A Multi-UAV Cooperative Task Scheduling in Dynamic Environments: Throughput Maximization

Liang Zhao, *Member, IEEE,* Shuo Li, Zhiyuan Tan, Ammar Hawbani, Stelios Timotheou, Keping Yu

Abstract—Unmanned aerial vehicle (UAV) has been considered a promising technology for advancing terrestrial mobile computing in the dynamic environment. In this research field, throughput, the number of completed tasks and latency are critical evaluation indicators used to measure the efficiency of UAVs in existing studies. In this paper, we transform these metrics to a single optimization objective, i.e., throughput maximization. To maximize the throughput, we consider realizing this goal in two respects. The first is to adapt the formation of the UAVs to provide cooperative computing service in a dynamic environment, we integrate a policy-based gradient algorithm and the task factorization network as a new reinforcement learning algorithm to improve the cooperation of UAVs. The second is to optimize the association process between UAVs and users, where the heterogeneity of tasks is considered. This algorithm is modified from the Gale-Shapley stability concept to optimize the appropriate association between tasks and UAVs in a dynamic time-varying condition to get the near-optimal association with few iterations. The scheduling of dependent tasks and independent tasks jointly also has to be considered. Finally, simulation results demonstrate the improvement of cooperation performance and the practicability of the association process.

Index Terms—Throughput Maximization, Multi-UAV Cooperation, Task Scheduling, Reinforcement Learning.

1 Introduction

The hlyellowuse of Unmanned aerial vehicles(UAV) to provide computation service arises significant concerns [1]. Compared with traditional terrestrial edge computing(EC), UAVs-assisted computation can break the constraint of topographic limitation to fly to the uncovered area by the edge servers to provide service, and the lack of the computation capacity of edge servers can be compensated by UAVs [2], [3]. Existing studies of UAVs-assisted computation are mainly divided into two categories. One prefers to study the trajectory or deployment optimization to adjust the location of UAVs to provide better service, and another prefers to optimize the association policy, such as the scheduling policy or the combination policy to maximize the computation efficiency of UAVs while satisfying the demand of tasks.

In the first category, by optimizing the flight trajectories or the deployment policy, UAVs can fly on more energy-efficient trajectories or deploy in more suitable locations while providing computation service. Existing studies focus on optimizing the cooperative trajectory or cooperative deployment policy [4]–[8]. However, there are still some challenges to solve. For example, one UAV cannot observe the whole environment due to the limited coverage area, and the environments of existing studies are assumed to be static. These solutions cannot be applied to practice directly. For the training process of the cooperation model, existing studies only combine the observation from each UAV, this makes the algorithm unable to converge stable with the increasing number of UAVs.

The second category ignores the cooperative flight of UAVs but focuses on the association process optimization, as well as optimizing the resource allocation policy, task scheduling policy and other metrics to improve the completed ratio or the completed latency of tasks [9]–[14]. Reasonable combination policy and task scheduling policy between tasks and UAVs all can reduce the process latency of tasks and the energy consumption of UAV. However, existing studies only consider a static environment, the details of tasks are ignored, and the set of tasks are unchanged. In practice, tasks generated by mobile devices (MDs) generally have a strong randomness. Although we believe that the arrival of a task follows the Poisson distribution, the sudden creation, withdrawal, and details of many tasks cannot be predicted in a complex environment. Thus, we need to design a converge-quickly and converge-stable algorithm, and it is insensitive to the change of tasks.

To sum up, many previous studies contribute to the optimization of throughput, latency and other metrics. However, some problems still have not been solved. For example, the dynamic of the environment has not been considered, the global information of MDs and tasks is assumed to

- *Liang Zhao, Shuo Li and Ammar Hawbani are with the School of Computer Science, Shenyang Aerospace University, Shenyang 110136, China; Liang Zhao is also with the Shenzhen Institute for Advanced Study, University of Electronic Science and Technology of China, Shenzhen 518110, China. (e-mail: lzhao@sau.edu.cn, lishuo1@stu.sau.edu.cn, anmande@ustc.edu.cn).*
- *Zhiyuan Tan is with the School of Computing, Engineering and the Built Environment, Edinburgh Napier University, Edinburgh EH10 5DT, Scotland, UK. E-mail:z.tan@napier.ac.uk.*
- *Stelios Timotheou is with the KIOS Research and Innovation Center of Excellence and the Department of Electrical and Computer Engineering, University of Cyprus, 1678 Nicosia, Cyprus (e-mail: stimo@ucy.ac.cy).*
- *Keping Yu is with the Computer Science Department, Community College, King Saud University, Riyadh 11437, Saudi Arabia, and with the Graduate School of Science and Engineering, Hosei University, Tokyo 184-8584, Japan (email: KepingYu@ksu.edu.sa; keping.yu@ieee.org).*
- *Ammar Hawbani and Stelios Timotheou are the corresponding authors.*

be known, the heterogeneity of tasks has been ignored, the association process is too complex. These shortages all make the solutions cannot be applied in practice. To solve these problems, we propose a UAVs-assisted terrestrial computing framework to cope with differentiated tasks in a dynamic environment. To improve the cooperation of UAVs, we especially optimize the global optimal action selection process to guarantee cooperative performance. In addition, we propose an association algorithm between UAVs and MDs, called Many-to-One Gale-Shapley(MOGS), which is improved by the Gale-Shapley algorithm. This algorithm can realize direct association optimization without the help of a third party such as the edge server. Thus, the communication latency can be reduced significantly. Finally, the throughput can be improved further. Our contributions are summarized as follows.

- *Throughput Maximization Problem Formulation*: A UAVs-aided offloading system in a continuous dynamic environment has been formulated, with the constraint of tolerant latency of tasks generated by terrestrial MDs and the limitation observation of UAVs. The locations of MDs change all the time in this environment and the stochastic generation of tasks has no rules to follow, while the volume, generation time slot, tolerant latency are also. The UAVs can serve terrestrial MDs with limited coverage and computation capacity. The objective is to maximize the throughput, maximize the completed number of tasks and minimize the completed latency of tasks, which is shown to be a non-convex problem. To solve this problem, UAVs must find a cooperative deployment location to maximize the transmission efficiency for air-to-terrestrial and inner communication and cover maximized MDs. These metrics are used to measure the cooperation performance of UAVs.
- *DRL-based Multi-UAV Cooperation Policy*: To solve the optimization problem mentioned above, the relationship between latency, the number of completed tasks, and another metric throughput has been analyzed. Then, the optimization has been transformed into maximizing the throughput. This optimization problem has been solved by two solutions, one is to optimize the trajectory and cooperation of UAVs, another solution is to optimize the association between UAVs and MDs to process more tasks as soon as possible. To optimize the deployment and cooperation of UAVs, a deep reinforcement learning algorithm, i.e., proximal policy optimization (PPO) has been adopted. To improve convergence speed and optimal action selection policy between UAVs, a novel action-value function factorization approach has been combined with PPO.
- *GS-based Tasks Scheduling Algorithm*: An improved association algorithm, Many-to-One Gale-Shapley (MOGS), has been proposed to realize fast task scheduling in a complex environment in continuous time. It is inspired by the Gale-Shapley (GS) algorithm to realize the association between UAVs and tasks directly without the assistance of a third party, i.e., some edge servers or central servers. It consumes very little computation power, and the constraint of the number of two sides in GS is broken. Some rules of association are proposed to optimize the computation load of each UAV to guarantee the performance of cooperation. Some simulation comparisons demonstrate the advantage of MOGS.

The organization of this paper later is as follows. Section 2 introduces some related studies in recent years. The system model and problem formulation are described in Section 3 and Section 4, respectively. In Section 5, the solution of the problem is introduced. In Section 6, we introduce the simulation environment and present the results. Finally, the whole work in this paper is concluded.

2 Related Work

In this section, we will review existing studies, which include UAV-centric studies and task-centric studies. Also, we briefly summarize the shortages of them to demonstrate the motivation of this work.

UAV-centric studies mainly research how to optimize the trajectory, cooperation policy and other metrics to improve the efficiency of UAV [4]–[8]. In these studies, the experience of users usually has not been considered in detail. The authors Zhang *et.al* use the DRL algorithm to plan the cooperative trajectory of UAVs-BSs to guarantee the throughput maximization of users in an emergency environment [4]. Guan *et.al* [5] use the PPO algorithm and K-means algorithm to plan the trajectory of UAV while minimizing the interaction consumption and improving the deployment efficiency. Furthermore, Table 1 summarizes the comparison between our study and previous studies [15]–[24]. For task-centric research, focus on improving the association process between UAVs and tasks, such as optimizing the task collecting policy, task scheduling policy, etc [2], [9]–[14]. These studies can improve the QoE of users in the considered environment. Some studies consider using UAVs to provide offloading service for devices [10] [11], where the Wang *et.al* use Generative Adversarial Networks (GANs) and the gradient-based policy to train a policy for online scheduling with partial observation [10]. Some studies focus on optimizing the energy-efficiency ratio and some other metrics to optimize these two shortages. Wang *et. al* and Hua *et. al* in [13] and [14] all maximize the throughput by optimizing the trajectory of UAVs and offloading decisions.

