, for every time slot, the UAVs transmit a wireless signal in order to on or activate its corresponding user equipment. It will help in either local executing of task or in task offloading in the respective delay. We band all the user equipment into a single cluster, further also the UAVs can fly inside a circular pattern with radial distance of 70 m, 90 m, and 110 m all around the nearest cluster, respectively. These multiple UAV trajectory pairs are referred to as the preliminary trajectories.

B. Overall energy consumption perfrmance analysis

In this, we discuss the overall energy use in the proposed schemes i.e., FRL and RLTC along with comparision to DDPG. Fig. 2(a) shows the overall energy use in the proposed schemes with respect to varying number of UAVs. It is assumed that all the UAVs uses same take off points. It also observed from Fig. 2(a) that the energy use of all the schemes i.e., FRL, RLTC, and DDPG, decreased with increment in number of UAVs. It is because the computatinal capacity increases with increase in the number of UAVs, which leads to more user equipment benefitting from task offloading. Thus, decreasing the overall energy use of the system. It can also be noted that the proposed FRL scheme outperforms DDPG and RLTC in terms of optimizing overall energy use of the system.

Fig. 2(b) shows the overall energy use of the proposed schemes with respect to varying number of take off points. It observed in Fig. 2(b) that proposed FRLmethod performs better than DDPG and RLTC schemes in terms of optimizing the overall energy use of the system.

Figure 2(c) represents the overall energy usage of the considered systems as a function of time slot number. According to Fig. 2(c), the overall energy usage of all systems increases with increasing time slots. Furthermore, the suggested FRL scheme outperforms the DDPG and RLTC schemes in terms of minimizing the overall energy usage.

V. Conclusion

In this article, we presented a moving MEC architecture that takes use of UAV as a moving platform. We want to reduce the energy consumption of all MDs by optimizing the UAV's trajectories, interactive features, and resource allocation. UAV enabled MEC system is introduced in this paper. In addition, an optimization problem of trajectory control and overall energy minimization is formulated with respect to task allocation. We provide a Trajectory optimization technique for making real-time choices while considering all the situation of the environment, followed by a DRL-based Trajectory control approach (RLCT). A DRL-based RLTC with a matching algorithm has been given to handle the issue of multi-UAV trajectory. A federated reinforcement learning (FRL) approach is also presented in order to reduce total energy usage and achieve optimal policy. The simulation results show that RLTC and FRL work efficiently.

References

[1] P. K. Gupta, S. Mittal, P. Consul, and J. K. Jindal, "A Review of Different Vulnerabilities of Security in a Layered Nwtwork," in *Advances and Applications in Mathematical Sciences*, pp. 227–236, Dec. 2020.

[2] X. Pei, H. Yu, X. Wang, Y. Chen, M. Wen, and Y.-C. Wu, "NOMA-Based Pervasive Edge Computing: Secure Power Allocation for IoV," *IEEE Transactions on Industrial Informatics*, vol. 17, pp. 5021–5030, Jul. 2021.

[3] 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*, vol. 7, pp. 117448–117459, Aug. 2019.

[4] W. Wu, F. Zhou, R. Q. Hu, and B. Wang, "Energy-Efficient Resource Allocation for Secure NOMA-Enabled Mobile Edge Computing Networks," *IEEE Transactions on Communications*, vol. 68, pp. 493–505, Jan. 2020.

[5] Z. Ding, D. W. K. Ng, R. Schober, and H. V. Poor, "Delay Minimization for NOMA-MEC Offloading," *IEEE Signal Processing Letters*, vol. 25, pp. 1875–1879, Dec. 2018.

[6] Z. Ding, P. Fan, and H. V. Poor, "Impact of Non-Orthogonal Multiple Access on the Offloading of Mobile Edge Computing," *IEEE Transactions on Communications*, vol. 67, pp. 375–390, Sep. 2019.

[7] L. Qian, Y. Wu, F. Jiang, N. Yu, W. Lu, and B. Lin, "NOMA Assisted Multi-Task Multi-Access Mobile Edge Computing via Deep Reinforcement Learning for Industrial Internet of Things," *IEEE Transactions on Industrial Informatics*, vol. 17, pp. 5688–5698, Aug. 2021.

[8] V. Dat Tuong, T. Phung Truong, A.-T. Tran, A. Masood, D. Shumeye Lakew, C. Lee, Y. Lee, and S. Cho, "Delay-Sensitive Task Offloading for Internet of Things in Nonorthogonal Multiple Access MEC Networks," in *2020 International Conference on Information and Communication Technology Convergence (ICTC)*, pp. 597–599, Dec. 2020.

[9] P. Consul, I. Budhiraja, D. Garg, and A. Bindle, "Power Allocation Scheme Based on DRL for CF Massive MIMO Network with UAV," in *Innovations in Information and Communication Technologies, Singapore*, pp. 33–43, Sep. 2023.

[10] J. Zhang, L. Zhou, Q. Tang, E. C.-H. 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*, vol. 6, pp. 3688–3699, Dec. 2019.

[11] M. Hua, Y. Huang, Y. Wang, Q. Wu, H. Dai, and L. Yang, "Energy Optimization for Cellular-Connected Multi-UAV Mobile Edge Computing Systems with Multi-Access Schemes," *Journal of Communications and Information Networks*, vol. 3, pp. 33–44, Dec. 2018.

[12] H. Sharma, I. Budhiraja, P. Consul, N. Kumar, D. Garg, L. Zhao, and L. Liu, "Federated learning based energy efficient scheme for MEC with NOMA underlaying UAV," in *Proceedings of the 5th International ACM Mobicom Workshop on Drone Assisted Wireless Communications for 5G and Beyond*, pp. 73–78, Oct. 2022.

[13] Y. Dai, D. Xu, S. Maharjan, and Y. Zhang, "Joint Load Balancing and Offloading in Vehicular Edge Computing and Networks," *IEEE Internet of Things Journal*, vol. 6, pp. 4377–4387, Oct. 2019.

[14] P. Consul, I. Budhiraja, R. Chaudhary, and N. Kumar, "Security Reassessing in UAV-Assisted Cyber-Physical Systems Based on Federated Learning," in *MILCOM 2022 - 2022 IEEE Military Communications Conference (MILCOM)*, pp. 61–65, Dec. 2022.

[15] F. Zhou, Y. Wu, R. Q. Hu, and Y. Qian, "Computation Rate Maximization in UAV-Enabled Wireless-Powered Mobile-Edge Computing Systems," *IEEE Journal on Selected Areas in Communications*, vol. 36, pp. 1927–1941, Sep. 2018.