Accepted Manuscript

Application of reinforcement learning in UAV cluster task scheduling

Jun Yang, Xinghui You, Gaoxiang Wu, Mohammad Mehedi Hassan, Ahmad Almogren, Joze Guna

PII: S0167-739X(18)32529-9
DOI: https://doi.org/10.1016/j.future.2018.11.014
Reference: FUTURE 4578

To appear in: *Future Generation Computer Systems*

Received date : 17 October 2018
Revised date : 5 November 2018
Accepted date : 11 November 2018

Please cite this article as: J. Yang, X. You, G. Wu et al., Application of reinforcement learning in UAV cluster task scheduling, *Future Generation Computer Systems* (2019), https://doi.org/10.1016/j.future.2018.11.014

Application of Reinforcement Learning in UAV Cluster Task Scheduling

Jun Yang, Xinghui You, Gaoxiang Wu, Mohammad Mehedi Hassan, Ahmad Almogren, Joze Guna

***Abstract*—Recently, unmanned aerial vehicle (UAV) clusters have been widely used in various applications due to its high flexibility, large coverage and reliable transmission efficiency. In order to achieve the collaboration of multiple UAV tasks within a UAV cluster, we propose a task-scheduling algorithm based on reinforcement learning in this paper, which enables the UAV to adjust its task strategy automatically and dynamically using its calculation of task performance efficiency. As the UAV needs to perform real-time tasks while working in a dynamic environment without centralized control, it needs to learn tasks according to real-time data. Reinforcement learning has the ability to carry out real-time learning and decision making based on the environment, which is an appropriate and feasible method for the task scheduling of UAV clusters. From this perspective, we discuss reinforcement learning that solves the channel allocation problem existing in UAV cluster task scheduling. Finally, this paper also discusses several research problems that may be faced by the further application of UAV cluster task scheduling.**

***Index Terms*—Reinforcement Learning, UAV Cluster, Task Scheduling**

I. Introduction

With the growing role of UAVs in the military, public and civilian fields, it can be widely used for "boring, dirty or dangerous" tasks, which are often inconvenient or unpleasable to human beings [1], [2], [3]. In the process of performing tasks, the UAV has the advantage of flexible deployment on demand, large coverage, and stationary hovering at any time, which usually produces special effects when performing tasks [4], [5], [6]. UAVs have been primarily used for military applications before, but are now rapidly expanding into business, science, entertainment, agriculture, and other fields, as shown in Figure 1. In addition to traditional usage of military applications, more recent examples include support of first responders, surveillance, express delivery, aerial photography, agriculture, and UAV competition [7], [8], [9].

Since UAVs require a reliable uplink transmission to transmit acquired data to the core network, the cluster network must support reliable data transmission. However, with the popularization of UAV deployment, the cooperation of multiple UAVs in the cluster becomes a prominent problem, which makes it necessary to investigate the collaboration protocol and control algorithms [10], [11], [12]. At present, most UAV clusters usually work in a centralized way. For example, the central control unit of the base station (BS) controls the operation of the UAV. With increasing task complexity and more complex and changeable environments, a centralized control mode will be unable to satisfy the demands of real-time and efficient control [13], [14], [15]. In addition, it is also difficult to make centralized control on all UAVs due to the limited spectrum of resources. Therefore, it is critical to study decentralized cooperation methods to solve the design and optimization problems of UAV cluster networks in the sensor network composed of multiple UAVs [16], [17], [18].

In this paper, we propose a decentralized networking protocol to coordinate the movement of UAVs and achieve real-time networking of UAV clusters. As the UAV is in a dynamic environment and performs real-time tasks without centralized control, the UAV needs to learn to collate data and perform transmission online at the same time. Reinforcement learning is an excellent candidate to satisfy these requirements for UAV cluster task scheduling. In this regard, we adopt reinforcement learning to solve the problems existing in real-time sensing of UAVs. Unlike supervised learning which requires offline data sets to learn the correct actions in each state, the agent (that is, the UAV) of reinforcement learning learns from real-time data from various sources. This method is more suitable for application scenarios under real-time UAV scheduling. Furthermore, reinforcement learning does not rely on complete and accurate environmental models, which is particularly useful for intelligent hardware with limited computing power like UAVs [19], [20].

In this paper, we will discuss possible solutions of applying reinforcement learning to UAV cluster task scheduling.

1) We apply the expansion strategy to solve the problem of UAV networking in the initial state.

2) We apply deep reinforcement learning to solve the dynamic allocation problem of wireless channel, so as to optimize the time delay of UAV data transmission.

3) Finally, we provide an example to introduce how to apply the above methods to solve the problem of UAV cluster task scheduling.

The rest of the paper is arranged as follows. Section II gives an overview of the UAV cluster and the tasks it faces.

J. Yang, X. You and G. Wu are with the School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan 430074, China. Email: jun_yang_cs@hust.edu.cn, xinghuiepic@hust.edu.cn.

M. M. Hassan and A. Almogren are with the College of Computer and Information Sciences, King Saud University, Riyadh, 11543, Saudi Arabia. Email: mmhassan@ksu.edu.sa, ahalmogren@ksu.edu.sa.

J. Guna is with the Faculty of Electrical Engineering, University of Ljubljana, Tržaška cesta 25, 1000 Ljubljana, Slovenia. Email: joze.guna@fe.uni-lj.si.

Gaoxiang Wu is the corresponding author (gaoxiangwu@hust.edu.cn).