The studies mentioned above all contribute to optimizing UAV-assisted MEC. However, they only consider optimizing some metrics for a static time slot but ignore the continuity of time in practice. In a complex environment, a huge number of tasks will be generated and canceled in a continuous time irregularly, and the location of some users will also change. How to train a cooperation model to adapt to the dynamic environment while guaranteeing offloading efficiency needs to be solved. There still are some problems to be solved in the task-centric research. For example, during a training process, the task set generated at the beginning of a one-time slot may change and the final result cannot guarantee optimality. The training process of the result also consumes a period, the latency-sensitive task may not be served due to the low-latency constraint. Thus, a lightweight

TABLE 1
A comparison with existing studies

Reference	Scheme	Environment		Observation		Objectives			
		Static	Dynamic	Local	Complete	Latency	Throughput	Task Number	Joint
[15]	A*	✓			✓	✓			
[16]	CO	✓			✓		✓		
[17]	DRL	✓			✓		✓		
[18]	DRL		✓		✓		✓		
[19]	DRL	✓		✓			✓		
[20]	DRL	✓			✓	✓			
[21]	DRL	✓			✓	✓			
[22]	DRL		✓		✓	✓			
[23]	CO	✓			✓			✓	
[24]	BCD	✓			✓				✓
Our work	DRL,MOGS		✓	✓					✓

and fast convergence task scheduling algorithm needs to be designed.

3 System Model

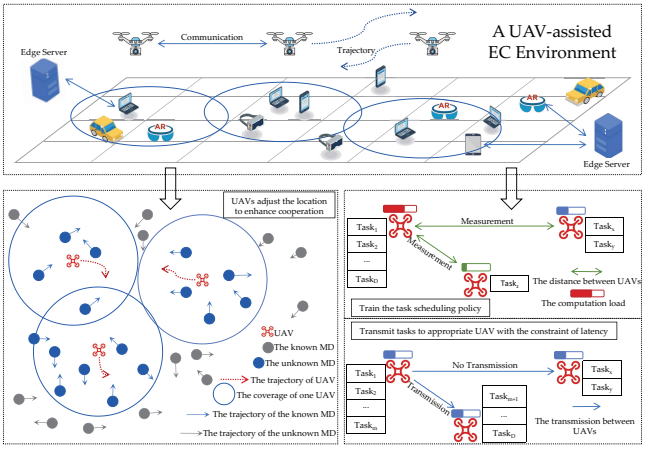


Fig. 1. An illustration of a UAV-assisted computation environment, the upper part represents the real world. The lower left and lower right parts represent the training of the UAV cooperation model and the task scheduling model for this environment, respectively.

In this work, we consider using multiple UAVs to serve moving MDs, which is shown in Fig. 1, and the number of MDs is much higher than the number of UAVs. MDs are moving all the time and tasks are generated by them. UAVs need to move cooperatively to cover MDs and compute tasks. The system model includes three parts, i.e., the environment model, transmission model and energy consumption model. Then we introduce these models in detail.

3.1 Environment Model

This model is used to describe some fundamental characters including the state of UAV and the details of tasks. We use $\mathcal{U} = \{u_1, ..., u_k, ..., U\}$ to denote the set of UAVs. The UAVs in $\mathcal{U}$ can communicate with each other to transmit tasks, results, and the topology information of the whole swarm. We consider a time horizon $\mathcal{T}$ in time interval $[\mathcal{T}_s, \mathcal{T}_e]$ and discretize it into T equal-size time slots by the length of t_i, which are indexed by the set $\mathcal{T} = \{t_1, t_2, ..., T\}$. There are $\mathcal{M}_{t_i} = \{\tau_{d_1,t_i}, ..., \tau_{d_j,t_i}, ..., \tau_{d_D,t_i}\}$ tasks generated by MDs $\mathcal{D} = \{d_1, d_2, ..., D\}$ in the t_i time slot.

The main properties of the UAV u_k are denoted as $u_k \triangleq << \chi^{lo}_{u_k,t_i}, \chi^{max}_{u_k} >, < \varphi^{coor}_{u_k,x,t_i}, \varphi^{coor}_{u_k,y,t_i}, \varphi^{coor}_{u_k,z,t_i} >>$, where $\varphi^{coor}_{u_k,t_i}$ denote the 3D coordination in the t_i time slot of the UAV u_k , $\chi^{lo}_{u_k,t_i}$ and $\chi^{max}_{u_k}$ denote the real-time computation load condition in the time slot t_i and the maximum computation capacity of the UAV u_k, respectively. We use $d_j \triangleq << \psi^{coor}_{d_j,x,t_i}, \psi^{coor}_{d_j,y,t_i} >, \tau_{j,t_i} >$ to describe the properties of one MD, where τ_{j,t_i} is a task generated by this MD, $\psi^{coor}_{d_j,x,t_i}$ and $\psi^{coor}_{d_j,y,t_i}$ are the 2D coordination of this MD in the t_i time slot. The main properties of the task τ_{j,t_i} are denoted as $\tau_{d_j,t_i} \triangleq < \upsilon_{\tau_{d_j,t_i}}, \gamma_{\tau_{d_j,t_i}}, \omega_{\tau_{d_j,t_i}}, \sigma_{\tau_{d_j,t_i}}, < \psi^{coor}_{d_j,x,t_i}, \psi^{coor}_{d_j,y,t_i} >>$, where $\upsilon_{\tau_{d_j,t_i}}$ denotes the volume of task τ_{d_j,t_i}, $\gamma_{\tau_{d_j,t_i}}$, $\omega_{\tau_{d_j,t_i}}$ and $\sigma_{\tau_{d_j,t_i}}$ denote the size of result data, the number of CPU cycles to process and the tolerant latency of task τ_{d_j,t_i}, respectively. We suppose that the coordination of MD and UAVs is static during a one-time slot, the distance between d_j and UAV u_k can be represented by (1). The expression of the notations in this paper are listed in Table.2.

$$\delta^{dis}_{d_j,u_k,t_i} = [(\varphi^{coor}_{u_k,x,t_i} - \psi^{coor}_{d_j,x,t_i})^2 + (\varphi^{coor}_{u_k,y,t_i} - \psi^{coor}_{d_j,y,t_i})^2 + (\varphi^{coor}_{u_k,z,t_i})^2]^{\frac{1}{2}} \quad (1)$$

3.2 Transmission Model

The transmission model is used to describe some details of the communication process, such as the calculation of transmission rate, etc. We use $P^{trans}_{u_k}$ to denote the transmitting power of the UAV u_k. Thus, the radius of the coverage area $R^{rad}_{u_k}$ of each UAV u_k is limited, which means the MD cannot connect to the UAV if the distance between MD and UAV surpasses $R^{rad}_{u_k}$. The uplink model is similar to the ground-to-air link, we need to consider the LoS and NLoS transmissions. Then we use a two-piece function $\zeta(\delta)$ to model the path loss [25], it is shown in (2), where the A^L and A^{NL} are the path losses with the reference distance $\delta = 1$,

TABLE 2
MAIN NOTATIONS USED IN SYSTEM MODEL

Notations	Expression
$\mathcal{U}$	The set of UAVs
$\mathcal{S}$	The set of ESs
$\mathcal{M}_{t_i}$	The set of tasks generated in t_i time slot
$\mathcal{D}$	The set of MDs
$\chi_{u_k}^{max}$	The computation capacity of the UAV u_k
φ_{u_k,t_i}^{coor}	The 3D coordination in the t_i time slot of UAV u_k
χ_{u_k,t_i}^{lo}	The real-time computation load condition in time slot t_i of the UAV u_k
$< \psi_{d_j,x,t_i}^{coor}, \psi_{d_j,y,t_i}^{coor} >$	The 2D coordinate of the MD d_j in the time slot t_i
τ_{d_j,t_i}	The task generated by MD d_j in the time slot t_i
$\upsilon_{\tau_{d_j,t_i}}, \gamma_{\tau_{d_j,t_i}}$	The volume, size of result data of task τ_{d_j,t_i}
$\omega_{\tau_{d_j,t_i}}, \sigma_{\tau_{d_j,t_i}}$	The number of CPU cycles to process and the tolerant latency
$R_{u_k}^{rad}$	The coverage area radius UAV u_k
$r_{d_j,u_k}^{up}, r_{u_k,d_j}^{down}, \xi, B$	The achievable upload and download rate of MD d_j to UAV u_k, the Signal-to-Noise Ratio and the bandwidth of the current channel
Ω^{u_k}	The throughput of the UAV u_k
$\Gamma_{u_k,d_j}, E_{u_k}^{trans}$	The latency and energy consumption to transmit task τ_{d_j,t_i} by the UAV u_k
$\mathcal{E}_{\tau_{d_j,t_i}}, E_{u_k}^{comp}$	The latency and energy consumption to process task τ_{d_j,t_i} by the UAV u_k
$C_{\tau_{d_j,t_i}}$	The finish time of the task τ_{d_j,t_i}
$\mathcal{F}_{u_k}, E_{u_k}^{move}$	The latency of the UAV u_k to fly a distance $l_{\alpha^{ang}}^{dis}$ and the energy consumption of this process
v_{u_k}	The velocity of UAV u_k
$P_{u_k}^{trans}, P_{u_k}^{comp}, P_{u_k}^{move}$	The transmission, computation and flying power of the UAV u_k

α^L and α^{NL} are the path loss exponents with respect to LoS and NLoS.

$$\zeta(\delta) = \begin{cases} \zeta^L(\delta) = A^L \delta^{-\alpha^L}, \, for\, LoS \\ \zeta^{NL}(\delta) = A^{NL} \delta^{-\alpha^{NL}}, \, for\, NLoS \end{cases} \tag{2}$$

We use $PLoS(\theta(u_k, d_j))$ to represent the LoS probability from a transmitter to a receiver, i.e., the u_k to d_j or d_j to u_k. This probability can be expressed as in (3), where λ and σ are coefficients determined by the specific environment, and θ is a function to describe the elevation angle between UAV u_k and MD d_j. The NLoS probability can be calculated by $PNLoS(\theta(u_k, d_j)) = 1 - PLoS(u_k, d_j)$.

