Digital-Twin-Assisted Task Assignment in Multi-UAV Systems: A Deep Reinforcement Learning Approach

Xin Tang, Xiaohuan Li, Rong Yu, *Member, IEEE*, Yuan Wu, *Senior Member, IEEE*, Jin Ye, Fengzhu Tang, and Qian Chen

***Abstract*—Most existing multiunmanned aerial vehicle (multi-UAV) systems focus on fly path or energy consumption for task assignment, while little attention has been paid to the dynamic feature of the task, resulting in poor task completion ratio. The machine learning (ML) paradigm provides new methodologies for task assignment. However, ML methods are usually of heavy resource-consumption that cannot be directly applied in the UAV. In this article, a digital-twin (DT)-assisted task assignment approach is proposed to improve the resource-intensive utilization and the efficiency of deep reinforcement learning (DRL) in multi-UAV system. The approach has a three-layer network structure which can dynamically assign tasks based on the task time constraints. Moreover, the approach is divided into two stages of initial task-assignment and task-reassignment. In the first stage, airship divides a task into multiple subtasks according to the shortest distance based on genetic algorithm and assigns them to UAVs. In the second stage, the DT can be leveraged to enable the airships to learn from the features of tasks and to generate the *Q*-value of the estimated value network of DRL for UAVs via pretrain of DT. The *Q*-value can be directly applied for deep *Q*-learning network (DQN) in the UAVs to reduce the training episode. Furthermore, the DQN is adopted to train task-reassignment strategy. Simulation results indicate that the DQN with DT can significantly reduce the training episode, improving 30% of the task completion ratio and 19% of the system energy efficiency compared with that of the baseline methods.**

***Index Terms*—Deep reinforcement learning (DRL), digital twin (DT), multiunmanned aerial vehicle (multi-UAV) system, task assignment.**

I. Introduction

In recent years, due to the advantages of low cost, high flexibility, fast deployment, and mobile intelligence, multiunmanned aerial vehicle (multi-UAV) system has been widely used in a variety of scenarios, such as Internet of Things (IoT) applications, intelligent transportation, agricultural protection and communication relay [1], [2], [3], [4], [5]. When the services are gradually evolving and becoming more and more complex, a single UAV is unable to complete complex tasks due to its limited energy reservation and payload capacity. As the number of tasks increases, multi-UAV cooperation has become very important to improve task completion ratio. To this end, the use case of multi-UAV cooperation has attracted growing interests. Many related works focus on static task assignment strategies, which are made according to the initial requirements of the tasks. In [6], a particle swarm optimization algorithm is used to plan the task path of UAVs according to the task position given in advance, and it realizes the task execution in the shortest time. In [7], the planning algorithm is used to assign tasks according to the known task position in order to maximize the task revenue, and the genetic algorithm (GA) is applied to plan UAV path for minimizing energy consumption. Wang et al. [8] proposed a task allocation model for heterogeneous targets, which executes task assignment and path planning through the improved GA to minimize the task execution time and energy consumption. A coalition formation algorithm based on cooperative planning is proposed in [9], which considers the relationship between UAV energy consumption and task types to improve the rationality of task assignment. Moreover, Liu et al. [10] proposed a novel divide and conquer framework for multi-UAV task scheduling, in which a tabu-list-based simulated annealing algorithm is used to finish task allocation among multiple UAVs. In [11], a modified Wolf Pack algorithm together with a joint digraph-based method and meta-heuristic optimization method is used to solve the problem of multi-UAV cooperative task assignment. However, the above-mentioned researches are mostly

Manuscript received 19 October 2022; revised 5 January 2023 and 7 February 2023; accepted 25 March 2023. Date of publication 31 March 2023; date of current version 24 August 2023. This work was supported in part by the Guangxi Natural Science Foundation of China under Grant 2019GXNSFFA245007; in part by the National Natural Science Foundation of China under Grant U22A2054 and Grant 61971148; in part by the Guangxi Key Laboratory of Wireless Wideband Communication and Signal Processing under Grant GXKL06200103; in part by the National Key Research and Development Program of China under Grant 2020YFB1807802; in part by the Research Grant of University of Macau under Grant MYRG2020-00107-IOTSC; and in part by the Guangdong-Macau Joint Laboratory for Advanced and Intelligent Computing under Grant GDST2020B1212030003. *(Corresponding author: Xiaohuan Li.)*

Xin Tang, Xiaohuan Li, Fengzhu Tang, and Qian Chen are with the School of Information and Communication, Guilin University of Electronic Technology, Guilin 541004, China, and also with Guangxi Research Institute of Integrated Transportation Big Data, National Engineering Laboratory for Comprehensive Transportation Big Data Application Technology, Nanning 530001, China (e-mail: tangx@mails.guet.edu.cn; lxhguet@guet.edu.cn; tang_fz@126.com; chenqian@mails.guet.edu.cn).

Rong Yu is with the School of Automation, Guangdong University of Technology, Guangzhou 510006, China (e-mail: yurong@ieee.org).

Yuan Wu is with the State Key Laboratory of Internet of Things for Smart City and the Department of Computer and Information Science, University of Macau, Macau, China (e-mail: yuanwu@um.edu.mo).

Jin Ye is with the Guangxi Key Laboratory of Multimedia Communications and Network Technology, School of Computer, Electronics and Information, Guangxi University, Nanning 530004, China (e-mail: yejin@gxu.edu.cn).

Digital Object Identifier 10.1109/JIOT.2023.3263574

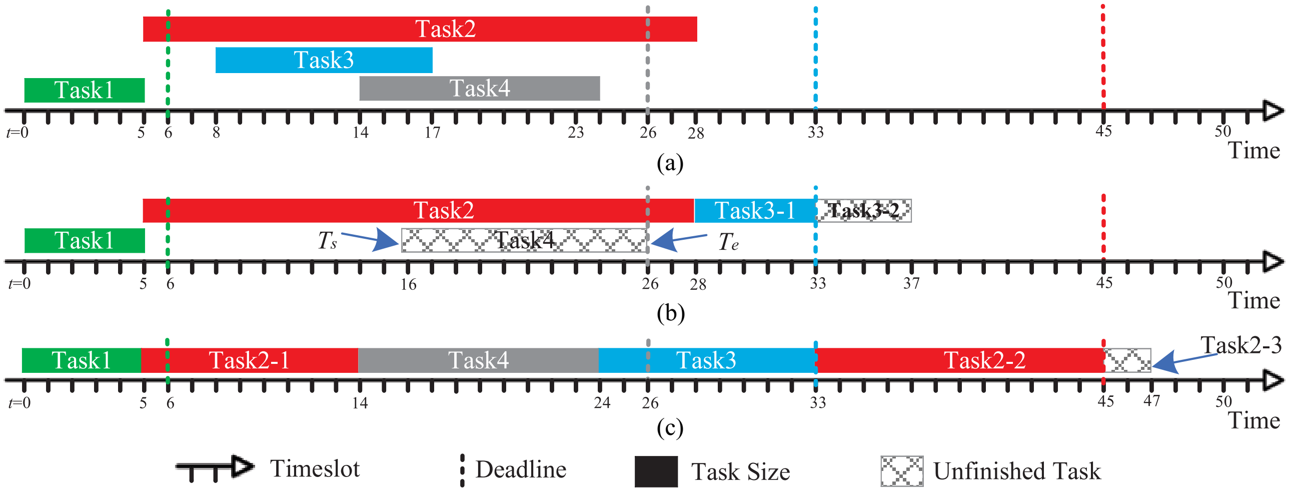


Fig. 1. Example of the comparison of different task assignment methods in a UAV, where the relationship of some time constraints of different task is described. (a) Tasks arriving randomly. (b) Arrival-time-considering task assignment. (c) Time-constrained-considering task assignment.

using static task assignment without considering dynamics during task execution, and thus it is difficulty to adjust execution behaviors or strategies according to the real-time state of UAVs and task. Moreover, most of tasks have strict deadlines to be completed, and an emerging task is considered to successfully executed only if it can be completed before its specified deadline. Such as, when the UAV arrives at the area where the task is executed, start time of task execution must be before end time. Otherwise, multi-UAV system has a low completion ratio of tasks.

To better introduce the research motivation of this article, an example in Fig. 1 is given below to illustrate the challenges that existing multi-UAV systems face when dealing with multiple tasks. We define task completion ratio as the ratio of the currently completed tasks to the total tasks in this article. Fig. 1(a) shows that the UAV receives four randomly arriving tasks within a period T includes 50 timeslots, and each task has its time constraints, such as the deadline or end time (i.e., T_e) and the latest start time (i.e., T_s). For simplicity, the size of each task can be represented by the number of timeslots. Fig. 1(b) shows the UAV executes tasks according to the sequence of the arrival time of the tasks (i.e., tasks that arrive first will be executed first). The result is that only task1 and task2 can be completed, and task3 is only executed for 5 timeslots. In addition, task4 has not been executed and UAV does not have enough timeslots to complete the task3. The task completion ratio is 50%. Fig. 1(c) shows the UAV dynamically adjusts the sequence of tasks according to the time constraints of tasks. A flexible method is to consider the latest start time of task in advance and assign different priorities to them, respectively. Then, task2 is divided into task2-1 and task2-2. The advantage is that task4, which is close to the deadline, will be executed after task2-1 and task3 will be executed after task4, so task1, task3, and task4 can all be executed by the UAV. The task completion ratio is 75%. It can also be seen from the above example that the task assignment in a single UAV is typically a dynamic process. And the real-time assignment of multitask will pose a great challenge to the static task assignment with the participation of multiple UAVs. Furthermore, lots of computation and storage resources of UAV are inevitably required as a guarantee for the implementation of dynamic task assignment. However, it is in contradiction with the limited resources of multi-UAV system. Therefore, a dynamic and high energy-efficiency task assignment approach is needed.