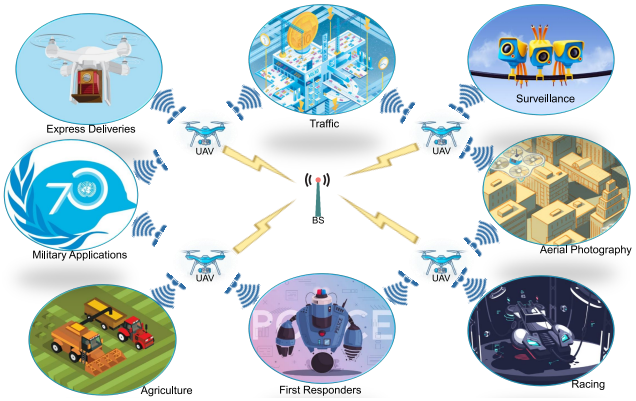

Fig. 1. Application of UAV Cluster

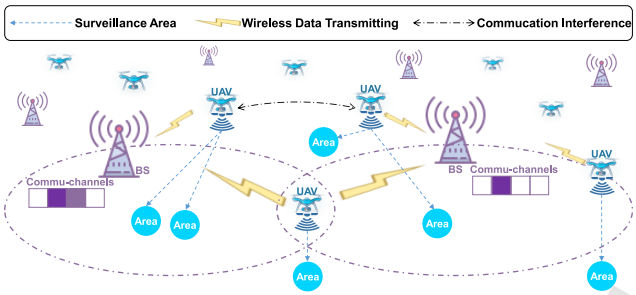

Fig. 2. Schematic Diagram of Performing Real-Time Task Scheduling under the UAV Cluster

In section III, we will discuss the reinforcement learning method, including its theoretical basis and the possible applications in UAV scheduling. In section IV, we elaborate on an example application of the reinforcement learning method in the UAV cluster scheduling task, which is used to solve the UAV data transmission task scheduling problem. In section V, we conclude this paper and discuss open challenges for future work.

II. Overview of UAV Clusters

In this section, we first introduce UAV clusters briefly, then describe the task scenarios faced by UAV clusters and discuss corresponding qualification conditions to the task scenarios.

A. Real-Time Task Scheduling of UAVs in the Cluster Network

As shown in Figure 2, multiple UAVs perform different real-time tasks simultaneously in a cluster network using multiple orthogonal frequency division multiplexing (OFDM), performing the sensing tasks by continuously monitoring their induction areas, and collect or generate real-time sensing data during this period [21], [22], [23]. The purpose of UAVs is to collect effective sensor data and transmit the data to the core network through relay UAVs and BSs. Here, effective sensor data refers to the sensor data containing accurate information of the task situation, that is, the sensor data generated by the successful detection of the target object by each UAV. In general, the effective probability of sensor data collected by the UAV is negatively correlated with the distance between the UAV and the target area. For transmitting the sensor data to the core network, the UAV selects and associates a BS, which then allocates a communication channel to the UAV in order to upload data. According to the deployment strategy of BSs, the frequency band used by adjacent an BS can be the same or different, and it must consider the method of time division multiplexing if the frequency bands are the same. To ensure the success rate of data transmission while optimizing its energy consumption, each UAV dynamically determines its own transmission power while performing the data transmission task while considering the blocking and interference of signals due to obstacles. Furthermore, each UAV also determines its motion trajectory, data perception behavior and data transmission strategy, so as to better collect data and transmit it back to BS.

B. Task Definition of the UAV Cluster

We now define and explain the relevant terms and objects to be used in this paper, which are shown in Table I.

The UAV cluster usually must perform monitoring tasks in specific areas. It must finish the deployment and networking of the UAV cluster in the initial scenario, and then consider the situation of terminating tasks when some UAV nodes fail or have low battery power. In those cases, the cluster is required to re-network. After networking, the UAV cluster must consider the data transmission efficiency of each UAV over the target monitoring area, which is defined as the delay efficiency. For improving the standby time of the UAV, the data transmission power must be considered. Each UAV should dynamically adjust the transmission power of the UAV cluster by analyzing the collective networking status. Oftentimes, multiple UAVs will compete for the same communication channel when performing time-division multiplexing. It is also necessary to optimize scheduling to reduce the collision probability of data transmission.

Therefore, in order to allow each UAV to perform its tasks while satisfying the requirements of the data transmission process, a virtual repulsive algorithm is proposed here to evenly distribute the UAVs in a specific region. Our algorithm optimizes the distance to the base station to reduce the delay time and transmission power. Given that there are N UAVs and M base stations in a specific region and it has a repulsive force between every pair, we model the system as an $(M+N)*(M+N)$ force matrix $A[M+N][M+N]$, where $A[i][i]=0$, $A[i][j]=-A[j][i]$. Taking the center of a specific region to be the origin, we construct the x-axis from west to east and construct the y-axis from south to north, to establish a two-dimensional rectangular coordinate system. As shown in Figure 3, the UAVs are randomly scattered in the initial conditions with coordinates (x_i, y_i),

TABLE I
TABLE OF OBJECT AND TERM DEFINITIONS.