$$PLoS(\theta(u_k, d_j)) = \frac{1}{1 + \sigma e^{-\lambda[\theta(l,k) - \sigma]}} \tag{3}$$

In this environment, the MD only communicates with at most one UAV to transmit its task to avoid repeated calculation. Some constraints have been defined as in (4) and (5).

$$\sum_{k=1}^{U} a_{d_j,u_k} \leqslant 1, \; \forall j \in D, k \in U \tag{4}$$

$$a_{d_j,u_k} \in \{0, 1\}, \;\; \forall j \in D, k \in U \tag{5}$$

Then the achievable upload rate of MD d_j to UAV u_k is shown in (6) [9], where ξ is the Signal-to-Noise Ratio(SNR), the difference between the receiving and sending process has been ignored. B is the bandwidth of the current channel.

$$r_{d_j,u_k}^{up} = B \log_2(1 + \frac{\xi p_{u_k}^{trans}}{(\delta_{d_j,u_k}^{dis})^2}) \tag{6}$$

The downlink rate between UAV u_k to MD d_j is also given in (7), where $h(u_k, d_j)$ is the power gain between the UAV u_k and MD d_j, N is the power spectral density.

$$r_{u_k,d_j}^{down} = B \log_2(1 + \frac{p_{u_k}^{trans} h(u_k, d_j)}{BN}) \tag{7}$$

Then we can calculate the throughput of the UAV u_k in a fixed length of time, as in (8), where the former term denotes an idea condition that all the tasks can be transmitted successfully and the UAV can receive tasks all the time. The latter term denotes the realistic condition, including discontinuous and uncompleted transmission during the fixed time.

$$\Omega^{u_k} = min \left\{ \sum_{i=0}^{T} \sum_{j=0}^{D} a_{d_j,u_k} \; \upsilon_{\tau_{d_j,t_i}} + \sum_{i=0}^{T} \sum_{j=0}^{D} a_{d_j,u_k} \gamma_{\tau_{d_j,t_i}}; \right. \\ \left. \sum_{i=1}^{T} \sum_{j=0}^{D} a_{d_j,u_k} r_{d_j,u_k}^{up} + \sum_{i=1}^{T} \sum_{j=0}^{D} a_{d_j,u_k} r_{u_k,d_j}^{down} \right\} \tag{8}$$

Based on the transmission rate, the transmission latency Γ_{u_k,d_j} between UAV u_k and MD d_j by (9).

$$\Gamma_{u_k,d_j} = \frac{\upsilon_{\tau_{d_j,t_i}}}{r_{d_j,u_k}^{up}} + \frac{\gamma_{\tau_{d_j,t_i}}}{r_{u_k,d_j}^{down}} \tag{9}$$

For simplicity, we set the size of $\gamma_{\tau_{d_j,t_i}}$ to be a proportional reduction of $\upsilon_{\tau_{d_j,t_i}}$.

3.3 Energy Consumption Model

The energy consumption model mainly includes three parts, transmission consumption, computation consumption, and movement consumption. The transmission energy consumption of UAV u_k and MD d_j can be denoted as in (10).

$$E_{u_k,\tau_{d_j,t_i}}^{trans} = \Gamma_{u_k.d_j} * P_{u_k}^{trans} \tag{10}$$

Besides, the computation latency and computation energy consumption of task τ_{d_j,t_i} are also considered, they can be denoted as in (11) and (12), where $P_{u_k}^{comp}$ denotes the computation power of UAV u_k.

$$\mathcal{E}_{\tau_{d_j,t_i}} = \frac{\chi_{u_k}^{max}}{\omega_{\tau_{d_j,t_i}}} \tag{11}$$

and

$$E_{u_k}^{comp} = \mathcal{E}_{\tau_{d_j,t_i}} * P_{u_k}^{comp} \tag{12}$$

We set the flying height of the UAV to under 400 feet, this obeys the rule of the Federal Aviation Administration of the US [26]. In the movement model, each UAV in this environment can choose a direction $\alpha^{ang} \in [0, 2\pi]$in a 2D plane and fly for a distance $l_{\alpha^{ang}}^{dis}$. Thus, the flying latency can be calculated by (13), where v_{u_k} denotes the velocity of UAV u_k. Then, the movement energy consumption of UAV u_k can be denoted as in (14).

$$\mathcal{F}_{u_k} = \frac{l_{\alpha^{ang}}^{dis}}{v_{u_k}} \tag{13}$$

$$E_{u_k}^{move} = \mathcal{F}_{u_k} * P_{u_k}^{move} \tag{14}$$

4 Problem Formulation

Based on the models mentioned above, our aim is to maximize throughput, as well as the number of completed tasks, and minimize the latency consumption of tasks in a fixed time $T$with the constraint of latency tolerance of tasks and the limited computation capacity of UAVs. We formulate this problem as a non-convex mixed integer programming problem which can be denoted as in (15).

$$(P): \quad max \left\{ \sum_{k=1}^{U} \Omega^{u_k}, \sum_{k=1}^{U} \sum_{i=1}^{T} (\hat{\mathcal{C}}_{t_i}^{in} + \hat{\mathcal{C}}_{t_i}^{de}), \sum_{k=1}^{U} \sum_{j=1}^{D} \frac{1}{\Gamma_{u_k,d_j}} \right\} \tag{15}$$

These three metrics are coupled with each other, and the throughput dominates the other two metrics. For example, suppose that the optimization objective is to maximize the number of completed tasks in a fixed time. In that case, throughput can be maximized with the increase of the completed number if the fairness of tasks with different volumes can be guaranteed. To simplify the problem 15, we transform it into a subproblem 16, which only needs to optimize the throughput in a fixed time while guaranteeing the fairness of different tasks. The problem $P1$ is shown as in (16).

$$(P1): \quad \underset{\hat{\mathcal{C}}_{t_i}^{in}, \hat{\mathcal{C}}_{t_i}^{de}, \frac{1}{\Gamma_{u_k,d_j}}}{max} \sum_{k=1}^{U} \sum_{j=1}^{D} \Omega^{u_k} \tag{16}$$

$$\text{s.t. } 0 \leq \varphi_{u_k,x,t_i}^{coor} \leq x_{max}, \forall k \in U, i \in T \tag{16a}$$

$$0 \leq \varphi_{u_k,y,t_i}^{coor} \leq y_{max}, \forall k \in U, i \in T \tag{16b}$$

$$\chi_{u_k,t_i}^{lo} \leq \chi_{u_k}^{max}, \forall i \in T \tag{16c}$$

$$\sum_{k=1}^{U} \chi_{u_k,t_i}^{lo} \leq \sum_{k=1}^{U} \chi_{u_k}^{max}, \forall i \in T, k \in U \tag{16d}$$

$$C_{\tau_{d_j,t_i}} \leq \sigma_{\tau_{d_j,t_i}}, \forall j \in U, i \in T \tag{16e}$$

$$\sum_{k=1}^{U} a_{d_j,u_k} \leqslant 1, \ \forall k \in U \tag{16f}$$

$$r_{d_j,u_k}^{up} \geq 0, \forall j \in D, k \in U \tag{16g}$$

$$r_{u_k,u_{k+1}}^{up} \geq 0, \forall k \in U \tag{16h}$$

In the optimization problem $P1$, Constraint (16a) and Constraint (16b) constrain the flight area of UAVs. Constraint (16c) and Constraint (16d) guarantee each UAV has a normal computation load, the high-loaded state may cause transmission failure, computation failure and even cooperation failure. Constraint (16e) is used to guarantee the task can be finished in time, where $C_{\tau_{d_j,t_i}}$ can be calculated by (17). Constraint (16f) constrains each MD only can communicate with one UAV. Constraint (16g) is used to help the UAV judge whether to communicate with one MD. Constraint (16h) guarantees the UAV can communicate with other UAVs, no matter whether it communicates directly or relay by the second UAV.

$$C_{\tau_{d_j,t_i}} = min \left\{ t_i + \Gamma_{u_k,d_j} + \mathcal{E}_{\tau_{d_j,t_i}}, t_i + \sigma_{\tau_{d_j,t_i}} \right\} \tag{17}$$

5 Proposed Solution

In this section, we give our solutions to the cooperation of UAVs and task scheduling problems, respectively. In the first problem, UAVs need to coordinate their formation according to the moving MDs which move with no regularity, and the observation of each UAV is limited, the cooperation metrics include transmission performance between UAVs and UAVs-to-MDs, etc. The second problem focuses mainly on optimizing the association between UAVs and MDs with latency and computation capacity constraints. To solve these two subproblems, we introduce a multi-agent reinforcement learning algorithm TF-PPO, which combines proximal policy gradient(PPO) [27] and task factorization network [28] with a deep neural network. After transmission from MDs to UAVs, UAVs need to schedule tasks to improve the completed ratio while balancing the computation load. This process influences the performance of UAVs and the QoE of MDs in the next. Then, we explain some specifics of these two algorithms.

5.1 The Cooperation Policy of UAVs

In this subsection, we first explain the components of the TF-PPO algorithm, the architecture is shown in Fig. 2. Then we introduce how to combine the task factorization network with the PPO algorithm in this environment.

5.1.1 The components of TF-PPO

To solve the multi-UAV cooperation problem under limited observation, we formulate it as a Partially Observable Markov Decision Process (POMDP) [10], which is defined as an eight-tuple $< S, A, T, O, R, \mathcal{Z}, \pi_\theta, \gamma^{dis} >$.