To overcome the challenges, we deploy deep Q-learning network (DQN) on UAVs to deal with highly dynamic task execution environments. At the same time, training an effective machine learning (ML) model is a complex and time-consuming process [12]. Moreover, the training will consume a lot of computation and storage resources as a cost, so the training period needs to be shortened. To this end, we consider to the training process of ML of UAVs on some edge devices in advance, such as airships. The advantage of this is as follows. On the one hand, the edge devices possess powerful computing capacities and storage resources, which can be used as an excellent resource supplement to the UAVs. On the other hand, the edge devices are capable to collect a large number of historical and environmental real-time state data. These data can provide comprehensive data input for the training process of ML, and the training results are often better than the training effect of a single UAV or a swarm of UAVs. Furthermore, the edge device sends the trained model parameters to multi-UAV for direct use or as the basis for its training, which solves the contradiction between the high consumption of resources of ML and the limited resources available. However, how to realize the above-mentioned model pretraining method so that the training results can be directly used by UAVs is a challenge. A novel approach is to create a avatar in virtual space for the UAV and train it synchronously. Fortunately, digital twin (DT) is a promising technology by creating virtual models of physical entities in the digital way and the models can understand the state of physical objects through collected data, so as to quickly and accurately predict and analyze the time-varying features.

In this article, we propose a DT-assisted multi-UAV system for achieving dynamic, fast and energy efficient task assignment. Different from the previous works, we leverage airships equipped with powerful computer to pretrain the DT model. The DT constructs a parallel virtualization of the physical multi-UAV system by mapping the system data to a virtual space. It enables the UAVs to learn the task assignment strategy via a pretrain model, which saves computing resources and reduces the delay of task assignment. Multi-UAV system collects the historical and environmental real-time state data by airships and UAVs for task assignment. These data empowered the decision-training process of the task assignment. Finally, a case study of DT assisted multi-UAV system for task assignment is investigated based on deep reinforcement learning (DRL). The aggregated task completion ratio, energy consumption and task priorities data are constructed as the training data set. Simulation results indicate that the proposed method can significantly increase the speed of model training, improving the task completion ratio and the system energy efficiency. The main contributions of this article are as follows.

1) A model of multi-UAV systems with DT assisted for task assignment is proposed. The DT model is built and stored in the airship which is used to enhance the efficiency of task assignment of UAV.
2) A three-layer network structure combining the advantages of hierarchy and distribution for task assignment in multi-UAV systems is designed. The impacts of flight distance and energy consumption on task completion ratio are also analyzed.
3) A method of task-reassignment for time-constrained tasks is proposed. The method can track the real-time features of tasks and UAVs by UAV-to-UAV (U2U), airship-to-airship (A2A), and UAV-to-airship (U2A), which enables the UAVs to quickly generate strategy via a pretrain approach at airship for increasing the ratio of task completion.

The remainder of this article is organized as follows. In Section II, a comprehensive survey of the related works is provided. In Section III, a model of multi-UAV systems with DT assisted task assignment is given and described. Section IV describes a approach of task-reassignment in multi-UAV system. Section V gives the experimental results and analysis. Finally, Section VI concludes this article.

II. Related Work

In this section, we review the related work about UAV dynamic task assignment, DRL-based techniques for UAV trajectory optimization and multi-UAV systems based on DRL and DT.

UAV dynamic task assignment has been an important topic in multi-UAV system in recent years. Some researchers have combined UAV with distributed architecture to satisfy the requirements of dynamic task assignment. Wang et al. [13] presented a two-layer optimization method for optimizing the deployment of UAVs and task scheduling, with the aim of minimizing system energy consumption by adaptively adjusting the number of UAVs. Moreover, to solve the problem of information coupling between task assignment and path planning of UAVs, a strategy based on the distributed architecture is proposed to improve the efficiency of UAV and re-evaluate the assigned tasks in [14]. Furthermore, in [15], an improved auction mechanism algorithm with the constraints of communication and endurance time is proposed to assign tasks. Li et al. [16] proposed a new multitask cooperated UAVs network framework and an AggreGate Flow-based scheduler in which ML is used to precisely estimate the task urgency-level, and improve efficiently the multitask completion rate. The above researches can carry out real-time task cooperation according to the features of UAV or task. However, the poor task completion caused by ignoring time constraints remains unsolved.

Trajectory optimization is a part that must be considered in the process of multi-UAV task assignment, and its performance is crucial to the completion of the task. Recently, DRL-based techniques for trajectory optimization are attractive to many researchers. Zhang et al. [17] studied to minimize the flying time of the rechargeable UAVs for completing the backscattering data collection task. They propose a single-agent deep option learning and a deep option learning base on hierarchical DRL and compared the proposed algorithms with different DRLs to prove their algorithms can achieve better performance. Challita et al. [18] focused on UAV path planning in the network of cellular-connected UAVs. They propose a DRL-based on echo state network cells algorithm for UAVs which learn its optimal path, transmission power, and cell association vector at different locations along its path. Furthermore, Han et al. [19] proposed a method of simultaneous target assignment and path planning based on a multiagent deep deterministic policy gradient algorithm, in which the system model is trained to solve target assignment and path planning simultaneously for deal with dynamic environments effectively and improve real-time performance. In addition, some researchers have considered the problem of obstacle avoidance in UAV trajectory optimization. Singla et al. [20] used a DRL-based method for UAV obstacle avoidance in unstructured and unknown indoor environments and proposed a deep recurrent Q-network with memory to learn the control policy. Ouahouah et al. [21] proposed a probabilistic and DRL-based algorithm, and they run on the top of the UAV or at a multiaccess edge computing that can gather data from UAVs sensors and then select the optional decision to avoid the obstacles. Although these researches achieve good performance in multi-UAV system, the training for the DRL model usually consumes a long time, which might not be acceptable to many latency-sensitive tasks.

The DT has been considered as a promising approach for addressing the above issue and attracted lots of interests in recent years [22]. The potential advantages of the DT has enabled the application of DT in various areas, such as real-time remote monitoring and control in intelligent transportation system,

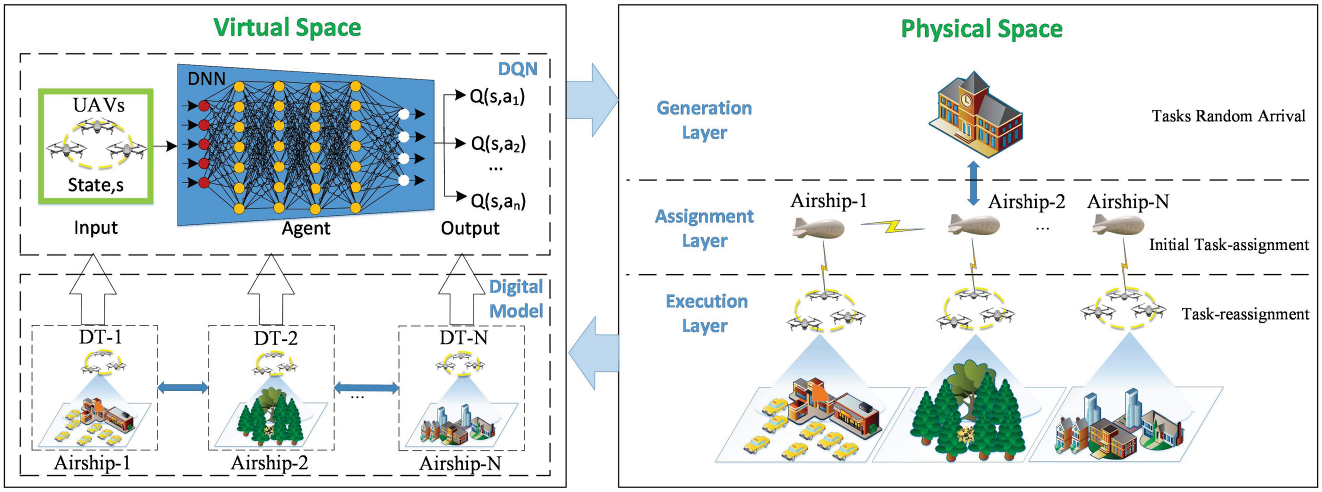


Fig. 2. System model.

testing and assessment in self-driving car, and the scheduling in intelligent industrial systems [23], [24]. Ji et al. [25] explored the effect of the DT in UAVs on providing medical resources quickly and accurately during COVID-19 prevention and control. Shen et al. [26] integrated DRL with DT to improve the practicality of flocking motion of multi-UAV systems in the real environment which unknown and stochastic. However, DT focuses on modeling an individual physical object in the virtual space, and a DT model always gathers and processes the objects state information in an independent mode without interacting with other models. As a result, the constructed object model may be not accurate, and both time and energy consumption of this construction process may be significant. Compared with a mapping between the physical object and its virtual twin, i.e., DT, the DT network uses advanced communication technologies to realize real-time information interaction between the physical object and its virtual twin, the virtual twin and other virtual twins, and the physical object and other physical objects. Zhang et al. [27] exploited the DT technology to map the edge caching system into virtual space, which facilitates constructing the social relation model. Dai et al. [28] proposed a paradigm DT network to establish model of network topology and the stochastic task arrival in Industrial IoT (IIoT) systems. Lu et al. [29] integrated DTs with edge networks and proposed the DT network with edge computing between physical edge networks and digital systems. In spit of its many application, the DT has not been leveraged for task assignment in multi-UAV cooperative systems.

Therefore, to meet different requirements of tasks in dynamic scenarios and improve the task completion ratio, this article proposes a three-layer network structure which can flexibly assign task. Moreover, a novel task-reassignment method is proposed to generate new priority of the tasks executed based on real-time state data, and makes decisions based on DRL with assistance from DT. Then, UAVs dynamically adjust the execution sequence of subtasks under time constraints and completes dynamic reassignment of multiple subtasks. The results indicate that the proposed method improves the task completion ratio in the scenario of issuing new tasks at any time.