Symbolic Representation	Explanation
N	Total number of UAV
M	Total number of base station
$B_i(x_{B_i}, y_{B_i})$	Base station
CH_{B_i}	the idle channels of Base station i
T-UAV	UAVs performing monitoring tasks
R-UAV	The relay UAV, providing relay communications to T-UAV
$T_i(x_{T_i}, y_{T_i})$	Monitored target
d_{min}	The minimum distance between any UAV-A and UAV-B is the safe distance
d_{max}	The maximum distance between UAV and UAV or UAV and BS is limited by the maximum transmission power of the UAV
t_m	Time needed to complete the monitoring by the UAV
v_s	Cruising Speed of the UAV

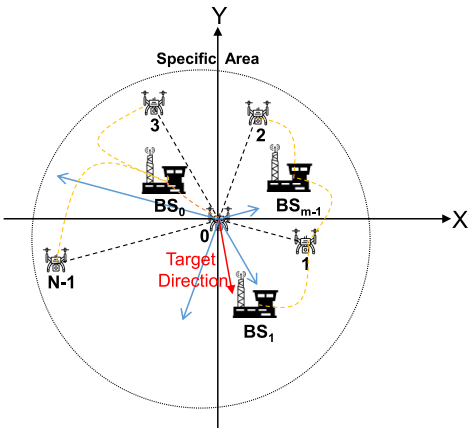

Fig. 3. Propagation Planning of the UAV Cluster

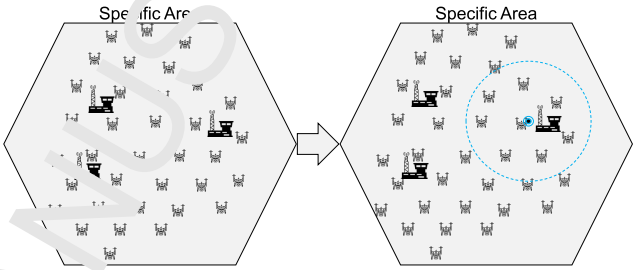

Fig. 4. Human Intervention of the UAV Cluster

and base stations are located with$(x_{B_i}, y_B$). Eve UAV receives $M + N$ forces, and the force f ar y UAV on itself is zero. We calculate the magnitude dire tion of the resultant force, which indicates the direction in which the UAV should move. For computati nal convenience, the repulsive force is decomposed into x and y co ponents. As shown in Equation 1 and Equation 2, t e repulsive force of the i UAV on the x-axis is x_i^m and the epulsive force on the y-axis is y_i^m. The quantities $\frac{x_i - x_j}{|x_i - x_j|}$ a $\frac{y_i - y_j}{|y_i - y_j|}$ refer to the direction of the repulsive forc . C_1 and C_2 are constant, and represent different weight coe ients orresponding to UAV and base station respective y. Furthe more, the UAV searches for the closest base stati n with an available channel.

$$x_i^m = \sum_{j=1}^{N} \frac{C_1}{(x_i - x_j)^2} \frac{x_i - x_j}{|x_i - x_j|} + \sum_{k=1}^{M} \frac{C_2}{(x_i - x_{B_k})^2} \frac{x_i - x_{B_k}}{|x_i - x_{B_k}|} \tag{1}$$

$$y_i^m = \sum_{j=1}^{N} \frac{C_1}{(y_i - y_j)^2} \frac{y_i - y_j}{|y_i - y_j|} + \sum_{k=1}^{M} \frac{C_2}{(y_i - y_{B_k})^2} \frac{y_i - y_{B_k}}{|y_i - y_{B_k}|} \tag{2}$$

$$d_{base} = \sqrt{(ArrayX[i] - x_{B_k})^2 + (ArrayY[i] - y_{B_k})^2} \tag{3}$$

According to Equation 1 and Equation 2, the coordinate of the UAV propagates to $(x_i + x_i^m, y_i + y_i^m)$ unless the UAV is close to the boundary area. After several iterations, if the displacement of each UAV d_{safe} is larger than a d_{min}, and the distance between each UAV and the base station d_{base} (Equation 3) is smaller than d_{max}, and the base station has available channels, the cluster is in a stable configuration and does not propagate further. The algorithm of the UAV cluster networking is shown as Algorithm 1.

In addition, we deploy many intensive UAV clusters in key areas, where human intervention is needed. As shown in Figure 4, we mark the key area manually and indicate its radius. After we mark these area, the algorithm will plan the nearest m UAVs to move into the region automatically, and then perform the UAV cluster scheduling algorithm untill the UAVs in the region reach a balanced state.

III. REINFORCEMENT LEARNING IN UAV CLUSTER SCHEDULING

A. Introduction to Reinforcement Learning

In reinforcement learning, each agent learns to take appropriate action by interacting with the environment and learning from its experience [24], [25]. Through the performance of each action, the reward is given as a quantitative

Algorithm 1 Cluster Scheduling of UAV

Input: UAV clusters in a random state
Output: UAV clusters in a stable state

function SCHEDULEUAV($ArrayX, ArrayY$)
  for $i = 0 \rightarrow N-1$ **do**
    $Tab[i] \leftarrow 0$
  end for
  $count \leftarrow 0$
  while $(count < N)$ **do**
    for $i = 0 \rightarrow N-1$ **do**
      if $(Tab[i] == 0)$ **then**
        $x_i^m \leftarrow Equation(1)$
        $y_i^m \leftarrow Equation(2)$
        $x_i \leftarrow x_i^m + x_i$
        $y_i \leftarrow y_i^m + y_i$
        find the nearest UAV (x_j, y_j) to UAV i
        $d_{safe} \leftarrow \sqrt{(x_i - x_j)^2 + (x_i - y_j)^2}$
        if $(d_{safe} > d_{min}$ **and** $d_{base} < d_{max}$ **and** $CH_{B_k} > 0)$ **then**
          $Tab[i] \leftarrow 1$
          $count \leftarrow count + 1$
          $CH_{B_k} \leftarrow CH_{B_k} - 1$
        end if
        if $((x_i, y_i)$ is at edge**)** **then**
          $Tab[i] \leftarrow 1$
          $count \leftarrow count + 1$
        end if
      end if
    end for
  end while
  return $(ArrayX, ArrayY)$
end function

feedback of the learning process. As reinforcement learning does not require a pre-existing data set for training and the agent can learn from online data, it has a strong appeal for real-time applications. Furthermore, since the behaviors of other agents can be viewed as the state of the environment, it can solve the optimization problem of multiple agents using a decentralization method by extending reinforcement learning. The following is an introduction to deep reinforcement learning, which is based on deep strategy gradient descent methods with actor-critic constraints to optimize value functions which will be used to perform the scheduling task of the UAV.