- *States*: $S \triangleq \{s_i\}$ is the state of UAVs which is shown in the lower left corner of Fig. 2, which includes the state of UAVs, and the number of MDs within the coverage of each UAV. We use $s_i = [s_{u_1,i}, ..., s_{u_k,i}, ..., s_{u_U,i}]$ to denote the states of UAVs at step i in the training process. For example, in the $i-th$ step, the state of UAV u_k can be denoted by $s_{u_k,i} =<< \varphi^{coor}_{u_k,x,t_i}, \varphi^{coor}_{u_k,y,t_i}, \varphi^{coor}_{u_k,z,t_i} >, a_{u_k,i}, r_{u_k,i}, s_{u_k,i+1} >$. After storing this state, the UAV can be transited to the next state $s_{u_k,i+1}$. This transition process can be used to measure whether the selected action is useful, then the action value $\widetilde{Q}_{u_k}$ of this action can be updated.
- *Action*: $A \triangleq \{a_i\}$ is the set of actions of all the UAVs. The matrix of all the joint-action of UAVs in the step i can be denoted as $a_i = [a_{u_1,i}, ..., a_{u_k,i} ..., a_{u_U,i}]$ where $a_{u_k,i} = [\alpha^{ang}, l^{dis}_{\alpha^{ang}}]$ includes the direction a^{dir} and angle a^{ang}. These two sub-actions determine the 3D coordinate of the UAV in the next step together. The change in altitude can help UAVs to avoid collisions. The UAV needs to select the direction and angle accurately to cover more MDs when the coordinates of MDs change. The core of this model is to get an optimal cooperation trajectory, where the trajectory is determined by the action selection in each state. The selection of action is related to the action value directly. If one action can achieve a higher reward during one step, the action value of this action will be updated to higher.
- *Transition Probability Function*: $T(S \times A \rightarrow S)$ is the probability of state $s' \in S$ after execute the joint action $[a_{u_1}, ..., a_{u_k}, ..., a_{u_U}]$ at the previous state $s \in S$.
- *Observation Probability Function*: O is the probability to observe $o \in O$ after executing a under the state s.
- *Partial Observation*: $\mathcal{Z}$ contains all the observations and $S \times A \times O \rightarrow \mathcal{Z}$ means the probability of getting the observation $z \in \mathcal{Z}$ according to the previous state s and action a.
- *Policy function*: π_θ is the policy function, which is a deep neural network with parameter $\theta_{u_k,a}$ to train the policy of selecting action a for the UAV u_k.
- *Discount Factor*: The notation $\gamma^{dis} \in [0, 1)$ is the discount factor, which is used to adjust the influence of the future reward to the calculation.
- *Reward*: $S \times A \rightarrow R$ denotes the immediate reward according to $a \in A$ to measure the selection of a. And the next state $s' \in S$ also influences the value of $r \in R$. To maximize the reward, i.e., to maximize the coverage and keep the cooperation of UAVs, the global reward of the state s_i has been defined in (18).

$$r = \sum_{k=1}^{U} \Omega^{u_k} \tag{18}$$

In this formula, Ω^{u_k} denotes the covered MDs by each UAV. Therefore, the total reward with discounted factor $\gamma^{dis} \in [0, 1]$ in the future can be shown as in (19).

$$\underset{\pi_\theta}{max} \mathbb{E}[\sum_{t=0}^{T-1} \sum_{\forall u_k \in \mathcal{U}} \gamma^{dis} r(s_i, a_i)] \tag{19}$$

$$\text{s.t. } s_i \in S,\ \pi_\theta(s_i) \in A \tag{19a}$$

$$\sum_{i=0}^{T} \sum_{k=1}^{U} u_k \delta^{is}_{u_k} = UT \tag{19b}$$

Constraint (19a) denotes the state s_i and the action a_i belong to S and A. Constraint (19b) means UAVs cannot lose contact with each other in every time slot t_i, $\delta^{is}_{u_K} = 1$ indicates that u_k can connect to either UAV, and $\delta^{is}_{u_K} = 0$ otherwise. If one UAV is isolated, its observation will be deduced, and the cooperation computation is unsustainable. To optimize this process, we use the task factorization network to train the global optimal action selection, and it will be introduced next.

5.1.2 Task Factorization Network in TF-PPO

Proximal Policy Optimization(PPO) reinforcement learning algorithm is developed from the actor-critic architecture [21] and the policy gradient technique [10]. The objective of this algorithm is to search for an optimal policy that can generate the optimal actions of the agents, which can be denoted as in (20), where ϵ is a clip fraction, and $A_i =$ is generalized advantage estimator(GAE) [27], which is used to optimize the advantage function. The $clip()$ function returns the upper and lower limits if the importance sampling [29] result is out of range. This equation directly limits the range of changes that the policy can make, then the stability during the training process has been improved.

$$J^{CLIP}(\pi_\theta) = \mathbb{E}[min(\frac{\pi_{\theta(a_{u_k,i}|s_i)}}{\pi_{\theta_{old}}(a_{u_k,i}|s_i)} A_i, clip(\frac{\pi_{\theta(a_{u_k,i}|s_i)}}{\pi_{\theta_{old}}(a_{u_k,i}|s_i)}, 1-\epsilon, 1+\epsilon) A_i] \tag{20}$$

The network architecture of PPO is developed from Actor-Critic (AC) architecture, it also has two network models to train, i.e., the actor network and the critic network. The actor network is used to select an appropriate action $a_{u_k,i}$ for UAV u_k in the $i-th$ step by policy gradient function, and the critic network is used to evaluate the result after executing action $a_{u_k,i}$ by value-based function. These two networks can act as an athlete and a judge to improve performance, respectively. The loss functions of these two networks in this paper are listed in (21) and (22).

$$L_a(\theta) = J^{CLIP}(\pi_\theta) \tag{21}$$

and

$$L_c(\phi) = E[(V(s_t) - (r_t + \gamma V(s_{t+1}))^2] \tag{22}$$

In MARL, each agent selects the optimal action by individual model according to the limited observation. As a result, the global optimal action set cannot be guaranteed. To compensate for this disadvantage, centralized training with decentralized execution(CTDE) has been proposed to expand the observation range of agents. However, the optimal joint action of all the agents still needs to be solved. In this background, the value decomposition network(VDN) [6] has been proposed to optimize the optimal joint-action selection process. VDN decomposes the actions of all the agents by assigning appropriate action value from the global value which is feedbacked from the joint-action execution. The idea of decomposition can be summarized as in (23), where $Q_{total}(a_i)$ is the sum of all the individual function $\widetilde{Q}(a_{u_k,i})$. Then a value decomposition neural network is trained to update the $Q_{total}(a_i)$ and $\widetilde{Q}(a_{u_k,i})$.

$$Q_{total}(a_i) = \sum_{k=1}^{U} \widetilde{Q}(a_{u_k,i}) \tag{23}$$

The loss function during the updating process can be shown as in (24), where the notation y_i and $Q^{pre}_{total}(a_{i-1})$ are calculated by $r_i + \gamma \underset{a}{argmax} Q^{pre}_{total}(a_{i-1})$ and $\sum_{k=1}^{U} Q^{pre}(a_{u_k,i-1})$, respectively. This decomposition network can optimize the Q function of each agent significantly. However, VDN cannot process complex tasks due to the accumulation, i.e., the sum of $\widetilde{Q}$ function cannot adapt to all the relationships between individual $\widetilde{Q}$ function and global Q_{total} function. Then the task factorization network has been proposed, which can cope with more complex relationships.

$$L(\theta) = \frac{1}{U}(y_i - Q^{pre}_{total}(a_i))^2 \tag{24}$$

The core of the task factorization network is to construct the relationship between the individual $\widetilde{Q}$ function and the global Q_{total} function. In this paper, in order to factorize Q_{total} to $\widetilde{Q}$, the individual-global-max(IGM) principle should be guaranteed. IGM is used to describe the equivalency of the individual optimality and the global optimality, which can be denoted as in (25).

$$\underset{a}{argmax} Q_{total}(a_i) = [\underset{a_{u_k}}{argmax} \widetilde{Q}(a_{u_k})]_{k=1}^{U} \tag{25}$$

It is the goal of the task factorization network, that can assign an appropriate reward value to each individual network from the global network. Then we need to find a set of individual $\widetilde{Q}$ to approximate the optimal Q_{total}, which can be shown as in (26), which means the sum of every optimal $\widetilde{Q}(a_{u_k})$ must higher than the sum of $\widetilde{Q}(a_{u_k})$ with other actions. If a set of $\widetilde{Q}$ functions satisfy this constraint, the IGM is also satisfied. To search for a $\widetilde{Q}$ set that meets this constraint, we combine the task factorization network with PPO as a multi-agent algorithm to estimate, the structure of this idea is shown in Fig. 2.

$$[\underset{a_{u_k}}{argmax} \widetilde{Q}(a_{u_k})]_{k=1}^{U} \geq [\widetilde{Q}(a_{u_k})]_{k=1}^{U} \tag{26}$$

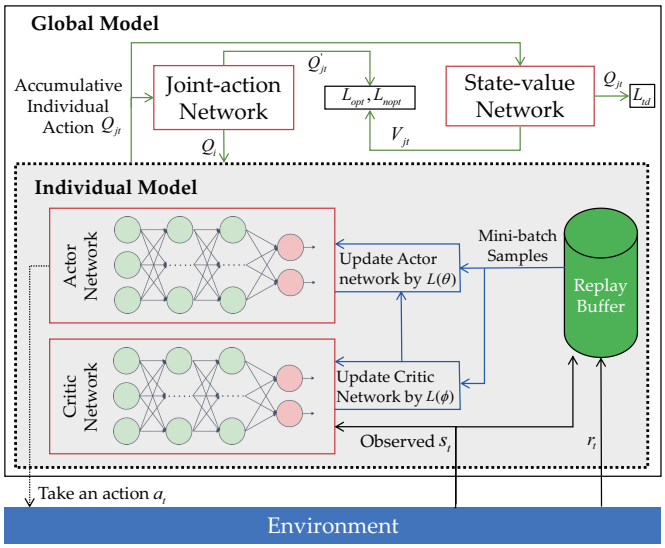


Fig. 2. The architecture of TF-PPO. The top part represents the global network to train the cooperation model of multiple UAVs, The bottom part represents the individual model training process of each UAV.