III. System Model and Problem Statement

This article proposes a multi-UAV system assisted by DT that includes airships and mult-UAV systems, in which UAVs are assigned to execute time-constrained tasks. The model of the multi-UAV systems is described in Fig. 2. We consider that the airships in an area of several hundred square kilometers are deployed to work in the task assignment layer. The airships are used to receive real-time tasks from the generation layer. Furthermore, each airship is interconnected with corresponding groups of UAVs in the execution layer to execute some specific tasks. Specifically, we design the role of the airships to include two aspects. The first aspect is to realize the initial assignment of tasks based on the GA. The second aspect is to construct the DT model of the UAVs and collect historical and real-time task assignment. In this way, the airship pretrain DRL network model, and sends the trained network parameters to the UAVs. Correspondingly, the main work of the UAVs is responsible for the specific execution of the task, and receives the DRL network parameters from the corresponding airship, which is used to build a more accurate local training model and complete the task-reassignment. The advantage of our proposed model is that the local network training of the UAV can converge quickly, thereby achieving fast dynamic task-reassignment and reducing energy consumption. The process of task assignment of a multi-UAV system is divided into five steps.

Step 1 *(Task Generation and Release):* The control center generates tasks and randomly releases them to the assignment layer.

Step 2 *(Initial Task-Assignment):* Each task is divided into several subtasks and distributed to different UAVs. This step assigns the subtasks based on their positions and the number of the UAVs.

Step 3 *(Subtask Execution):* The UAVs receive and execute subtasks assigned. During the process of execution, UAVs interact with each other to obtain the real-time features of subtasks.

Step 4 *(Task-Reassignment):* Each UAV generates new subtask priority based on the subtask features. The subtask execution sequence of each UAV is adjusted to complete the task assignment again.

Step 5 *(Task Completion):* When all the subtasks are completed, the corresponding tasks are completed.

In order to realize the above five steps, in this article, the multi-UAV system is divided into the generation layer, the assignment layer and the execution layer. The generation layer generates and releases tasks from time to time according to the requirements of the control center. The assignment layer performs the initial task-assignment according to the global state of tasks and UAVs. The initial task-assignment divides the tasks into several subtasks and distributes them to the UAVs based on the shortest distance. The distance is obtained by GA. Furthermore, due to the limited computing power of UAV, the assignment layer runs as a computing agent for each UAV. It provides computing resources for UAV to dynamically adjust task priority under current task completion ratio and time constraints. The computing results are transmitted through the wireless connection between U2A. Moreover, to capture the features of dynamic tasks and to obtain accurate DT model of current tasks, in this article, the DT model of UAV is established from the perspective of task execution in the airship to update the estimated value network of DRL for UAVs in real time. Considering the influence of the information transmission delay between the airship and the UAV on the DT model, we designed a correction model (see Section IV-B). The model calculates the action of the UAV during the information transmission process between the airship and the UAV through the DQN algorithm, calibrates the DT model according to the completion of the task, and make the DT model accurate. The execution layer is responsible for executing subtasks and performing the task-reassignment. Each UAV executes task-reassignment algorithm to make decisions and dynamically adjust the sequence of subtasks, according to the real-time priority and ratio of completion of subtasks and the energy consumption of UAV, to improve the completion ratio of tasks. The relevant parameters of task assignment approach and its physical meaning are described in Table I.

The task set is represented by $z = \{z_1, z_2, \dots, z_{n_t}\}$, n_t represents the number of tasks. Along with the task, there are also related features of the task, and $W = \{P, A, T_s, T_e, \eta, P'\}$ represents the feature set of each task. Among them, the priority of the task z is represented by $P = \{P_1^z, P_2^z, \dots, P_{n_t}^z\}$, the execution area of the task z is represented by the set $A = \{A_1^z, A_2^z, \dots, A_{n_t}^z\}$, and the task start time and end time are represented, respectively, by the sets $T_s{}^z = \{T_{s1}^z, T_{s2}^z, \dots, T_{sn_t}^z\}$ and $T_e{}^z = \{T_{e1}^z, T_{e2}^z, \dots, T_{en_t}^z\}$.

TABLE I
LIST OF PARAMETERS

Notation	Explanation
z	Set of tasks
m	Set of subtasks
n_t	Number of tasks
n_u	Number of subtasks
n_p	Number of task positions
W	Feature set of task
g	Set of task positions
x_i	Abscissa value of two-dimensional plane of subtask i
y_i	Ordinate value of two-dimensional plane of subtask i
d^i	Distance of subtask m_i
D	Total distance of subtasks
v	Flight speed of UAV
ζ	State of executing subtask
P	Initial priority of subtasks
P'	Task-reassignment priority of subtasks
S	Size of subtasks
E_f	Energy consumption of UAV by flying
E_h	Energy consumption of UAV by hovering
E_w	Energy consumption of UAV by working
L_m	Position of subtask
L_n	Position of UAV
T_s	Start time to execute subtask
T_e	End time to execute subtask
A	Area of task
t	Timeslot of system
T	Periodic timeslot of system
η	Task completion ratio
E_u	Maximum energy consumption of UAV
G_{ij}	State of task position either executed or not
M_{ij}	State of subtask either executed or not
DT_n	Virtual model of UAV n
$\tilde{a}_n^t$	Action space of UAV n to execute tasks at time t
$\tilde{X}_n^t$	State space of UAV n to execute tasks at time t
τ	Communication delay between UAV and airship

When the assignment layer receives the task from the generation layer, task target positions are assigned in this task which are represented by the set $g = \{g_1, g_2, \dots, g_{n_p}\}$, n_p is the number of task target positions. The total number of UAVs in a multi-UAV system is n_u, and the UAVs are represented by the set $n = \{n_1, n_2, \dots, n_{n_u}\}$. It divides the task z into several subtasks m according to the number of UAVs n_u and the coordinates of the subtask target positions L_i. The subtasks are represented by the set $m = \{m_1, m_2, \dots, m_{n_u}\}$, where n_u represents the number of subtasks into which the task z is divided. Among them, subtask m_i includes n_ip target positions, and n_ip indicates that the i task contains n_p task target positions. When dividing tasks, it is necessary to achieve the goal of the shortest distance of all subtasks. In subtask m_i, the coordinate of target position g_i is $L_i = (x_i, y_i)$, and the coordinate of target position g_j is $L_j = (x_j, y_j)$.

A. Stage of Initial Task-Assignment

For initial task-assignment, we focus on the path planing of executing subtasks. The optimization objective is to minimize the flying distance of the UAVs. We transform the path planning problem into a GA optimal solution problem. Consider each solution as a chromosome, and then several chromosomes form a population. The population continues to evolve, and the chromosomes continue to mutate and inherit until the fitness function converges or reaches the maximum number of iterations. Specifically, the GA selects chromosomes through the combination of optimal retention strategy and roulette. It not only retains the best samples, but also improves the diversity of samples. And it performs the genetic operation of classifying and crossing the chromosomes according to the fitness function value during the crossover. At the same time, the crossover probability and mutation probability are dynamically adjusted according to the fitness function value of the current chromosome which speeds up the GA convergence. Hence, the fitness function can be modeled by

$$f_{\text{route}} = \frac{1}{d^i} \tag{1}$$

where the fitness function value is maximized, the shortest path of the UAV is determined. At the same time, the execution sequence of subtasks is also the position sequence of subtasks on the shortest path. The distance d^i of target positions of subtask m_i is set by

$$d^i = \sum_{i=1}^{n_ip}\sum_{j=1}^{n_jp} d_{ij}k_{ij} \tag{2}$$

where k_{ij} represents whether the target position g_i is connected to, if it is connected, $k_{ij} = 1$. If the task target position g_i and g_j are not connected, $k_{ij} = 0$. The Euclidean distance d_{ij} between the target position g_i and g_j is set by

$$d_{ij} = \sqrt{\left(x_i - x_j\right)^2 + \left(y_i - y_j\right)^2}. \tag{3}$$

Then, the distance of all subtasks in task z is defined as

$$D = \sum_{i=1}^{n_u} d^i. \tag{4}$$

In order to make the distance in each subtask relatively balanced, we find the variance of the distance which is given by

$$l^2 = \frac{\sum_{i=1}^{n_u}\left(d^i - \mathrm{AV}\right)^2}{n_u} \tag{5}$$

where AV is the average of the execution distance of the subtask, which is given by

$$\mathrm{AV} = \frac{D}{n_u}. \tag{6}$$

The objective function f_1 to divide tasks of the assignment layer can be expressed as

$$f_1 = \frac{1}{D} + \frac{1}{l^2}. \tag{7}$$

To simplify the analysis, the following constraints in the above process of task-assignment model are as follows.