1) Deep Reinforcement Learning Based on Value Function: In ordinary Q-learning, Q-table is used to store the Q value of each state action pair when the state and action spaces are discrete and the dimension is not high. However, Q-tables are difficult to solve for high-dimensional continuous state or action spaces. Deep reinforcement learning is a combination of deep learning and reinforcement learning, which can directly learn control strategies from high-dimensional data. The deep neural network can extract complex features automatically, and represent the input high-

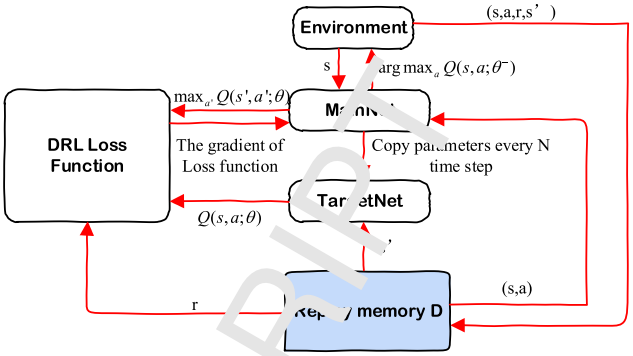

Fig. 5. DQN Training Process

dimensional state-action pair as low dimension approximation, and the output is the corresponding Q value computed for each action. Figure 5 shows the training process of deep Q-learning network (DQN). The comparison value function has the following three characteristics:

1) An experience replay mechanism is applied in the training process to store the transferred samples obtained by the interaction between the intelligent body and the environment in the replay memory unit. A small batch of transferred samples is selected randomly in the training process. The network parameter θ is updated according to an optimal gradient descent. This random sampling method greatly reduces the correlation between samples and improves the stability of the algorithm.

2) In addition to using the deep neural network to represent the current value function, it uses a separate network to generate the target Q value. Specifically,$Q(s, a|\theta_i)$ is the output of the current network and is the value function for computing the current state-action pair. $Q(s, a|\theta_i^-)$ is defined as the output of the target network, and it generally adopts $Y_i = r + \gamma max_{a'} Q(s'a'|\theta_i^-)$ to approximately represent the optimization target of the value function, which is the target Q value. The parameter, θ of the current value network, is updated in real time. After N iterations, the parameter of the current value network is copied to the target value network. The network parameter is updated by minimizing the mean-squared error between the current and target Q values. The introduction of the target network reduces the correlation between the current and target Q values to some extent, and improves the stability of the algorithm.

3) The reward values and error terms are reduced to a limited time, which ensures that the Q value and the gradient value are kept within a reasonable range, which improves the stability of the algorithm.

2) The Deep Strategy Gradient Method Based on an Actor-Critic Framework: The method of the continuation property of the action in the UAV's control task under the real scenario allowing for the online extraction of bulk trajectories cannot always reach a satisfactory coverage and may converge to a local minimum. Therefore, it is proposed to extend the actor-critic (AC) framework in the traditional

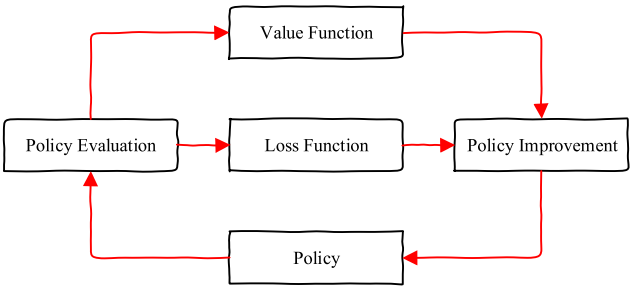

Fig. 6. Actor-Critic Framework.

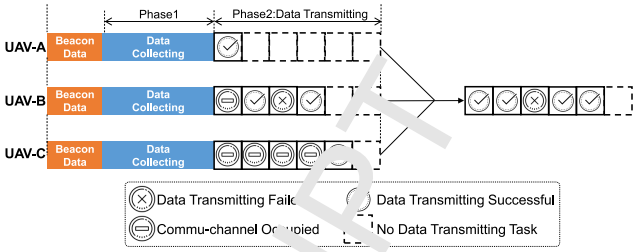

Fig. 7. Frame Sequence Example of Data Transmission Protocol

reinforcement learning (RL) to the deep strategy gradient method. As shown in Figure 6, this uses the learning structure of the deep strategy gradient method based on an AC framework.