From Fig. 2, we can see the TF-PPO consists of two main networks, i.e., the individual network model in each UAV and a global network model for UAVs. The individual network model is mainly used to interact with the environment, monitor changes in the environment and choose appropriate action $a_{u_k,i}$ for UAV u_k in the $i-th$ step. This individual network model consists of two sub-network models, i.e., the actor network and the critic network. The actor-network is used to select action $a_{u_k,i}$ and the critic network evaluates the performance of action $a_{u_k,i}$ to improve the actor network. The global network includes two sub-network models, the first model is the joint action-value network model, which is used to approximate joint action-value Q_{total}, it receives the selected action by each UAV and outputs the $\widetilde{Q}$value of them. Another model is the state-value network model, which is used to compute a state-value $V(s)$to reduce the gap between $\widetilde{Q}$ and Q_{total}. Pseudocode is shown in Algorithm 1 and Algorithm 2.

5.2 The Association Policy Between UAVs and Tasks

In the dynamic environment, the generation of tasks is random and difficult to predict. Therefore, MOGS focuses on achieving a near-optimal association between UAVs and tasks under constraints such as delay and computing capacity in dynamic environments through a few iterations.

In order to solve this problem, we propose an improved many-to-one association algorithm based on Gale-Shapley [30], which is called Many-to-One Gale-Shapley(MOGS). This idea is inspired by [31] and the association process has been improved. The GS algorithm is also called the deferred-acceptance algorithm. It is usually used to solve the stable association problem. In an association problem, the two sides' agents to associate have an equal number and each agent has a rank list to select the preferred agent on the other side. During this process, if one agent has been selected by two agents on another side simultaneously,

it will select the preferred agent and release the original agent that has matched with it. The GS algorithm can realize a stable association with few iterations although it cannot guarantee the result is optimal. In this work, the MOGS inherits the advantages of the GS to realize a stable association and break its disadvantages, i.e., constrain the number of members on both sides. Next, the components and the association of MOGS are introduced.

Algorithm 1 Individual Model

Input: MD locations, UAV locations
Output: The cooperation model
Initialize actor network A_θ and critic C_ϕ network randomly with parameters θ and ϕ
Initialize old actor network $A_{\theta,old} \leftarrow \theta$ with parameter θ_{old}
Initialize buffer $\mathcal{D} \leftarrow \oslash$, mini-batch k_{mini}
Set the position of u_k randomly
for Episode in $1, 2, ..., N$ **do**
  $t \leftarrow 0$
  for t in $1, 2, ..., T$ **do**
    Observe current state s_t
    Select an action a_t by the old actor network $A_{\theta,old}$
    Obtain the next state s_{t+1}, reward r_t
    Collect previous trajectory $\mathcal{D} \leftarrow \mathcal{D} \cup \{s_t, a_t, r_t, s_{t+1}\}$
    Update the position of u_k
    if $|\mathcal{D}| = D$ **then**
      Sample mini-batch k_{mini} data from $\mathcal{D}$
      Send data to Global model and wait for the result
      Update $A_\theta \leftarrow \theta$ by Loss function
      Update $C_\phi \leftarrow \phi$ bu Loss function
      Update old actor $A_{\theta,old}$ by $\theta_{old} \leftarrow \theta$
      Clear the buffer $\mathcal{D} \leftarrow \oslash$
    end if
  end for
end for

Algorithm 2 Global Model

Input: Total observations from UAVs
Output: Factorized Q_{jt} to each UAV
Initialize replay memory D
Initialize $[Q_i]$, Q_{jt}, V_{jt} with random parameters θ
Initialize target parameters $\theta^- = \theta$
for episode = $1, ..., N$ **do**
  Collect initial state s^0 and observation $o^0 = [O(s^0, i)]_{i=1}^N$ from each agent i
  for $t = 1$ to T **do**
    With probability ϵ select a random action a_i^t
    Update $\theta^- = \theta$ with fixed period
  end for
end for

5.2.1 The components of MOGS algorithm

- *Active Party*: The active party in MOGS is the set of tasks $\mathcal{M}_{t_i}$ generated by MDs in t_i, every task has a rank list to sort the UAVs which can process it, and it requests to associate with the first UAV in its list.
- *Passive Party*: The passive party in MOGS is the set of UAVs $\mathcal{U}$, it only receives the tasks from MDs but does not select the tasks actively. Every UAV also has an equality to rank different types of tasks; this equality is the criterion to judge whether a task is suitable to process and will be introduced below.
- *Procedure*: The process of association mainly consists of three steps, and we use UAV u_k to denote the first UAV in the rank list of MD d_j here. First, in the t_i time slot, UAVs send the topology information and the state information to MDs in their coverage area. Then, after receiving the information from UAVs, every MD computes the rank list and sends its task to the UAV which is the first in the rank list. Finally, the UAV u_k receives the task τ_{d_j,t_i} generated by MD d_j due to the highest level in the rank list of MD d_j. The UAV u_i judges whether to process or relay τ_{d_j,t_i} to other UAVs according to the latency constraint of τ_{d_j,t_i} and the optimization formula which will be introduced below. To avoid some tasks being transmitted to another high-loaded UAV, we divide UAVs into two categories, i.e., $\mathcal{U}_{low}$ and $\mathcal{U}_{high}$. $\mathcal{U}_{high}$ includes high-loaded UAVs, another category $\mathcal{U}_{low}$ includes low-loaded UAVs. High-loaded UAVs in $\mathcal{U}_{high}$ transmit tasks in their task queue to low-loaded UAVs $\mathcal{U}_{low}$ under the latency constraint. This process can avoid task collision while guaranteeing the tasks transmitted can be computed.
- *Rule*: To solve the problem P while realizing the MOGS association under the constraints, some rules have been proposed to solve the subproblems during the association process. The first subproblem is to assign a suitable UAV to process the sudden tasks. The UAV can provide a higher transmission rate at closer distances, however, the UAV that is the closest to the task may not provide process capacity due to the overlong task queue and limited computation capacity. Thus, how to find a suitable UAV from the whole swarm within the constraint of tolerant latency of tasks under a dynamic environment needs to be solved. We propose an equation as a rule to help construct the association between UAVs and tasks, which can be formulated as in (27), where α^r, β^r are coefficients to trade the weight of uplink rate, downlink between UAV and task, γ^l is used to adjust the weight of the computation load of UAV. This equation mainly focuses on selecting the most suitable UAV for the task τ_{d_j,t_i}, the other UAVs will also be sorted based on this equation. However, its shortcomings are also obvious, the cooperation performance of UAVs may be degraded due to the overload of some UAVs. Then we consider proposing a rule to help UAVs get better cooperative task scheduling, it is also the second subproblem.

$$u^{ord} = \begin{cases} (max\left\{\alpha^r r^{up}_{d_j,u_k} + \beta^r r^{down}_{u_k,d_j} + \gamma^l \chi^{lo}_{u_k,t_i}\right\} \\ \forall j,k,i),\ 1 \\ other,\ 0 \end{cases} \tag{27}$$

The second subproblem is to coordinate the cooper-

ation between UAVs to avoid overloading a single UAV. The concurrency of tasks happens frequently in practice, and an overloaded UAV may cause optimization performance degradation while receiving overmuch tasks. To help the UAV u_k to judge whether a task τ_{d_j,t_i} can be processed by itself or transmit to UAV u_{k+1}, we design a rule as in (28).

$$u^{su} = \begin{cases} \Gamma_{u_{k+1},d_j} + \frac{\gamma_{\tau_{d_j,t_i}} + \chi^{lo}_{u_{k+1},t_i}}{\mathcal{E}_{u_{k+1}}} + \Delta^l \frac{\chi^{max}_{u_{k+1}}}{\chi^{lo}_{u_{k+1},t_i}} \\ \leq \sigma_{\tau_{d_j,t_i}} & ,1 \\ other & ,0 \end{cases} \tag{28}$$

From the equation (28), the UAV u_k can judge whether u_{k+1} is suitable for computing task τ_{d_j,t_i}. We additionally add a term $\Delta^l \frac{\chi^{max}_{u_{k+1}}}{\chi^{lo}_{u_{k+1},t_i}} \leq \sigma_{\tau_{d_j,t_i}}$ to denote the queue latency before computing τ_{d_j,t_i} if it is transmitted to u_{k+1}, where the $\Delta^l \in 0, 1$ can be set based on the environment, the more complex the environment, it closer to 1 if the environment is more complex.