1) Each task target position is executed by at least one UAV and at most one UAV

$$\sum_{j=1}^{n_u} G_{ij} = 1\left(\forall i = 1, 2, \ldots, n_p\right). \tag{8}$$

2) Each subtask is executed by at least one UAV and at most one UAV

$$\sum_{j=1}^{n_u} M_{ij} = 1, (\forall i = 1, 2, \ldots, n_u). \tag{9}$$

3) Each UAV can only execute one subtask of the same task at most

$$\sum_{i=1}^{n_u} M_{ij}^z = 1, (\forall z = 1, 2, \ldots, n_t \quad \forall j = 1, 2, \ldots, n_u). \tag{10}$$

4) Each UAV is assigned at least two subtasks

$$\left|\sum_{i=1}^{n_u} M_{ij}\right| \geq 2, (\forall j = 1, 2, \ldots, n_u). \tag{11}$$

5) Each UAV can only execute one subtask at the same time

$$\left|\sum_{i=1}^{n_u} M_{ij}{}^t\right| = 1, (\forall j = 1, 2, \ldots, n_u). \tag{12}$$

6) Different UAVs can execute subtasks from different tasks at the same time

$$\sum_{j=1}^{n_u} M_{ij}{}^t \geq 1, (\forall i = 1, 2, \ldots, n_u). \tag{13}$$

B. Stage of Task-Reassignment

In the stage of initial task-assignment, the sequence of subtasks forms the initial task execution list of the UAV by GA. The amount of subtask is represented by the set $S = \{S_1^1, S_2^1, \ldots, S_{n_u}^1, \ldots, S_1^{n_t}, S_2^{n_t}, \ldots, S_{n_u}^{n_t}\}$. After receiving a series of assigned subtasks, the UAVs in the execution layer will fly to the planned area for executing the subtasks in the initial execution sequence. As the task is executed, the task completion ratio gradually increases. A constant v is the speed of each UAV to execute tasks. Thus, in the time slot t, subtask m_i^z has been completed by UAV n_j which is defined as

$$S_{ij}^{zt} = vt\zeta_{ij}^z \tag{14}$$

where

$$\zeta_{ij}^z = \begin{cases} 1, \text{ UAV } n_j \text{ is executing subtask } m_i^z \\ 0, \text{ otherwise.} \end{cases} \tag{15}$$

Then, the completion ratio η_{ij}^{zt} of subtask m_i^z that is executed by UAV n_j in time slot t is defined as

$$\eta_{ij}^{zt} = \frac{S_{ij}^{zt}}{S_i^z}. \tag{16}$$

In the process of task execution, each UAV continuously exchanges the state information of their respective tasks. Meanwhile, every UAV updates the task completion ratio to quantify the task features based on the task completion ratio and task size in real time. Thus, at the time kT, the completion ratio of task z is defined as

$$\begin{aligned}\eta^z &= \sum_{i=1}^{n_u}\sum_{k=1}^{k} \eta_i^{zkT} \cdot \frac{S_i^z}{\sum_{i=1}^{n_u} S_i^z} \\ &= \sum_{i=1}^{n_u}\sum_{k=1}^{k}\sum_{j=1}^{n_u} \eta_{ij}^{zkT} \cdot \frac{S_i^z}{\sum_{i=1}^{n_u} S_i^z}.\end{aligned} \tag{17}$$

Thus, the ratio of total completion is as follows:

$$\begin{aligned}\eta &= \sum_{K=1}^{n_t}\left(\eta^z \cdot \frac{S_K}{\sum_{x=1}^{n_t} S_x}\right) \\ &= \sum_{K=1}^{n_t}\sum_{i=1}^{n_u}\sum_{k=1}^{k}\sum_{j=1}^{n_u} \frac{S_{ij}^{KkT}}{\sum_{x=1}^{n_t} S_x} \\ &= \sum_{K=1}^{n_t}\sum_{i=1}^{n_u}\sum_{k=1}^{k}\sum_{j=1}^{n_u} \frac{vkT\zeta_{ij}^K}{\sum_{x=1}^{n_t} S_x}\end{aligned} \tag{18}$$

where S_x is the size of task z.

Moreover, UAVs will generate a certain amount of energy consumption in the process of executing tasks. Thus, we use E_w to represent the energy consumption of UAV when executing tasks, E_f to represent the energy consumption of UAV when flying, and E_h to represent the energy consumption of UAV when hovering, which is given by in the time slot t

$$E_w = \xi_1 e \tag{19}$$

$$E_f = \xi_2 \gamma \tag{20}$$

$$E_h = \xi_3 h \tag{21}$$

where e, γ, and h are the average energy consumption during task execution, flying, and hovering, respectively. ξ_1, ξ_2, ξ_3 is the corresponding effective coefficient. When the UAV is executing, flying, and hovering, ξ_1, ξ_2, ξ_3 is equal to 1, respectively. Otherwise, it is equal to 0, respectively. In time slot t, the energy consumption E_j of UAV n_j is as follows:

$$E_j = E_w + E_f + E_h. \tag{22}$$

The total energy cost E of all UAVs is as follows:

$$E = \sum_{j=1}^{n_u} E_j. \tag{23}$$

As the generation layer issues new tasks from time to time, the execution list of subtasks is constantly updated. However, when the UAV executes tasks according to this list, the time constraints that are important for completing the tasks are ignored. Furthermore, UAVs is difficult to judge the urgency of task execution, resulting in the low completion ratio of all task. Therefore, for multiple task execution in dynamic scenarios, the initial task-assignment is no longer applicable. In order to improve the capabilities of the multi-UAV system when tasks are randomly assigned, and the ratio of task complement with time constraints. This article proposes a method of task-reassignment. The UAV can make decisions by DRL based on the real-time state of the task, and dynamically adjust the sequence of subtask execution. We consider a new task priority of subtask during the subtask execution is generated according to the time constraints and the task completion ratio at the current moment, and then continuously update it. When the remaining of the task is the largest, its new priority will be the highest. And the task execution sequence of each UAV is adjusted dynamically to realize task assignment considering time constraints and the new task priority, which can complete more tasks in the shortest time. We use P' to represent the new priority of the subtask m_i^z in the current slot t, and P' is defined as follows:

$$T_{ri}^{zt} = \frac{S_i^z - S_{ij}^{zt}}{v} \tag{24}$$

$$P' = \frac{T_{ri}^{zt}}{T_{ei}^z - t} \tag{25}$$

where S_i^z is the size of subtask m_i^z, and T_{ri}^z and T_{ei}^z are the remaining time and end time of subtask m_i^z, respectively. The larger the value of P', the earlier the task is executed.

Then, the optimization goal f_2 of the energy consumption efficiency of task-reassignment is defined as energy ratio, and the energy ratio is expressed as follows:

$$f_2 = \frac{P' + \eta}{E}. \tag{26}$$

In order to ensure the real-time performance of task-reassignment, we set up periodic interaction between UAVs to obtain the real-time task features of each other. The periodic T should be less than the time $T_{\min}$ required to complete the minimum task, and to avoid wasting computing resources, the T should be greater than or equal to half of the time required to complete the minimum task. Moreover, the E generated when each UAV executes tasks in the sequence of new tasks is less than the maximum energy consumption E_u of the UAV itself. The following constraints in the above model are as follows:

$$\frac{T_{\min}}{2} \le T < T_{\min} \tag{27}$$

$$E < E_u. \tag{28}$$

IV. DT and DQN-Based Task-Reassignment

Fig. 3 illustrates the main framework of the proposed DT and DQN-based task-reassignment of a multi-UAV system model construction approach. Airships execute all operations of data fusion, analysis and computation in the digital models. This approach not only increases the depth and breadth of the training model and the predicting accuracy, but also shortens data training period. Meanwhile, the DT model in airship remedies the limited computation and storage capabilities of small-sized UAVs. Thus, the performance of multi-UAV system can be effectively improved by concentrating UAVs on the operations of dynamic tasks. Besides, the historical and real-time date of the multi-UAV system both facilitates the optimization of network parameters and improves the efficiency of DQN. In the multi-UAV system, the DT model consists of four parts, the data collection part gets the parameters of UAVs through U2A. These parameters include UAVs position, subtask features and energy consumption. The state update part gets priority of tasks and completion ratio of tasks. As the virtual model of UAVs is built in the airships hovering in an area, the multiairship collaboration part is responsible for exchanging data between the airships and maintaining their DT model consistency. The model is builted and stored in the DT construction module in the airship, and periodically updated based on the data collected and state updated. The adjustment of the value networks of DQN in airship will be issued to the UAVs through the instruction output part, thereby changing the state sampling of UAVs and the estimated value network and the target value network. After establishing the DT, which offers a virtual representation of the physical UAVs, we need to extract some key features of task-reassignment and construct a multi-UAV system model.

A. DQN Model Construction on UAV

The essence of task-reassignment is the NP-hard problem. This problem is solved in a dynamic environment that tasks are released at any time, and the decision of the UAV is only related to the dynamic task state at the current moment. Furthermore, the dynamic Markov decision process can be

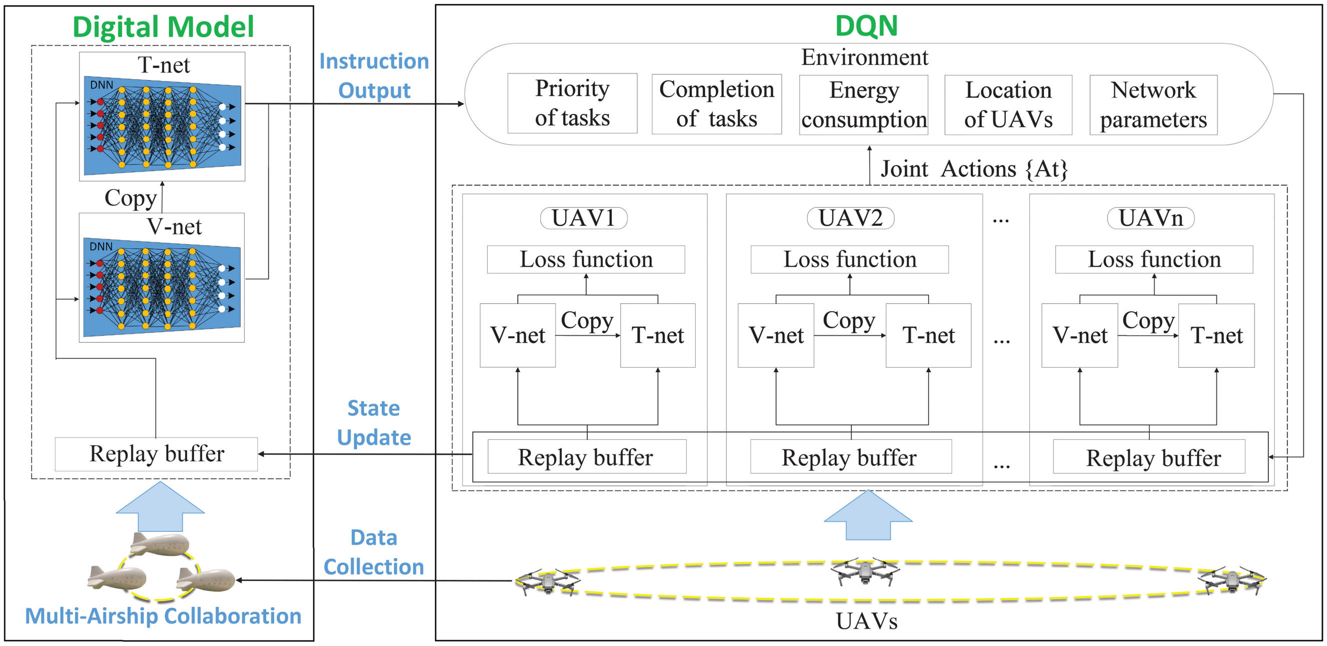


Fig. 3. DT and DQN-based task-reassignment of multi-UAV model construction.