The deep deterministic policy gradient (DDPG) algorithm based on an AC framework can be used to solve deep reinforcement learning (DRL) problems of continuous motion space. DDPG uses deep neural networks with the parameters θ^{μ} and θ^{Q} to represent the deterministic strategy $a = \pi(s|Q^{\mu})$ and the value function $Q(s, a|\theta^{Q})$, respectively. The strategy network is used to update the strategy, corresponding to the actor in the AC framework. The value network is used to approximate the value function of the state action pair and provide gradient information, corresponding to the critic in the AC framework. In DDPG, the target function is defined as a discounted reward sum: $J(\theta^{\mu}) = E[r_1 + \gamma r_2 + \gamma^2 r_3 + ...]$

According to the deterministic strategy $a = \pi(s|Q^{\mu})$, the gradient is:

$$\frac{\partial J(\theta^{\mu})}{\partial \theta^{\mu}} = E_s[\frac{\partial Q(s, a|\theta^{Q})}{\partial a} \frac{\partial \pi(s|\theta^{\mu})}{\partial \theta^{\mu}}] \quad (4)$$

The critic network is updated using the value network in DQN, with gradient:

$$\frac{\partial L(\theta^{Q})}{\partial \theta^{Q}} = E_{s,a,r,s' \sim D}[(y - Q(s, a|\theta^{Q}))] \frac{\partial Q(s, a|\theta^{Q})}{\partial \theta^{Q}} \quad (5)$$

Then, a random gradient descent is applied to make end-to-end optimization on the target function. DDPG uses an experience replay mechanism to obtain training samples from D and transmit the gradient information of the Q value function on the action to the actor network from the critic network. The parameters of the strategy network are updated along the direction of increasing Q value, according to Equation 5.

B. Description of UAV Task Scheduling

We now introduce the strategy to transmit UAV data under time division multiplexing. As shown in Figure 7, the UAV performs the sensing task in a synchronous and iterative manner. In the data transmission protocol, time is divided into discrete time periods and it takes data perception and data transmission as a cyclic unit, which consists of several frames. At the beginning of the cycle, each UAV determines its matching BS, relay UAV, data transmission power, and uplink data transmission channel, then transmits the information to the BS in the beacon frame through the control channel. The rest of the cycle is divided into a data perception phase and a data transmission phase.

1) UAV Data Perception Phase: Here, each UAV senses several frames and collects sensor data during this period. It is worth mentioning that, since the data processing capability is limited, the UAV may not be able to determine whether the sensor is effective. However, it can evaluate its performance based on the successful calculation of the sensor probability alone. To confirm whether the collected sensor data is effective, the UAV needs to transmit its sensor data to the BS in the subsequent transmission phase.

2) UAV Data Transmission Phase: The transmission phase consists of a certain number of frames where UAVs transmit the collected sensor data to BSs. If two UAVs use different sub-channels to collect their sensor data, there will be no interference between them, since the two channels are orthogonal. However, when two UAVs from the same (or different) unit attempt to transmit data in the same channel, they will suffer data transmission interference. In each frame of the transmission phase, a UAV may be in one of the following situations:

- **Communication-channel Occupied:** Under this situation, the BS cannot allocate the channel for the UAV, so it cannot transmit its collected sensor data to the BS. The UAV must wait for the frame that allocates the channel in order to transmit its collected sensor data.
- **Data Transmission Failed:** Under this situation, the BS allocates the channel for the UAV to transmit its sensor data. However, as the BS is at a low signal to noise ratio (SNR) and the transmitted data is not received successfully, the UAV must transmit the sensor data to the BS in the next available frame.
- **Data Transmission Successful:** Under this situation, the UAV is allocated the channel and successfully transmits its collected sensor data to the BS.
- **No Task:** Under this situation, the UAV remains idle without attempting to transmit uplink data and it has transmitted the collected sensor data successfully in the previous frames during the cycle.

As the uplink channel resources are normally scarce, there may be insufficient uplink channels to support transmitting their sensor data in each frame of the transmission phase. To solve this problem, the BS applies the centralized channel allocation mechanism to allocate the uplink channels to the UAVs. Alternatively, the UAVs can determine their uplink channels in a decentralized manner. In the case of centralized allocation mechanisms, in each frame the BS can allocate the uplink channels, which maximizes the total data transmission success rate. Under normal conditions, the number of UAVs competing for the same channel decreases over time in a cycle, since some UAVs may have completed data transmission in previous frames and remain idle for the rest frames of the transmission phase.

C. Task Channel Allocation Based on Deep Reinforcement Learning

As shown in Figure 7, in a UAV cluster, UAVs may interfere with by each other in the same unit if they are allocated to transmit sensor data in the same channel. Therefore, we optimize the channel allocation to improve the probability of success of the UAV uplink transmission. The optimization of channel allocation can finally be reflected in the optimization of data transmission delays, which are defined as Equation 6.

$$T = min \sum (\frac{d_{T_i q_m} x_{T_i q_m}}{v} + \frac{d_{T_i p_1} x_{T_i p_1}}{v} + ... + \frac{d_{T_i p_n} x_{T_i p_n}}{v} + t_m) \tag{6}$$

Including,
$\sum_q x_{T_i q} = 1$
$\frac{d_{T_i q} x_{T_i q}}{v} + \frac{d_{T_i p} x_{T_i p}}{v} + t_m \leq T_0$
$d_{pq} > d_{min}$
$\sum q_m + \sum p_j \leq N$
The corresponding mathematical symbols are defined as:
$x_{T_i q}$: If the q_mth UAV conducts the reconnaissance on the target Ti, the reconnaissance is equal to 1, otherwise 0.
$d_{T_i q_m}$: Refers to the distance between the q_mth UAV and the target Ti.

$$x_{T_i p_j} = \begin{cases} 0 & \min d_{T_i B_j} < 2 d_{max} \\ 1 & \text{else} \end{cases}$$

Here,$d_{T_i B_j}$ refers to the distance between T_i and B_i. When the distance between T_i and any base station is more than twice the maximum transmission distance of a single UAV, d_max, it sends one more UAV as a relay to ensure that the base station can receive the monitoring information from T_i.