Algorithm 3 MOGS Algorithm

1: **Input:** The set of tasks and UAVs
2: **Output:** The association result of tasks and UAVs
3: Initialization Data: The set of UAVs $\mathcal{U}$, the set of tasks$\mathcal{M}_{t_i}$
4: Compute the preference order $P^{UAV}_{\tau_{d_j,t_i}}$ of UAVs $\mathcal{U}$ by each task $\tau_{d_j,t_i}, j \in M$ according to $\alpha^r r^{up}_{d_j,u_k} + \beta^r r^{down}_{u_k,d_j} + \gamma^l \chi^{lo}_{u_k,t_i}$
5: **for** $u_k \in \mathcal{U}$ **do**
6: Compute the computation load of each UAV
7: **end for**
8: **for** steps=$[1, ..., N]$ **do**
9: **for** $u_k \in \mathcal{U}$ **do**
10: Divide $\mathcal{U}$ into $\mathcal{U}_{low}$ and $\mathcal{U}_{high}$ according to the computation load
11: **end for**
12: **for** $\tau_{d_j,t_i} \in u_k, u_k \in \mathcal{U}_{high}$ **do**
13: **if** Priority is satisfied **then**
14: **if** Latency is satisfied **then**
15: Transmit τ_{d_j,t_i} to u_k
16: **end if**
17: **end if**
18: **end for**
19: **end for**

Algorithm 3 describes the whole process of constructing association between UAVs and tasks, and how to schedule tasks between UAVs in detail. To construct an association relationship, tasks need to sort UAVs according to the transmission rate and the state of UAVs (Line 3). After sorting, some UAVs may be in the overload state and some tasks need to be scheduled (Line 5). To balance the computation load of UAVs, they are divided into $\mathcal{U}_{low}$ and $\mathcal{U}_{high}$ to transmit tasks (Lines 7-18). To select appropriate next UAV for one task τ_{d_j,t_i} in the queue of UAV u_k, the Equation (27) and Equation (28) are used to judge whether one UAV is suitable for τ_{d_j,t_i} (Line 11-15). More details on the stability analysis and the complexity analysis are provided in the Supplemental material.

6 Simulation Results And Discussion

In this section, we demonstrate the effectiveness of TF-PPO and MOGS through extensive simulations. These simulations are conducted on a DELL workstation with one RTX3090 graphic card and Intel(R) Xeon(R) Gold 6226R @2.90GHz, and the operation system is Win10 21H2. We set the size of the environment as a 2km $\times$ 2km, the MDs are distributed in this area and follow a PPP distribution. The number of UAVs ranges from 2 to 12 to demonstrate the performance of the TF-PPO. The coverage radius of UAV ranges from 30m to 130m, the bigger coverage radius means more UAVs can communicate with MDs directly without moving. The initial position of UAVs is not the same, and we guarantee each UAV can communicate with at least one UAV. The number of MDs is set as 800, dependent tasks and independent tasks are generated by other MDs and follow a normal distribution. Some more specific information on parameters are listed in Table 3.

TABLE 3
SIMULATION PARAMETERS

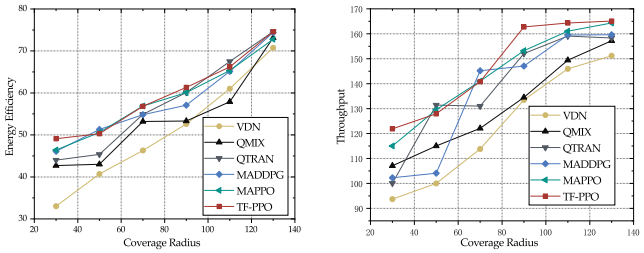
Parameters	Value
Computation capacity of UAV	$100MHz$
Transmission bandwidth	$50Mbps$
The volume of task	$0.1M \sim 1M$
The length of task set	1000
The number of UAVs	$2 \sim 12$
The propel power of UAV	$7W$
The coverage radius of UAV	$20m \sim 140m$
The flight speed of UAV	$25km/h$
The computation power of UAV	$5W$
The transmission power of UAV	$1W$
The value of Δ^l	0.8

6.1 Performance Verification and Discussion of TF-PPO

To verify the performance of TF-PPO, we use some metrics to measure, including the throughput and the energy consumption of UAVs, the computation latency of tasks and the number of completed tasks. Some reinforcement learning algorithms including VDN [6], QMIX [32], QTRAN [33], MADDPG [17] and MAPPO [18] are additionally selected to compare with TF-PPO. More details on these algorithms are provided in the Supplemental Material.

While training these algorithms, the maximum iteration steps are all set as 5000, and 100 steps in each episode. First, we compare the influence of energy efficiency and the throughput with different coverage radii of UAVs. The result is shown in Fig. 3.

In Fig. 3(a) and Fig. 3(b), The VDN algorithm has the worst performance. It cannot achieve a good performance due to its value decomposition function, which only relies on the accumulation from every individual Q value, it cannot reflect the complex environment accurately. The QMIX algorithm also has a disadvantage during the value decomposition process, it leverages the monotonicity between individual Q_i value and global Q_{total} value. They still cannot



(a) (b)

Fig. 3. The energy efficiency and throughput comparison with different coverage radii of UAVs.

approximate the complex environment. The performance of the QTRAN algorithm is close to the MADDPG but better than VDN and QMIX. It alternatively constructs a deep neural network for task factorizing, which is adapted to the TF-PPO. This network can train a model to study the Q_{total} value which is very close to the real Q_{total} value, this can make sure the global Q_{total} is right. The final two algorithms, i.e., the MADDPG and the MAPPO, their performance are very close to the TF-PPO. However, the TF-PPO still has an advantage while the coverage radius is 30m, which means the TF-PPO can cope with a more complex environment. With the help of the task factorization network, the TF-PPO can construct the right relationship between individual Q_i value and global Q_{total} value. The architecture of the TF-PPO is very similar to the MAPPO. They all adopt the centralized training and decentralized execution mode, and we especially optimize the global network by integrating the task factorization network. Thus, the TF-PPO can trade the decomposition between each individual action selection, and then the joint action selection process of all the UAVs can be optimized.

By observing Fig. 3, We can observe that the TF-PPO algorithm can achieve the upper bound efficiency while the radius is 30m, 90m and 130m, and the upper bound throughput while the radius is 30m, 90m, 110m and 130m in Fig. 3(a) and Fig. 3(b), respectively. We also discuss some disadvantages of the TF-PPO alternatively. For example, it cannot play to its strengths to achieve the upper bound energy efficiency when the coverage radius is 50m and 70m. We speculate the alternative task factorization network may influence the efficiency during the training process or the factorization of Q_{total} value still needs to be improved. We will continue executing more simulations to verify our idea.

We additionally verify the impact of the number of UAVs on the TF-PPO. The metrics are still energy efficiency and throughput. We additionally add a comparison respect, i.e., the convergence curve to observe the difference during the training process. The result can be summarized as follows.

Fig. 4(a) depicts the change of the energy efficiency under different numbers of UAVs, we can observe that the VDN algorithm has the worst performance, as well as the QMIX algorithm. The reason can be concluded that the value decomposition function cannot adapt to the complex environment. When the number of UAVs is increased, the state space and joint action space grow exponentially. Therefore, the training result cannot be improved quickly. Finally, their result is the worst under limited episodes. The efficiency of the MAPPO is higher than the MADDPG, the reason can be summarized as the optimization of the convergence process, especially the clip() function, which limits the update range to get a more stable result. We adapt this advantage and combine the task factorization network in the global network so that the performance of the TF-PPO can be better than the other algorithms.

Fig. 4(b) describes the throughput result achieved by these six algorithms. When the number of UAVs is 2, the throughput achieved by these algorithms is very similar. This result is because the movement of UAVs is not directly related to the throughput. After increasing the number of UAVs, more MDs can be covered, the performance of the cooperation policy decides the throughput significantly. From Fig. 4(b), we can find the throughput increases observably. However, the TF-PPO cannot achieve the best performance when the number of UAVs is 4, 6 and 8. We speculate that the MADDPG and the MAPPO still have some advantages when the environment is not very complex. When the number of UAVs increases to 10 and 12, the advantage of the task factorization network can be highlighted, i.e., the complex relationship between global Q_{total}value and individual $Q_i$can be constructed and the cooperation policy can be optimized. We can conclude that the TF-PPO is more suitable for the complex multi-agent environment.

Fig. 4(c) and Fig. 4(d) are the convergence curves of these six algorithms. Fig.4(c) describes the curve when the number of UAVs is 2. We can observe that the TF-PPO and the MAPPO can achieve the highest reward. However, the converge speed of the TF-PPO is slower than the MAPPO. We speculate the training of the task factorization network consumes some resources and the MAPPO is easier to train than the TF-PPO. From Fig. 4(d), we can observe the convergence of the TF-PPO is much quicker than other algorithms. The task factorization network makes the cooperative joint action selection improved, then the cooperative training is also improved.

In summary, we demonstrate the advantages of TF-PPO from five aspects, i.e., the energy efficiency and the throughput with different coverage radii, the energy efficiency and the throughput with different numbers of UAVs, the convergence curve with different numbers of UAVs. In these simulations, TF-PPO can construct the relationship between global Q_{total} and individual Q_i more accurately than the VDN, the QMIX and the QTRAN. And the TF-PPO can adapt to a more complex environment than the MADDPG and the MAPPO, especially in a multi-agent environment. Next, we will verify the performance of the MOGS by extensive simulations.