used to solve task-reassignment problems. Therefore, this article chooses DQN with a dynamic Markov properties algorithm to solve the task-reassignment problem. The process of the DT assisted task-reassignment are shown in Algorithm 1. As an unsupervised learning method, the DQN continuously interacts with the dynamic environment through the agent, and then takes actions to obtain corresponding rewards. At the same time, it learns the laws of the environment until the optimal solution of the strategy is obtained [30], [31]. Each UAV is an agent in solving the task-reassignment problem for the multi-UAV system. In order to reduce the difficulty of training, each agent finishes training separately while sharing the environment to generate experience replay and network structure. However, since the agents collectively affect the environment, they can be represented by joint actions and states. We suppose there are n_u agents, and each agent can choose n_a actions, which are represented by $\{a_1, a_2, \ldots, a_{n_a}\}$. At the same time, each agent generates n_s states, which are represented by $\{s_1, s_2, \ldots, s_{n_s}\}$. The DQN parameters are set as follows.

Algorithm 1 DT Assisted Task-Reassignment of UAV Based on DRL.

```
Input: List of actions which are taken by each UAV(i.e. agent)
Output: Optimal sequences of actions to maximum the ratio of task
    completion
Initialize the memory replay
Receive and store $Q(s, a; \theta)$ issued by the airship
Update the Q-network in UAV according to (36)
begin
for episode $e = 1, \ldots, n$ do
    Initialize simulation environment
    Randomly generate and receive an initial state, including the
    coordinates, size, priority and time constrains of tasks, as well
    as the coordinates of multi-UAV
    Update $s$ for each UAV
    for time step $t = 1, \ldots, T$ do
        Calculate the ratio of task completion $\eta$, energy cost $E$,
        priority of subtask $P'$
        Select an action $a_t$ with $\varepsilon$, which means the UAV selects
        the task that it is currently executing
        Select $a_t = max_a Q(s_t, a; \theta)$ with $1 - \varepsilon$
        Calculate and observe $R_t$ and next state $s_t$
        Store $(s_t, a_t, R_t, s_{t+1})$ in the replay buffer
        Select $(s_{t'}, a_{t'}, R_{t'}, s_{t'+1})$ from the replay buffer
        Using SDG to train DQN model by the loss function (37)
        Update $\theta$, $Q(s_t, a; \theta)$, and $Q_{target}(s_t, a; \theta)$ according
        to (38)
    end for
    Store the Q-network
end for
end
```

Action Space: The action space of UAV n is the number of subtasks to be executed, $a_n^t = \{1, 2, 3, \ldots, l_n\}$ represents the action of UAV n in time slot t. l_n is the number of subtasks assigned to UAV n.

State Space: In our scenario, the decision process of multiple agents and produces a large amount of data. We preprocess the global state of the environment and eliminate the features irrelevant to the optimization objective of this article. The goal of the proposed method is to improve the completion ratio of all tasks and minimize the energy consumption. We define the features of the task as W(see Section III). After the initial-assignment, the task is divided into multiple subtasks. From the global state of the environment, each subtask is treated as a specific target position, so the task area A is transformed into the coordinate position L_m of the task. In addition, the real-time task priority P' can be obtained by parameters, such as the start time T_s and the end time T_e of the task, as well as the task completion ratio η during the executing task of the UAV. Then, the original task priority P and parameters, such as T_s and T_e are replaced by P'. In addition, the decision process of the UAV also includes its own coordinate L_n. Therefore, the states of multiple agents is represented by $X = \{\eta, P', L_M, L_N\}$. By the way, $X_n = \{\eta, P', L_m, L_n\}$ is the state of subtask m_i^z being executed by UAVn.

Reward Functions: The ultimate goal of task-reassignment is to improve the task completion ratio and reduce energy consumption. Thus, the reward settings are as follows:

$$R\big(s, a, s'\big) = \begin{cases} r_1 \\ r_2 \\ r_3 \end{cases} \tag{29}$$

where r_1 is the reward defined according to the new priority P' in the task execution process. Depending on the level of priority, different rewards are given. r_1 is given by

$$r_1 = \begin{cases} b_1, \ P' = \max\big(P'\big) \\ c_1, \ P' = \min\big(P'\big) \\ 0, \ \ \text{else} \end{cases} \tag{30}$$

where b_1 is a positive constant value, and c_1 is a negative value whose absolute value is smaller than b_1. When the priority of a task is the highest among all tasks in the UAV, r_1 equals to b_1. On the contrary, r_1 is equal to c_1. It means that the behavior of the UAV to execute tasks with high priority will be rewarded. Otherwise, the behavior of the UAV will not only not be rewarded ($r_1 = 0$), but will be punished ($r_1 = c_1$).

r_2 is a reward defined according to the task completion ratio η in the task execution process

$$r_2 = \begin{cases} b_2, \ \eta_m = \max(\eta_m) \\ c_2, \ \eta_m = \min(\eta_m) \\ 0, \ \ \text{else} \end{cases} \tag{31}$$

where b_2 is a positive constant value, and c_2 is a negative value whose absolute value is smaller than b_2. It means that the UAV executes a task with a high completion ratio, the behavior of UAV will receive a positive reward.

r_3 is a reward defined according to the energy consumption E during task execution

$$r_3 = -\rho E \tag{32}$$

where ρ is the reward coefficient of UAV energy consumption. The reward value r_3 is negative, which means that the behavior of the UAV is punished, so as to urge the UAV to reduce unnecessary energy consumption, such as reducing the consumption of hovering.

In order to maximize the of the task completion ratio and reduce the energy consumption of UAV, it is necessary to optimize the cumulative expected reward. R_t used as the reward parameter of DRL, which means that UAV is encouraged to execute tasks with high priority and high task completion ratio, while reducing unnecessary energy consumption. That is, the energy consumption is reduced while improving the task completion of the UAV system.

The reward R at time slot t is expressed as follows:

$$R_t = r_1 + r_2 + r_3. \tag{33}$$

Value Function: Each state is described by a certain value, so as to judge whether the state is good or bad. Then, the value function is used to quantify the cumulative reward of a state at time t, considering the discount factor γ. The value function is defined as

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}. \tag{34}$$

The function of action value directly measures each action in state s, and strategy π specifies a certain action a in state s. The cumulative reward obtained by performing the action a in state s is as follows:

$$Q_\pi(s, a) = E_\pi[G_t | S_t = s, A_t = a]. \tag{35}$$

The value iteration algorithm is used to find the Q-value, and then updates the value by selecting the largest Q-value in the next state s'. The update scheme is as follows:

$$Q(s, a) = Q(s, a) + \alpha\Big[R + \gamma \max_a Q\big(s', a\big) - Q(s, a)\Big] \tag{36}$$

where α presents the learning rate.

The optimization objective of DQN is to make the Q-value of the estimated value network to close to the Q-value of the target value network, which is as follows:

$$\text{Loss}(\theta) = E\left[\left(\left(R + \gamma \max_{a'} Q\big(s', a'; \theta\big)\right) - Q(s, a; \theta)\right)^2\right]. \tag{37}$$

The network parameters θ are updated by using the stochastic gradient descent (SGD), as follows:

$$\begin{aligned} \theta_{t+1} = \theta_t &+ \nabla Q(s, a; \theta) \\ &\times \alpha\big[R_{t+1} + \gamma \max\nolimits_{a'} Q\big(s', a'; \theta\big) - Q(s, a; \theta)\big]. \end{aligned} \tag{38}$$

B. DQN Model Construction Based on DT at Airship

High-fidelity DT models rely on real-time interactions between physical entities and virtual models. Therefore, by analyzing the transmission delay between the UAV and the airship, the action space and the state space are calibrated in real time which is to update the input data of the DQN in the airship, so as to ensure the validity of the pretraining results on the airship. For UAV n, its virtual model DT_n^t at time t can be expressed as

$$DT_n^t = \big\{\tilde{a}_n^t, \tilde{X}_n^t\big\} \tag{39}$$

where $\tilde{a}_n^t$ and $\tilde{X}_n^t$ represent the action space and the state space of UAV n to execute tasks, respectively.

When the UAV transmits the data of action and state space to the airship at time t, the delay τ can be measured at the airship. The calibrated DT model of the UAV n in the airship at time $t + \tau$ as follows:

$$DT_n^{t+\tau} = \big\{\tilde{a}_n^{t+\tau}, \tilde{X}_n^{t+\tau}\big\} \tag{40}$$

where $\tilde{a}_n^{t+\tau}$ and $\tilde{X}_n^{t+\tau}$ represents the calibrated action space and the state space of UAV n to execute tasks at time $t + \tau$, respectively. At time $t + \tau$, the airship receives the action and state space data of UAV n at time t. Furthermore, the airship needs to infer the action and state space data of the UAV at time $t + \tau$. Assuming that the UAV n is executing task m_i^z at time $t + \tau$, the remaining completion time of each task at time

t can be calculated by (24). Furthermore, the task completion size is calculated, the priority of the task in UAV n received in the airship is updated, and the correction of the action and state space data is further completed

$$P_i^{z'} = \begin{cases} \frac{T_{ri}^{zi} - \tau}{T_{ei}^{z} - (t+\tau)}, & \sigma = 0 \\ \frac{T_{ri}^{zt} - \left(\tau - \sum_{x=1}^{\sigma} \frac{S_x^z}{v}\right)}{T_{ei}^{z} - (t+\tau)}, & \sigma > 0 \end{cases} \tag{41}$$

where σ represents the number of completed tasks in the UAV within the time τ, and σ can be obtained according to the task priority and the remaining completion time of each task. If $\sigma = 0$, the priority of task m_i^z is updated in the state space. If $\sigma > 0$, removing the completed σ actions and their corresponding state space data in the action space, and updating the priority of task m_i^z executed by the UAV n in the state space at time $t + \tau$.