Furthermore, it can apply the reinforcement learning method to allocate the channels in a decentralized way to achieve lower costs than centralized scheduling. Due to the huge size of the channel (100), there is a large state space. Therefore, we apply deep reinforcement learning to learn the schedule for channel allocation. In the deep reinforcement learning model, each UAV corresponds to an agent, whose action is to select a set of channels to transmit sensor data. As UAVs can only observe the situation of the channels when using them for transmission, we can define the state as the combination of the observations from several previous cycles. In each cycle, the input to the deep neural network is a combination of previous actions and states, and the output is the corresponding Q value. The UAV can select the best channel to transmit sensor data in the new cycle based on the obtained Q value. We now discuss the representation of UAV cluster scheduling constructed according to an AC algorithm.

The quantity $Q(s, a|^{\theta^Q}) = E(r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} ... | s_t = s, a_t = a, \pi)$represents the reward obtained after the UAV propagates, where the parameter is the deep neural network of θ^Q, representing the value function. The strategy function is $a = \pi(s|\theta^\mu)$, representing the cluster planning of the UAV. The evaluation function is $V(s, a) = \max \sum_i^n n_{chanel\_i} \times \min \sum_i^n d_{base_i}$, where $\max |V(s) - V'(s)| < \alpha$ at step $t = T+1$, d_{base_i} represents the distance between each UAV and the base station, and $n_{channel}$ represents whether the corresponding UAV has an available channel (with yes being 1 and no being 0). The loss function is $\arg\min_{a \in A} (T - T')$, and representing the change of the time delay after carrying out the new planning. The improvement function is $\pi'(s) = \arg\max_{a \in A} Q^\pi(s, a)$. After entering the initial state, the strategy function first plans the locations according to the UAV cluster planning algorithm. The evaluation function subsequently applies a gradient descent to update the weights of the deep neural network, the loss function calculates the time delay, and the value function calculates the reward value. Finally, the optimization parameter θ^Q of the value function is updated continuously in the deep neural network and the improvement function corrects the weights. In this way, after several iterations, the distribution of minimum time delays of the UAV clusters is obtained, thus reaching a stable state.

IV. UVA Cluster Task Scheduling

We will now demonstrate how to use reinforcement learning to schedule UAV cluster tasks. As shown in Figure 8, we consider that a cluster network consists of 2 BSs, 5 UAVs, and 3 target monitoring areas. Given that the two BSs have 2 channels to support uplink transmission of the UAVs and the 2 BSs, there is no band conflict. The accessible monitoring space of the UAV is defined as the cylindrical volume with the position of the UAV being the center of the circle and its maximum communication distance, $dmax$, being the radius. To effectively plan UAV tasks, we divide the feasible flight space into a group of discrete space points that represent a square area. When each UAV selects a neighboring space point in a single decision-making step to maximize the cumulative reward, it can correct the network structure to further maximize data transmission volume or optimize the transmission delay [26], [27], [28]. As shown in Figure 9, we compare the performance of data transmission with one UAV under different concurrent load, and the result show that the communication performance of one connection was better than that of 2 concurrent connections.

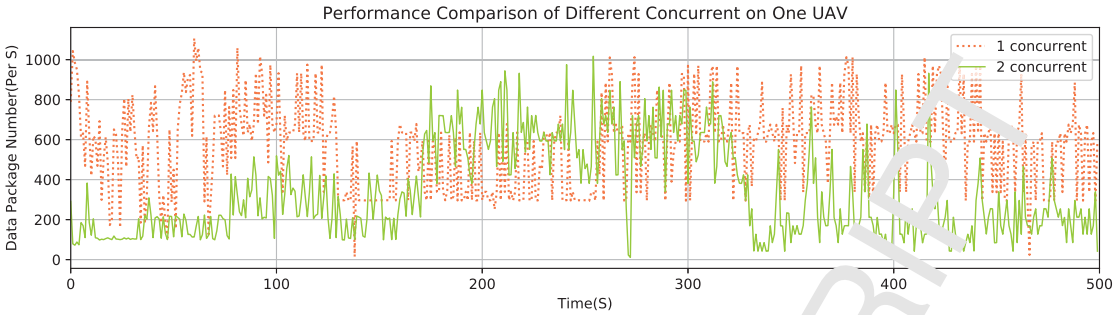

Fig. 9. Experiment Comparison

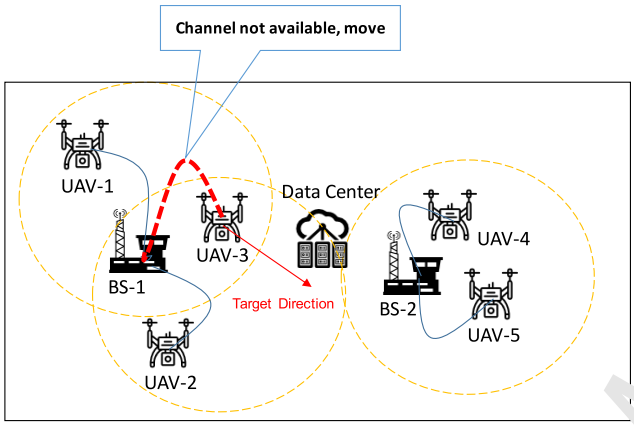

Fig. 8. Experiment Scene Graph

Reinforcement learning shows its application potential in the UAV cluster, but there are still many open problems to be solved, which may further promote the application and research of the UAV cluster. Some potential future research problems are listed below.