6.2 Performance Verification and Discussion of MOGS

In the second simulation, we verify the effectiveness of MOGS by comparing it with other algorithms, which include the Genetic algorithm, Ant Colony Optimization(ACO), Particle Swarm Optimization(PSO) and Greedy algorithm. We verify the performance of the MOGS compared with other algorithms from two aspects, the first metric is the completed ratio. It is calculated by dividing the number of tasks completed by the total number of tasks. The completed ratio reflects whether a task queue of one

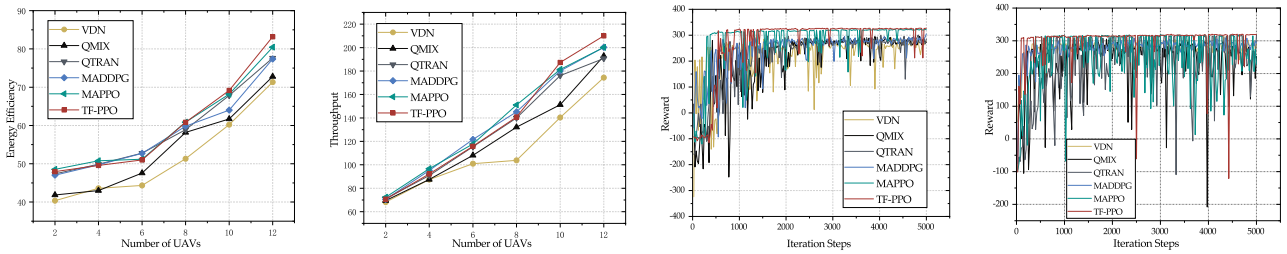


Fig. 4. Some comparisons with different numbers of UAVs.

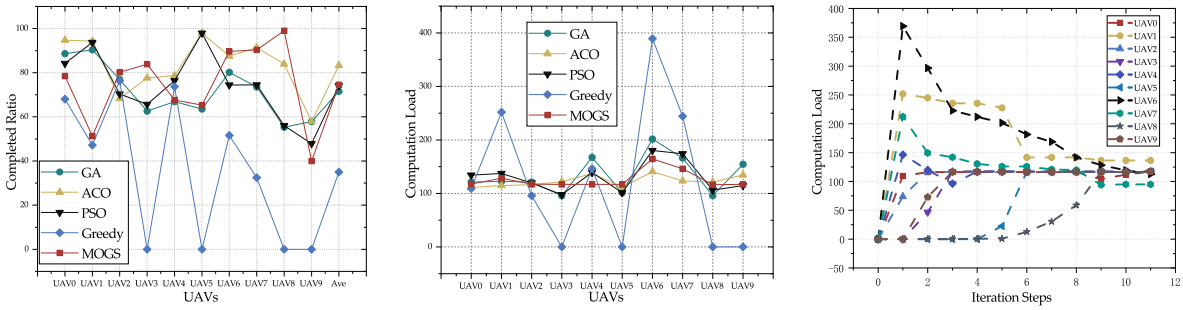
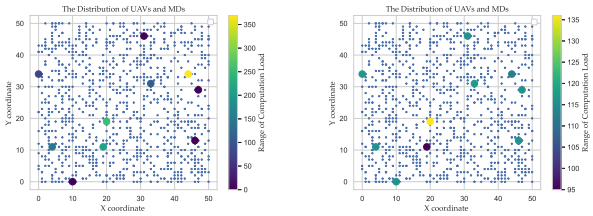


Fig. 5. The completed ratio and computation load comparison.

UAV is suitable for this UAV to compute. The second metric is the computation load of each UAV. The results of these two metrics are shown in Fig. 5.



(a) The initial state of the MOGS. (b) The final state of the MOGS.

Fig. 6. Visualizations of the MOGS.

Fig. 5(a) describes the completed ratio and the average completed ratio of each UAV in these algorithms. Where the 10 points on the left side denote the completed ratio of each UAV in these five algorithms, and the point on the far right denotes the average completed ratio of 10 UAVs in these 5 algorithms. We can observe that the Greedy algorithm has the worst performance. The performance of the PSO and the genetic algorithm are close to the MOGS. While the PSO is a bit higher than the genetic algorithm, we speculate the reason is that the PSO is better than the genetic algorithm in the global search respect. The genetic algorithm may fall into the local optimization although it has the mutation ability. The ACO has the best performance in the completed ratio, it can search for the optimal solution globally. However, we find it converges more slowly than others. Thus, the PSO may not be applied for practice although it has the best performance.

Fig. 5(b) describes the computation load of each UAV in these five algorithms. The smoother curve denotes the more balanced load of UAVs. The balanced status of UAVs can avoid the overloading of some UAVs to a certain extent. We can find the ACO is the most stable, and the MOGS has a relatively poor performance but is better than PSO and GA.

In this simulation, although the MOGS has not achieved the best performance in the above simulations, we still hold the point that the MOGS is the most suitable for the dynamic environment. Compared with the other four heuristic algorithms, the MOGS is extremely simple to achieve the result. The convergence process of the MOGS has been shown in Fig. 5(c). We can observe the MOGS only needs 12 iterations to be converged. The task set can be transmitted to UAVs and canceled all the time due to the insensitivity of the MOGS.

We also visualize the environment at the initial and final states in Fig. 6. In Fig. 6(a), we can observe that the computation load range of the UAV is 0 to 350, with different loads of the UAV with different colors. We can find that there is a high-load UAV, and three normal-load UAVs, the rest of the low-load. This situation reflects what happens if the task is only transmitted to the selected optimal UAV, that is, some UAVs are overloaded and some are under-loaded. The computation efficiency will be reduced in this situation. In Fig. 6(b), we can observe that the computation load of UAVs has been balanced. The computation load ranges from 95 to 135. There are merely two UAVs with high and low loads respectively. From this figure, it can be verified that MOGS is highly effective in balancing the loads of UAVs.

There are still some limitations of MOGS. It is impossible to transmit tasks multiple times due to the constraint of latency. The final matching result can not guarantee that all tasks can be completed, which is also related to the

latency constraint, but in the case of high concurrency, it is difficult to ensure that all tasks are completed due to the computation power limitation of the UAV. This is consistent with previous studies.

7 Conclusion

In this paper, we focus on optimizing multiple objectives within a dynamic and complex environment. The multiple objectives encompass the latency of the computation process of tasks, the energy consumption of UAVs, and the number of completed tasks. We convert these objectives into a single objective, i.e., maximizing throughput. This single-optimization problem is addressed through two processes: one is to optimize the cooperation among UAVs, and the other is to optimize the association process between tasks and UAVs. To optimize the cooperation of UAVs, we integrate the task factorization network with a deep reinforcement learning algorithm to train a multi-agent algorithm. To optimize the scheduling process of tasks, we propose an improved Gale-Shapley algorithm to enhance the performance of the association process. Finally, some simulations illustrate the performance of our solutions. In the future, we will continue to research a superior solution to optimize the multi-agent cooperation and task scheduling process.

Acknowledgment

This work is supported in part by the National Natural Science Foundation of China under Grant 62372310, and in part by the Liaoning Province Applied Basic Research Program under Grant 2023JH2/101300194, and in part by the LiaoNing Revitalization Talents Program under Grant XLYC2203151. Hawbani's work was supported in part by the Open Fund of Anhui Engineering Research Center for Intelligent Applications and Security of Industrial Internet, under Grant IASII24-04, and in part by Shenyang Aerospace University Talent Research Start-up Fund under Grant 502/120423005.

References

[1] Y. Zeng, R. Zhang, and T. J. Lim, "Wireless communications with unmanned aerial vehicles: Opportunities and challenges," *IEEE Communications magazine*, vol. 54, no. 5, pp. 36–42, 2016.

[2] Z. Sun, G. Sun, Y. Liu, J. Wang, and D. Cao, "Bargain-match: A game theoretical approach for resource allocation and task offloading in vehicular edge computing networks," *IEEE Transactions on Mobile Computing*, vol. 23, no. 2, pp. 1655–1673, 2024.

[3] H. Pan, Y. Liu, G. Sun, P. Wang, and C. Yuen, "Resource scheduling for uavs-aided d2d networks: A multi-objective optimization approach," *IEEE Transactions on Wireless Communications*, vol. 23, no. 5, pp. 4691–4708, 2024.

[4] J. Zhang, L. Zhou, Q. Tang, E. C.-H. Ngai, X. Hu, H. Zhao, and J. Wei, "Stochastic computation offloading and trajectory scheduling for uav-assisted mobile edge computing," *IEEE Internet of Things Journal*, vol. 6, pp. 3688–3699, Dec 2019.

[5] R. Liu, A. Liu, Z. Qu, and N. N. Xiong, "An uav-enabled intelligent connected transportation system with 6g communications for internet of vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, pp. 2045–2059, Oct 2023.

[6] Y. Hu, M. Chen, W. Saad, H. V. Poor, and S. Cui, "Distributed multi-agent meta learning for trajectory design in wireless drone networks," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 10, pp. 3177–3192, 2021.

[7] S. Mao, S. He, and J. Wu, "Joint uav position optimization and resource scheduling in space-air-ground integrated networks with mixed cloud-edge computing," *IEEE Systems Journal*, vol. 15, no. 3, pp. 3992–4002, 2021.

[8] Y. Shi, J. Wu, L. Liu, D. Lan, and A. Taherkordi, "Energy-aware relay optimization and power allocation in multiple unmanned aerial vehicles aided satellite-aerial-terrestrial networks," *IEEE Systems Journal*, vol. 16, no. 4, pp. 5293–5304, 2022.

[9] W. Zhou, L. Fan, F. Zhou, F. Li, X. Lei, W. Xu, and A. Nallanathan, "Priority-aware resource scheduling for uav-mounted mobile edge computing networks," *IEEE Transactions on Vehicular Technology*, vol. 72, pp. 9682–9687, Feb 2023.

[10] X. Wang, Z. Ning, S. Guo, M. Wen, L. Guo, and H. V. Poor, "Dynamic uav deployment for differentiated services: A multi-agent imitation learning based approach," *IEEE Transactions on Mobile Computing*, vol. 22, pp. 2131–2146, Sep 2023.