According to the analysis of the above-mentioned transmission delay and the calculation of the completion progress of the task, the priority of the task received by the airship is updated, which improves the fidelity of the DT model on the airship to a certain extent.

V. Performance Evaluation

In this section, we evaluate our proposed schemes performance in terms of convergence, transmission delay, task completion ratio, and energy ratio.

A. Simulation Setup

In order to verify the effectiveness of the method, we design a simulation map of 40 km * 40 km. Assuming that the control center randomly distributed new tasks. We deploy four multi-UAV systems in which each system has four UAVs connected by U2U. Each UAV executes the task at a uniform speed. Each task is sent to the assignment layer from time to time. The computing platform is Intel Xeon, E5-2660, the memory is 32 GB, and the GPU is NVIDIA GeForce RTX 2080. We use the TensorFlow deep learning framework to implement the neural network part of the DQN.

In general, the parameter configuration of ML model has a direct impact on the models performance [12]. In this article, the maximum number of timeslots in the synchronization period is 200 to run the pretraining model, and on these timeslots, the airships are used to receive real-time tasks from the generation layer. The pretrained network parameters are send by U2A to the UAVs. Meanwhile, the UAVs execute the specific the task, and receives the network parameters from the corresponding airship. $\varepsilon -$ greedy is used as the method of action selection strategy. When the probability is $1 - \varepsilon$, the agent selects the action corresponding to the maximum Q-value. If the probability is ε, an action is randomly selected, ε is 0.99, and the discount factor γ is 0.9. The DQN only needs to learn through states, actions and rewards. In order to relieve the correlation between the states, we construct an experience replay to store the sample data (s, a, r, s') which is generated during the agent training process, and then randomly select some samples for training. The algorithm proposed in this article is trained for 10 000 periods, of which one period is from the beginning to the end of the task execution. The following Table II is related parameters of simulation. We compare our proposed algorithm in the training of task-reassignment with the following schemes.

TABLE II
Simulation Parameters

Parameter	Value
Number of UAV n	4
Number of task z	4
Number of subtask m	16
Size of subtask S	200
Priority of subtask P	(1,16)
Timeslot t	1 s
Period T	(100,200) s
Position of task L	(26,18) km

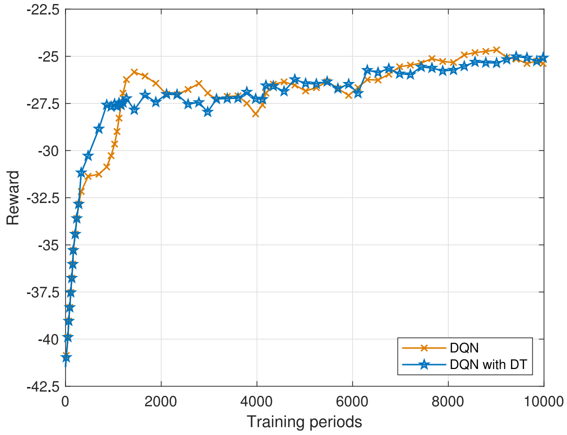


Fig. 4. Reward with the training periods.

1) *Greedy Algorithm (Greedy):* Greedy executes the task with the highest value and the earliest start time which focuses on the task with the greatest return at present and does not pay attention to the completion ratio of all tasks.
2) *GA:* GA is based on the shortest distance to select the sequence of task execution, and does not consider the time characteristics of dynamic tasks. In this way, it is easy to increase the hover time in the process of task execution.
3) *Optimal Selection Algorithm (Priority):* Priority is to preferentially select tasks with a small size of tasks for execution. The advantage is to improve the task completion ratio in terms of the number of tasks completed. However, it is also easy to lead to excessive hover waiting time, which increases energy consumption.
4) *Random Selection Algorithm (Random):* Random is not based on any features of the task, such as high value, short distance, and small size, but rather performs the task in a random way. The consequence of this is that the performance of the system is not prone to very good or bad results.

B. Convergence of DQN and DQN With DT

The cumulative reward value of the algorithm is shown in Fig. 4. The algorithms are in the exploratory stage during the

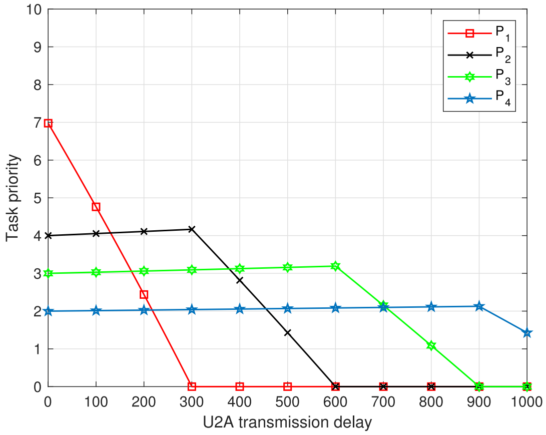


Fig. 5. Task priority with task time constraints and random arrival.

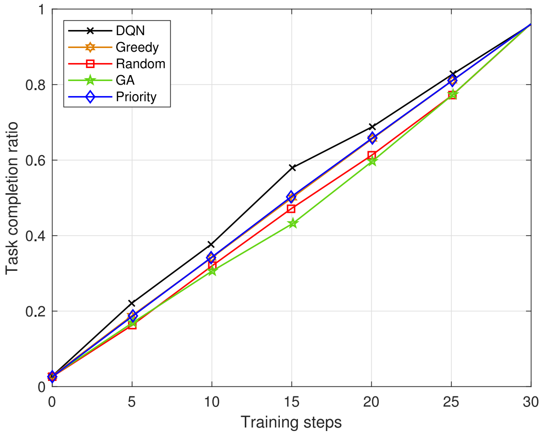


Fig. 6. Task completion ratio without task time constraints.

beginning of training. After the training is started, the cumulative reward value increases rapidly. Then, the algorithms will update the parameters regularly during the process. Thus, there are a slight fluctuation in the middle of the training. Finally, the DQN with DT shows a slight upward trend and stabilizes gradually, indicating that the strategy of dynamic task-reassignment with UAVs has been learned and continuously applied. However, the original DQN algorithm needs more than 2000 times to gradually stabilize, and the training period of the DQN with DT proposed in this article is shortened by 1000 times compared with the DQN, that is, the model training speed is increased by 50%, and the fluctuation of the reward value is lower than the DQN. This benefits from we leverage airship to pretrain the DT model. This model then can be directly applied for DQN in the UAVs to reduce the training episode. Therefore, combining DT for the model training of the DQN can increase the speed of training greatly.

C. Transmission Delay Analyze

Fig. 5 shows the variation of task priority with transmission delay and the time-varying priority of four tasks executed by a UAV in a synchronization period. At the beginning of the period, task1 with the highest priority is executed first, and the priority of task1 will decrease. At the same time, with the increase of time τ, the deadline of each task gradually approaches, and the priority of other unexecuted tasks will gradually increase. The experimental results also verify the effectiveness of (25). At $\tau = 300$, task1 is completed. According to the task priority at the beginning of the period, the UAV will continue to execute task2, and so on for other tasks. It should be noted that when $\tau = 0$, the UAV takes the priority of each task as state space data and uploads it to the airship. After the transmission delay τ, the airship receives the state space data. If the airship directly uses the state space data as the input for the construction of the DT model of the UAV, it will cause the DT model in the airship to be different from the real UAV data by a period of time τ. Moreover, the experimental results also show that if τ is less than the remaining completion time of a task (e.i. $\sigma = 0$), then the airship only adjusts the priority of the currently executed task. For example, if $\tau = 100$, the state space data received by the airship indicates that the priority of task1 is 7, and the priority of task1 tends to decrease with the execution of the task. In order to reflect the changes of the real state space data on the airship, this article proposes the DQN optimization approach base on DT (see Section IV-B). According to (41), it can be estimated that after the delay τ, the true priority of task1 is about 4.7, indicating that the task priority input to the DT has a 32 % error, such poor input data is not conducive to DT-based DQN model pretraining effect. The experimental results further illustrate that the transmission delay analysis model and priority calibration method of the DT model proposed in this article can correct the priority of tasks at $\tau > 0$. Then, the original priority at $\tau = 0$ will be replaced by the estimated priority at $\tau > 0$, so that the data in the state space of the airship can reflect the changes in the state space of the actual UAV in real time, which can effectively improve the fidelity in the construction of the DT model.

D. Task Completion Ratio Comparison

As can be seen from Fig. 6, with the change of training steps, the task completion ratio of each algorithm increases steadily, and can complete all tasks. But in the process of task execution, the task completion ratio of DQN is higher than others. It shows that at the same time, the DQN can dynamically adjust the order of task execution according to the task completion ratio and priority.