UAV Decision-Making Cooperation: When the UAV cluster needs to perceive real-time conditions of multiple tasks in the shortest time, each UAV may need to cooperate to perform those tasks. The centralized method has a high computational complexity, so it must distribute its scheduling to achieve task cooperation. Task cooperation is very challenging, because each UAV must consider tasks and possible decisions of other UAVs when choosing its own task and decisions. One promising approach is to use reinforcement learning algorithms to solve for the cooperating decisions.

UAV Cognitive Sensing. The large amount of sensor data generated by UAVs in UAV data sensing (for example: video streaming) can be a huge burden on traditional UAV cluster networks. To ensure the data transmission efficiency of the UAV, one could use cognitive video to enable UAVs to access the available channels of the UAV cluster in a timely manner. Using this approach, the priority mechanism of communication could be established for UAVs or applications, and the dynamic channel could be selected according to the determined priority. The channel selection could then be modeled through a reinforcement learning algorithm.

V. Conclusion

In this paper, we have described the UAV cluster and then proposed a networking scheme based on the expansion strategy. The UAV task schedules can be improved through autonomous learning, which can then make corresponding behavioral decisions and achieve autonomous behavioral control. We have used the method of reinforcement learning on the design of a UAV autonomous behavior decision-making strategy, and conducted experiments on UAV cluster task scheduling optimization in specific cases. We also discussed possible future research directions of the UAV cluster.

Acknowledgement

This paper is financially supported by King Saud University through the Vice Deanship of Research Chairs: Chair of Pervasive and Mobile Computing.

References

[1] A. Khan, B. Rinner, A. Cavallaro, "Multiscale observation of multiple moving targets using micro aerial vehicles," Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on, 2015, pp. 4642–4649.

[2] Y. Saleem, MH. Rehmani, S. Zeadally, "Integration of cognitive radio technology with unmanned aerial vehicles: issues, opportunities, and future research challenges," Journal of Network and Computer Applications, vol. 50, 2015, pp. 15–31.

[3] Z. Yong, Z. Rui, JL. Teng, "Wireless communications with unmanned aerial vehicles: opportunities and challenges," IEEE Communications Magazine, vol. 54, no. 5, 2016, pp. 36–42.

[4] Y. Zhang, M. Chen, N. Guizani, D. Wu, VCM. Leung, "SOVCAN: Safety-Oriented Vehicular Controller Area Network," IEEE Communications Magazine, vol. 55, no. 8, 2017, pp. 94–99.

[5] M. Chen, Y. Hao, "Task Offloading for Mobile Edge Computing in Software Defined Ultra-dense Network," IEEE Journal on Selected Areas in Communications, Vol. 36, No. 3, 2018, pp. 587-597.

[6] Y. Li, H. Lu, Y. Nakayama, et al., "Automatic road detection system for an airCland amphibious car drone," Future Generation Computer Systems, vol. 85, 2018, pp. 51-59.

[7] M. Chen, Y. Hao, H. Kai, L. Wang, L. Wang, "Disease Prediction by Machine Learning Over Big Data From Healthcare Communities," IEEE Access, 2017, pp. (99):1–1.

[8] W. Xiang, T. Huang, W. Wan, "Machine learning based optimization for vehicle-to-infrastructure communications," Future Generation Computer Systems, https://doi.org/10.1016/j.future.2018.10.047.