[11] Z. Ning, Y. Yang, X. Wang, L. Guo, X. Gao, S. Guo, and G. Wang, "Dynamic computation offloading and server deployment for uav-enabled multi-access edge computing," *IEEE Transactions on Mobile Computing*, vol. 22, pp. 2628–2644, Nov 2023.

[12] R. Zhou, X. Wu, H. Tan, and R. Zhang, "Two time-scale joint service caching and task offloading for uav-assisted mobile edge computing," in *IEEE INFOCOM 2022 - IEEE Conference on Computer Communications*, pp. 1189–1198, Jun 2022.

[13] C. Wang, D. Zhai, R. Zhang, H. Li, H. Cao, and A. Jindal, "Joint uavs position optimization and offloading decision for blockchain-enabled intelligent transportation," in *IEEE INFOCOM 2022 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pp. 1–6, Jun 2022.

[14] M. Hua, L. Yang, C. Pan, and A. Nallanathan, "Throughput maximization for full-duplex uav aided small cell wireless systems," *IEEE Wireless Communications Letters*, vol. 9, pp. 475–479, Dec 2020.

[15] B. Zhu, E. Bedeer, H. H. Nguyen, R. Barton, and Z. Gao, "Uav trajectory planning for aoi-minimal data collection in uav-aided iot networks by transformer," *IEEE Transactions on Wireless Communications*, vol. 22, no. 2, pp. 1343–1358, 2023.

[16] P. Qin, X. Wu, Z. Cai, X. Zhao, Y. Fu, M. Wang, and S. Geng, "Joint trajectory plan and resource allocation for uav-enabled c-noma in air-ground integrated 6g heterogeneous network," *IEEE Transactions on Network Science and Engineering*, vol. 10, no. 6, pp. 3421–3434, 2023.

[17] J. Wu, D. Li, Y. Yu, L. Gao, J. Wu, and G. Han, "An attention mechanism and adaptive accuracy triple-dependent maddpg formation control method for hybrid uavs," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–16, 2024.

[18] J. Kang, J. Chen, M. Xu, Z. Xiong, Y. Jiao, L. Han, D. Niyato, Y. Tong, and S. Xie, "Uav-assisted dynamic avatar task migration for vehicular metaverse services: A multi-agent deep reinforcement learning approach," *IEEE/CAA Journal of Automatica Sinica*, vol. 11, no. 2, pp. 430–445, 2024.

[19] X. Zhang, H. Zhao, J. Wei, C. Yan, J. Xiong, and X. Liu, "Cooperative trajectory design of multiple uav base stations with heterogeneous graph neural networks," *IEEE Transactions on Wireless Communications*, vol. 22, no. 3, pp. 1495–1509, 2023.

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

[21] J. Hu, H. Zhang, L. Song, R. Schober, and H. V. Poor, "Cooperative internet of uavs: Distributed trajectory design by multi-agent deep reinforcement learning," *IEEE Transactions on Communications*, vol. 68, no. 11, pp. 6807–6821, 2020.

[22] T. Ren, J. Niu, B. Dai, X. Liu, Z. Hu, M. Xu, and M. Guizani, "Enabling efficient scheduling in large-scale uav-assisted mobile-edge computing via hierarchical reinforcement learning," *IEEE Internet of Things Journal*, vol. 9, no. 10, pp. 7095–7109, 2022.

[23] Q. Tang, Z. Yu, C. Jin, J. Wang, Z. Liao, and Y. Luo, "Completed tasks number maximization in uav-assisted mobile relay communication system," *Computer Communications*, vol. 187, pp. 20–34, 2022.

[24] Z. Hu, F. Zeng, Z. Xiao, B. Fu, H. Jiang, H. Xiong, Y. Zhu, and M. Alazab, "Joint resources allocation and 3d trajectory optimization for uav-enabled space-air-ground integrated networks," *IEEE Transactions on Vehicular Technology*, vol. 72, no. 11, pp. 14214–14229, 2023.

[25] C. Liu, M. Ding, C. Ma, Q. Li, Z. Lin, and Y.-C. Liang, "Performance analysis for practical unmanned aerial vehicle networks

with los/nlos transmissions," in *2018 IEEE International Conference on Communications Workshops (ICC Workshops)*, pp. 1–6, Jul 2018.
[26] F. A. Administration, "PART 107—SMALL UNMANNED AIRCRAFT SYSTEMS." Website, June 2016. https://www.ecfr.gov/current/title-14/chapter-I/subchapter-F/part-107#107.41.
[27] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.
[28] K. Son, D. Kim, W. J. Kang, D. E. Hostallero, and Y. Yi, "QTRAN: Learning to factorize with transformation for cooperative multi-agent reinforcement learning," in *Proceedings of the 36th International Conference on Machine Learning* (K. Chaudhuri and R. Salakhutdinov, eds.), vol. 97 of *Proceedings of Machine Learning Research*, pp. 5887–5896, PMLR, 09–15 Jun 2019.
[29] A. R. Mahmood, H. P. Van Hasselt, and R. S. Sutton, "Weighted importance sampling for off-policy learning with linear function approximation," *Advances in neural information processing systems*, vol. 27, 2014.
[30] D. Gale and L. S. Shapley, "College admissions and the stability of marriage," *The American Mathematical Monthly*, vol. 69, no. 1, pp. 9–15, 1962.
[31] H. Hydher, D. N. K. Jayakody, K. T. Hemachandra, and T. Samarasinghe, "Uav deployment for data collection in energy constrained wsn system," in *IEEE INFOCOM 2022-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pp. 1–6, IEEE, Jun 2022.
[32] T. Rashid, M. Samvelyan, C. S. De Witt, G. Farquhar, J. Foerster, and S. Whiteson, "Monotonic value function factorisation for deep multi-agent reinforcement learning," *Journal of Machine Learning Research*, vol. 21, no. 178, pp. 1–51, 2020.
[33] K. Zhang, Z. Yang, and T. Başar, "Multi-agent reinforcement learning: A selective overview of theories and algorithms," *Handbook of reinforcement learning and control*, pp. 321–384, 2021.

Liang Zhao (Member, IEEE) is a Professor at Shenyang Aerospace University, China. He received his Ph.D. degree from the School of Computing at Edinburgh Napier University in 2011. He is also a JSPS Invitational Fellow (2023). He was listed as Top 2 % of scientists in the world by Standford University (2022 and 2023). He served as the Chair of several international conferences and workshops, including 2022 IEEE BigDataSE (Steering Co-Chair), 2021 IEEE TrustCom (Program Co-Chair), 2019 IEEE IUCC (Program Co-Chair), and 2018-2022 NGDN workshop (founder). He is Associate Editor of Frontiers in Communications and Networking and Journal of Circuits Systems and Computers. He is/has been a guest editor of IEEE Transactions on Network Science and Engineering, Springer Journal of Computing, etc.

Shuo Li is a student at Shenyang Aerospace University, Shenyang, China. He received the B.S. degree from Zaozhuang University, Zaozhuang, China in 2019. He is currently pursuing the M.S. in Shenyang Aerospace University. His research interests include Edge Computing, UAV trajectory planning and Digital Twin.

Zhiyuan Tan is an Associate Professor with the School of Computing, Engineering and the Built Environment, Edinburgh Napier University, UK. He received his Ph.D. degree from the University of Technology Sydney, Australia, in 2014, and was a Postdoctoral Researcher with the University of Twente, NL between 2014 and 2016. He is an Associate Editor of IEEE Transactions on Reliability, IEEE Open Journal of the Computer Society, Journal of Ambient Intelligence and Humanized Computing and the Journal of Ambient Intelligence and Humanized Computing, as well as an Academic Editor of Security and Communication Networks. He is a Senior Member of the IEEE and a Member of the ACM.

Ammar Hawbani is a Full Professor at the School of Computer Science at Shenyang Aerospace University. He earned his B.S. in Computer Software and Theory from the University of Science and Technology of China (USTC) in 2009. His academic journey continued with an M.S. in 2012 and a Ph.D. in 2016, all from USTC. Following his Ph.D. completion, he served as a Postdoctoral Researcher in the School of Computer Science and Technology at USTC from 2016 to 2019. Later, he worked as an Associate Researcher in the School of Computer Science and Technology at USTC from 2019 to 2023. Currently, he holds the position of Full Professor at the School of Computer Science in Shenyang Aerospace University. His research interests span IoT, WSNs, WBANs, WMNs, VANETs, and SDN.

Stelios Timotheou (Senior Member, IEEE) received the Dipl.-Ing. degree in electrical and computer engineering from the National Technical University of Athens and the M.Sc. degree in communications and signal processing and the Ph.D. degree in intelligent systems and networks from the Department of Electrical and Electronic Engineering, Imperial College London, in 2010. His research interests include monitoring, control, and optimization of critical infrastructure systems, with emphasis on intelligent transportation systems and communication systems. He is an Associate Editor of IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS and IEEE TRANSACTIONS ON INTELLIGENT VEHICLES.

Keping Yu (Senior Member, IEEE) received the M.E. and Ph.D. degrees from the Graduate School of Global Information and Telecommunication Studies, Waseda University, Tokyo, Japan, in 2012 and 2016, respectively. He was a Research Associate, a Junior Researcher, and a Researcher with the Global Information and Telecommunication Institute, Waseda University, from 2015 to 2019, from 2019 to 2020, and from 2020 to 2022, respectively. He is currently an Associate Professor, the Vice Director of Institute of Integrated Science and Technology, and the Director of the Network Intelligence and Security Laboratory (YU Lab), Hosei University, Japan.