As shown in Fig. 7, the DQN can achieve the maximum task completion ratio. At the beginning of the execution, the task completion ratio of the DQN is the same as that of the greedy algorithm, and it is slightly lower than that of the greedy algorithm. The reason for this is that in the early stage of the execution, the essence of the greedy algorithm determines that the UAV chooses to execute the task with the earliest start time at the current time step. But it also ignores the task completion time, resulting in the low completion of the late task. On the contrary, the DQN considers the time constraints in the process of task execution, and makes task-reassignment. Therefore, compared with other algorithms, the DQN can achieve the highest ratio of task completion. In Fig. 8, the task completion ratio shows a slight downward trend when the generation layer

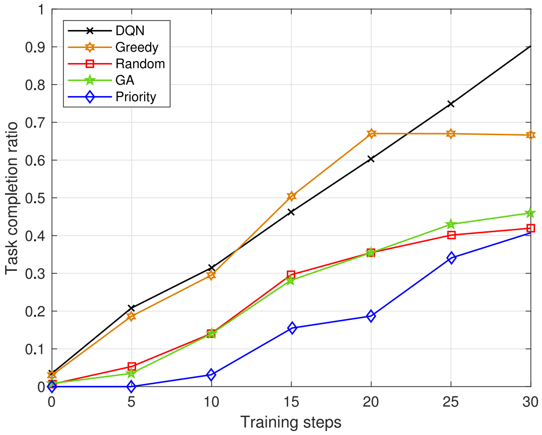


Fig. 7. Task completion ratio with task time constraints.

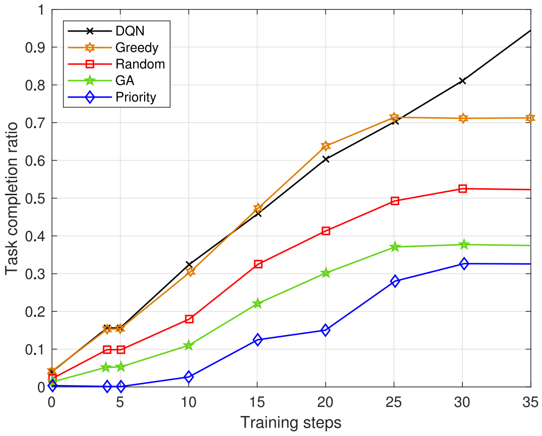


Fig. 8. Task completion ratio with task time constraints and random arrival.

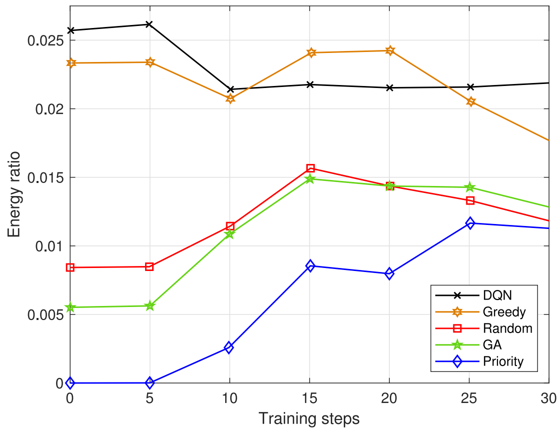


Fig. 9. Energy ratio with task time constraints.

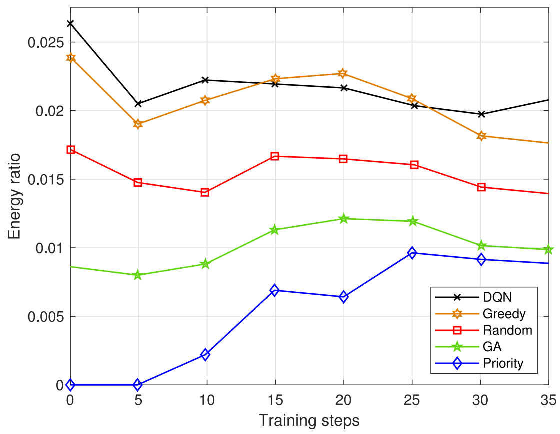


Fig. 10. Energy ratio with task time constraints and random arrival.

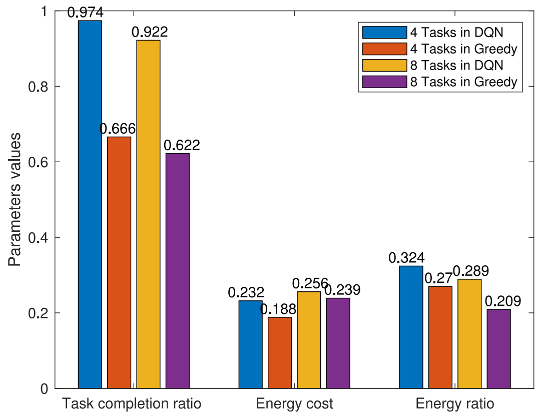


Fig. 11. Comparison of parameters with different number of tasks.

releases a new task. However, with the continuous implementation of the task, the completion of the task gradually returns to the upward trend. The DQN focuses on the task completion time with considering the time constraints. Therefore, the task completion ratio of the DQN is still significantly higher than other algorithms, which indicates that the DQN can improve task completion ratio of the random arrival tasks with time constraints.

E. Energy Ratio Comparison

As can be seen from Fig. 9, the energy ratio of the DQN is the highest, followed by greedy algorithm. It can be seen from Fig. 7 that the task completion ratio of the algorithm in the mid-term phase of execution is lower than that achieved by the greedy algorithm. Thus, the energy ratio in the mid-term phase is lower than that of the greedy algorithm. But due to the dynamic decision-making ability, in the end, the DQN can achieve the highest energy ratio. In Fig. 10, when new tasks randomly arrive, the energy ratio of each algorithm decreases in the beginning and increases gradually. It can be seen from Fig. 8 that the task completion ratio of the DQN in the mid-term phase is lower than that of the greedy algorithm, so its energy ratio is lower than that of the greedy algorithm in the mid-term phase. Because the DQN focuses on the dynamic decision-making, the energy ratio of the DQN is higher than that of the greedy algorithm in the end. Therefore, the DQN in this article can achieve the highest energy ratio.

Fig. 11 compares the parameters of algorithms under time constraints when the number of tasks is different. The abscissa represents different parameters, and the ordinate represents the value of the parameters. Among them, for the convenience of comparison, task completion ratio is the real value, and system energy consumption and energy ratio are the normalized values. When the number of tasks increases gradually, the overall task completion ratio of the multi-UAV cooperation implemented by the proposed method decreases, the energy consumption increases, and the energy ratio decrease. Obviously, the performances of the DQN are higher than the greedy algorithm. It shows that with the increase of the number of tasks, the DQN can still dynamically process tasks under time constraints, 30% task completion and 19% system energy efficiency are improved.

VI. Conclusion

In order to solve the poor ratio of task completion caused by time constraints of existing multi-UAV systems, this article proposed a DT and DRL-based task assignment method in multi-UAV systems. Moreover, we proposed a novel task assignment method which includes the initial task-assignment and the task-reassignment. In the initial task-assignment, according to the task area, the task is divided into subtasks and distributed to the UAV by using GA. Next, in the task-reassignment, the behavior decision of UAV is made by DRL based on DT to achieve task-reassignment and improve the task completion subject to the time constraints. Simulation results show that the training period of the DQN with DT proposed in this article is shortened. Furthermore, compared with other algorithms, the task completion ratio and the energy ratio can be improved in the scenario of task delivery under stringent time constraints. For future work, we will design a novel multiagent DRL-based algorithm to assign task properly in the multi-UAV system. And the integration model of communication and computing for U2A and U2U should be considered to improve the latency performance of moving edge devices and minimize the overall costs of the multi-UAV systems.

References

[1] C.-H. Zhou et al., "Deep reinforcement learning for delay-oriented IoT task scheduling in SAGIN," *IEEE Trans. Wireless Commun.*, vol. 20, no. 2, pp. 9111–925, Feb. 2021.

[2] H.-X. Peng and X.-M. Shen, "Multi-agent reinforcement learning based resource management in MEC and UAV-assisted vehicular networks," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 1, pp. 131–141, Jan. 2021.

[3] Y.-J. Zheng, Y.-C. Du, H.-F. Ling, W.-G. Sheng, and S.-Y. Chen, "Evolutionary collaborative human-UAV search for escaped criminals," *IEEE Trans. Evol. Comput.*, vol. 24, no. 2, pp. 217–231, Apr. 2020.

[4] Z. Ning et al., "5G-enabled UAV-to-community offloading: Joint trajectory design and task scheduling," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 11, pp. 3306–3320, Nov. 2021, doi: 10.1109/JSAC.2021.3088663.

[5] C.-H. Pan, H. Ren, Y.-S. Deng, M. Elkashlan, and A. Nallanathan, "Joint blocklength and location optimization for URLLC-enabled UAV relay systems," *IEEE Commun. Lett.*, vol. 23, no. 3, pp. 498–501, Mar. 2019.

[6] K.-A. Ghamry, M.-A. Kamel, and Y.-M. Zhang, "Multiple UAVs in forest fire fighting mission using particle swarm optimization," in *Proc. Int. Conf. Unmanned Aircraft Syst. (ICUAS)*, Miami, FL, USA, Jan. 2017, pp. 1404–1409.

[7] Z.-Y. Zhou et al., "When mobile crowd sensing meets UAV: Energy-efficient task assignment and route planning," *IEEE Trans. Commun.*, vol. 66, no. 11, pp. 5526–5538, Nov. 2018.

[8] Z. Wang, L. Liu, T. Long, and Y.-L. Wen, "Multi-UAV reconnaissance task allocation for heterogeneous targets using an opposition-based genetic algorithm with bouble-chromosome encoding," *Chin. J. Aeronaut.*, vol. 31, no. 2, pp. 339–350, Feb. 2018.

[9] H.-Y. Luan et al., "Energy efficient task cooperation for multi-UAV networks: A coalition formation game approach," *IEEE Access*, vol. 8, pp. 149372–149384, 2020.

[10] H. Liu et al., "An iterative two-phase optimization method based on divide and conquer framework for integrated scheduling of multiple UAVs," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 9, pp. 5926–5938, Sep. 2021.

[11] Y. Chen, D. Yang, and J. Yu, "Multi-UAV task assignment with parameter and time-sensitive uncertainties using modified two-part wolf pack search algorithm," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 54, no. 6, pp. 2853–2872, Dec. 2018.