[9] R. Yu, J. Kang, X. Huang, et al., "MixGroup: Accumulative Pseudonym Exchanging for Location Privacy Enhancement in Vehicular Social Networks," IEEE Transactions on Dependable and Secure Computing, vol. 13, no. 1, 2016, pp. 93–105.
[10] M. Hossain, et al., "Big Data-Driven Service Composition Using Parallel Clustered Particle Swarm Optimization in Mobile Environment," IEEE Trans. Serv. Comput., vol. 9, no. 5, Aug. 2016, pp. 806–817.
[11] M. Chen, Y. Hao, L. Hu, et al., "Edge-CoCaCo: Towards Joint Optimization of Computation, Caching and Communication on Edge Cloud," IEEE Wireless Communications, Vol. 25, No. 3, 2018, pp. 21-27.
[12] E. A. Khalil, S. Ozdemir, S. Tosun, "Evolutionary task allocation in Internet of Things-based application domains," Future Generation Computer Systems, vol. 86, 2018, pp. 121-133.
[13] H. Lu, Y. Li, S. Mu, et al., "Motor Anomaly Detection for Unmanned Aerial Vehicles Using Reinforcement Learning," IEEE Internet of Things Journal, 2017, pp. (99):1–1.
[14] M. Chen, LT. Yang, T. Kwon, et al., "Itinerary Planning for Energy-Efficient Agent Communications in Wireless Sensor Networks," IEEE Transactions on Vehicular Technology, vol. 60, no. 7, 2011, pp. 3290–3299.
[15] M. Chen, YF. Qian, SW. Mao, et al., "Software-Defined Mobile Networks Security," Mobile Networks and Applications, vol. 21, no. 5, 2016, pp. 729–743.
[16] Y. Li, M. Chen, "Software-Defined Network Function Virtualization: A Survey," IEEE Access, vol. 3, 2015, pp. 2542–2553.
[17] M. Chen, et al., "Green and Mobility-aware Caching in 5G Networks", IEEE Trans. Wireless Communications, vol. 16, no. 12, 2017, pp. 8347-8361.
[18] A.S. Gomes, B. Sousa, D. Palma, et al., "Edge caching with mobility prediction in virtualized LTE mobile networks", Future Generation Computer Systems, vol. 70, 2017, pp. 148-162.
[19] M. Chen, V. Leung, "From Cloud-based Communications to Cognition-based Communications: A Computing Perspective", Computer Communications, Vol. 128, 2018, pp. 74-79.
[20] H. Wang, F. Xu, Y. Li, et al., "Understanding Mobile Traffic Patterns of Large Scale Cellular Towers in Urban Environment," Internet Measurement Conference, vol. 25, no. 2, 2015, pp. 225–238.
[21] M. Chen, Y. Hao, M. Qiu, et al., "Mobility-aware Caching and Computation Offloading in 5G Ultradense Cellular Networks," Sensors, Vol. 16, No. 7, 2016, pp. 974-987.
[22] Y. Li, F. Zheng, M. Chen , et al., "A unified control and optimization framework for dynamical service chaining in software-defined NFV system," Wireless Communications IEEE, vol. 22, no. 6, 2015, pp. 15–23.
[23] R. Roman, J. Lopez, M. Mambo, "Mobile edge computing, Fog et al.: A survey and analysis of security threats and challenges," Future Generation Computer Systems, vol. 78, 2016, pp. 680–698.
[24] M. Chen, Y. Hao, K. Lin, et al., "Label-less Learning for Traffic Control in an Edge Network," IEEE Network, vol. 32, No. 6, 2018, DOI: 10.1109/MNET.2018.1800110.
[25] C. Qiu, S. Cui, H. Yao, et al., "A novel QoS-enabled load scheduling algorithm based on reinforcement learning in software-defined energy internet," Future Generation Computer Systems, vol. 92, 2019, pp. 43-51.
[26] K. Hwang, M. Chen, "Big Data Analytics for Cloud/IoT and Cognitive Computing," Wiley, U.K., 2017, ISBN: 9781119247029.
[27] M. Chen, F. Herrera, K. Hwang, "Cognitive Computing: Architecture, Technologies and Intelligent Applications," IEEE Access, Vol. 6, 2018, pp. 19774-19783.
[28] P. Li, J. Li, Z. Huang, et al., "Multi-key privacy-preserving deep learning in cloud computing," Future Generation Computer Systems, vol. 74, 2017, pp. 76–88.

Jun Yang received Bachelor and Master degree in Software Engineering from Huazhong University of Science and Technology (HUST), China in 2008 and 2011 respectively. Then, he got his Ph.D degree at School of Computer Science and Technology, HUST, on June 2018. Currently, he works as a postdoctoral fellow at Embedded and Pervasive Computing (EPIC) Lab in School of Computer Science and Technology, HUST. His research interests include cognitive computing, software intelligence, Internet of Things, cloud computing and big data analytics etc.

Xinghui You received his B.S. degree in Huazhong University of Science and Technology (HUST), China in 2017. Currently, he is a Master student in School of Computer Science and Technology at HUST. His research includes 5G Mobile Communication System, edge computing, Internet of Things, and Software Defined Network, etc.

Gaoxiang Wu received Bachelor degree in Software Engineering from University of Electronic Science and Technology of China (UESTC) in June 2017. During the undergraduate study, he has won national inspiration scholarship for several times. In September 2016, he was recommended for direct admission to Huazhong University of Science and Technology without entrance examination. Currently, he is studying in Embedded and PervasIve Computing Lab at HUST. His research interests include congnitive system.

Mohammad Mehedi Hassan is currently an Associate Professor of Information Systems Department in the College of Computer and Information Sciences (CCIS), King Saud University (KSU), Riyadh, Saudi Arabia. He received his Ph.D. degree in Computer Engineering from Kyung Hee University, South Korea in February 2011. He received Best Journal Paper Award from IEEE Systems Journal in 2018. He received Excellence in Research Award from CCIS, KSU in 2015 and 23016 respectively. His research areas of interest are cloud computing, mobile cloud, sensor-cloud, Internet of things, Big data, and social network.

Ahmad Almogren received PhD degree in computer sciences from Southern Methodist University, Dallas, Texas, USA in 2002. Previously, he worked as an assistant professor of computer science and a member of the scientific council at Riyadh College of Technology. He also served as the dean of the college of computer and information sciences and the head of the council of academic accreditation at Al Yamamah University. Presently, he works as an associate professor and the vice dean for the development and quality at the college of computer and information sciences at King Saud University in Saudi Arabia. He has served as a guest editor for several computer journals. His research areas of interest include mobile and pervasive computing, computer security, sensor and cognitive network, and data consistency.

Dr. Joze Guna (Mr.) is an Assistant Professor at the Faculty of Electrical Engineering, University of Ljubljana. His area of research focuses on Internet technologies, multimedia technologies and IPTV systems with special emphasis on user centred design, user interaction modalities and designing the user experience, VR/AR/MR technologies, including gamification and flow aspects. Currently he is involved in a number of projects focusing on the development of intuitive user interfaces for elderly users of eHealth application and interactive multimedia HBBTV and VR/AR/MR applications. He is an expert in Internet, ICT and IPTV technologies and holds several industrial certificates from CISCO, Comptia and Apple, including trainer licenses from Cisco and Apple. He is a senior member of the IEEE organization and IEEE Slovenia Section Secretary General.

We utilize Reinforcement Learning algorithm to perform real-time tasks scheduling in UAV cluster.