[12] Y. Li and S. Abdallah, "On hyperparameter optimization of machine learning algorithms: Theory and practice," *Neurocomputing*, vol. 415, pp. 295–316, Nov. 2020.

[13] Y. Wang, Z.-Y. Ru, K. Wang, and P.-Q. Huang, "Joint deployment and task scheduling optimization for large-scale mobile users in multi-UAV-enabled mobile edge computing," *IEEE Trans. Cybern.*, vol. 50, no. 9, pp. 3984–3997, Sep. 2020.

[14] W.-R. Yao, N.-M. Qi, N. Wan, and Y.-B. Liu, "An iterative strategy for task assignment and path planning of distributed multiple unmanned aerial vehicles," *Aerosp. Sci. Technol.*, vol. 86, no. 6, pp. 455–464, Mar. 2019.

[15] C. Chen, W.-D. Bao, T. Men, X.-M. Zhu, J. Wang, and R. Wang, "NECTAR-an agent-based dynamic task allocation algorithm in the UAV swarm," *IEEE Trans. Mobile Comput.*, vol. 6, no. 6, pp. 55291–55301, Sep. 2020.

[16] X.-H. Li et al., "An aggregate flow based scheduler in multi-task cooperated UAVs network," *Chin. J. Aeronaut.*, vol. 33, no. 11, pp. 2989–2998, Nov. 2020.

[17] Y. Zhang, Z.-Y. Mou, F. Gao, L. Xing, J. Jiang, and Z. Han, "Hierarchical deep reinforcement learning for backscattering data collection with multiple UAVs," *IEEE Internet Things J.*, vol. 8, no. 5, pp. 3786–3800, Mar. 2021.

[18] U. Challita, W. Saad, and C. Bettstetter, "Interference management for cellular-connected UAVs: A deep reinforcement learning approach," *IEEE Trans. Wireless Commun.*, vol. 18, no. 4, pp. 2125–2140, Apr. 2019.

[19] Q. Han, D.-X. Shi, T.-L. Shen, X.-H. Xu, Y. Li, and L.-J. Wang, "Joint optimization of multi-UAV target assignment and path planning based on multi-agent reinforcement learning," *IEEE Access*, vol. 7, pp. 146264–146272, 2019.

[20] A. Singla, S. Padakandla, and S. Bhatnagar, "Memory-based deep reinforcement learning for obstacle avoidance in UAV with limited environment knowledge," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 1, pp. 107–118, Jan. 2021.

[21] S. Ouahouah, M. Bagaa, J. Prados-Garzon, and T. Taleb, "Deep reinforcement learning based collision avoidance in UAV environment," *IEEE Internet Things J.*, vol. 9, no. 6, pp. 4015–4030, Mar. 2022.

[22] B. Fan, Y. Wu, Z. He, Y. Chen, T. Q. S. Quek, and C.-Z. Xu, "Digital twin empowered mobile edge computing for intelligent vehicular lane-changing," *IEEE Netw. Mag.*, vol. 35, no. 6, pp. 194–201, Nov./Dec. 2021.

[23] Y.-W. Wu, K. Zhang, and Y. Zhang, "Digital twin networks: A survey," *IEEE Internet Things J.*, vol. 8, no. 18, pp. 13789–13804, Sep. 2021.

[24] X. Shen, J. Gao, W. Wu, M. Li, C. Zhou, and W. Zhuang, "Holistic network virtualization and pervasive network intelligence for 6G," *IEEE Commun. Surveys Tuts.*, vol. 24, no. 1, pp. 1–30, 1st Quart., 2021.

[25] G. Ji, J.-G. Hao, J.-L. Gao, and C.-Z. Lu, "Digital twin modeling method for individual combat quadrotor UAV," in *Proc. IEEE 1st Int. Conf. Digit. Twins Parallel Intell. (DTPI)*, Sep. 2021, pp. 1–4.

[26] G. Shen et al., "Deep reinforcement learning for flocking motion of multi-UAV systems: Learn from a digital twin," *IEEE Internet Things J.*, vol. 9, no. 13, pp. 11141–11153, Jul. 2022.

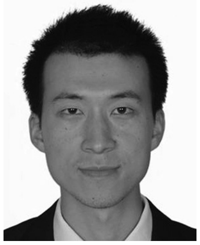
[27] K. Zhang, J. Cao, S. Maharjan, and Y. Zhang, "Digital twin empowered content caching in social-aware vehicular edge networks," *IEEE Trans. Comput. Social Syst.*, vol. 9, no. 1, pp. 239–251, Feb. 2022.

[28] Y.-Y. Dai, K. Zhang, S. Maharjan, and Y. Zhang, "Deep reinforcement learning for stochastic computation offloading in digital twin networks," *IEEE Trans. Ind. Informat.*, vol. 17, no. 7, pp. 4968–4977, Jul. 2021.

[29] Y.-L. Lu, X.-H. Huang, K. Zhang, S. Maharjan, and Y. Zhang, "Communication-efficient federated learning and permissioned blockchain for digital twin edge networks," *IEEE Internet Things J.*, vol. 8, no. 4, pp. 2276–2288, Feb. 2021.

[30] E.-M. Vartiainen, Y. Ino, R. Shimano, K.-G. Makoto, Y.-P. Svirko, and K.-E. Peiponen, "Numerical phase correction method for terahertz time-domain reflection spectroscopy," *J. Appl. Phys.*, vol. 96, no. 8, pp. 4171–4175, Oct. 2004.

[31] R.-S. Sutton and A.-G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: MIT Press, 2018.



Xin Tang received the B.S. and M.S. degrees from Guilin University of Electronic Technology, Guilin, Guangxi, China, in 2011 and 2015, respectively, where he is currently pursuing the Ph.D. degree in information and communication engineering.

He has been working with the China Mobile Communications Corporation Guangxi Branch, Guilin, since 2015. In 2016, he joined the Institute of Information Technology, Guilin University of Electronic Technology, where he is a Full-Time Lecturer. He is currently an Engineer with the Guangxi Research Institute of Integrated Transportation Big Data, National Engineering Laboratory for Comprehensive Transportation Big Data Application Technology, Nanning, China. His current research interests include multiagent system, vehicular networks, and wireless communications and networking.

Xiaohuan Li received the B.S. and M.S. degrees from Guilin University of Electronic Technology, Guilin, Guangxi, China, in 2006 and 2009, respectively, and the Ph.D. degree from the South China University of Technology, Guangzhou, Guangdong, China, in 2015.

He was a Visiting Scholar with the Université de Nantes, Nantes, France, in 2014. He is currently a Professor with the School of Information and Communication, Guilin University of Electronic Technology and a Research Fellow with the National Engineering Laboratory of Application Technology of Integrated Transportation Big Data, Beihang University, Beijing, China. His current research interests include wireless sensor networks, vehicular networks, UAV networks, and cognitive radios.

Rong Yu (Member, IEEE) received the B.S. degree in communication engineering from Beijing University of Posts and Telecommunications, Beijing, China, in 2002, and the Ph.D. degree in electronic engineering from Tsinghua University, Beijing, in 2007.

After that, he worked with the School of Electronic and Information Engineering, South China University of Technology, Guangzhou, China. In 2010, he joined the School of Automation, Guangdong University of Technology, Guangzhou, where he is currently a Professor. His research interests mainly focus on wireless networking and mobile computing, such as edge computing, federated learning, blockchain, digital twin, connected vehicles, and smart grid.

Prof. Yu received the Best Paper Awards from International Conferences, including IEEE Blockchain 2022 and IEEE ICCC 2016. He was a member of the Home Networking Standard Committee, China, where he led the standardization work of three standards.

Yuan Wu (Senior Member, IEEE) received the Ph.D. degree in electronic and computer engineering from Hong Kong University of Science and Technology, Hong Kong, China, in 2010.

He was a Visiting Scholar with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada, from 2016 to 2017. He is currently an Associate Professor with the State Key Laboratory of Internet of Things for Smart City, University of Macau, Macau, China, and also with the Department of Computer and Information Science, University of Macau. His research interests include resource management for wireless networks, green communications and computing, edge computing and edge intelligence, and energy informatics.

Dr. Wu received the Best Paper Award from the IEEE ICC'2016, WCSP'2016, IEEE TCGCC'2017, and IWCMC'2021. He is currently on the editorial board of the IEEE Transactions on Vehicular Technology, IEEE Transactions on Network Science and Engineering, and IEEE Internet of Things Journal.

Jin Ye received the Ph.D. degree with School of Science and Engineering, Central South University, Changsha, Hunan, China, in 2008.

She is currently a Professor with the School of Computer, Electronics and Information, Guangxi University, Nanning, Guangxi, China. She worked as a Visiting Scholar with the Department of Computer Science and Engineering, University of Minnesota, Twin Cities, Minneapolis, MN, USA, in 2018. Her current research interests include network protocol design, data center networks.

Prof. Ye is also the member of China Computer Federation.

Fengzhu Tang received the B.S. degree in communication engineering from Sichuan Normal University, Chengdu, Sichuan, China, in 2018, and the M.S. degree in information and communication engineering from Guilin University of Electronic Technology, Guilin, China, in 2021.

Her research interests mainly focus on wireless networks, edge computing, and edge intelligence.

Qian Chen received the B.S. and M.S. degrees from Guilin University of Electronic Technology, Guilin, Guangxi, China, in 2007 and 2012, respectively.

From 2007 to 2015, she joined the Institute of Information Technology, Guilin University of Electronic Technology, where she is a Full-Time Lecturer. Since 2016, she has been working with Guilin University of Electronic Technology, as a Senior Engineer with the School of Information and Communication and with the Guangxi Research Institute of Integrated Transportation Big Data, National Engineering Laboratory for Comprehensive Transportation Big Data Application Technology, Nanning, China. Her current research interests include air–ground-integrated networks, vehicular networks, and Internet of Things system.