

# Collaborative Route Planning of UAVs, Workers and Cars for Crowdsensing in Disaster Response

Lei Han, Chunyu Tu, Zhiwen Yu, Senior Member IEEE, Zhiyong Yu, Member IEEE, Weihua Shan, Liang Wang, and Bin Guo

**Abstract**—Efficiently obtaining the up-to-date information in the disaster-stricken area is the key to successful disaster response. Unmanned aerial vehicles (UAVs), workers and cars can collaborate to accomplish sensing tasks, such as life detection task in disaster-stricken areas. In this paper, we explicitly address the route planning for a group of agents, including UAVs, workers, and cars, with the goal of maximizing the sensing task completion rate. we propose a MARL-based heterogeneous multi-agent route planning algorithm called MANF-RL-RP. The algorithm has made targeted designs in terms of global-local dual information processing and model structure for heterogeneous multi-agent, making it effectively considers the collaboration among heterogeneous agents and the long-term impact of current decisions. Finally, we conducted detailed experiments based on the rich simulation data. In comparison to the baseline algorithms, namely Greedy-SC-RP and MANF-DNN-RP, MANF-RL-RP has exhibited a significant performance improvement. Compared to MANF-DNN-RP and Greedy-SC-RP, the task completion rate based on MANF-RL-RP increased by an average of 8.82% and 56.8%, respectively.

**Index Terms**—Mobile Crowdsensing, collaborative route planning, mult-agent reinforcement learning, disaster response.

## I. INTRODUCTION

Devastating disasters, as depicted in Figure 1 (e.g., earthquakes), can result in significant loss of life and widespread casualties within a short period of time. In particular, the chances of survival for individuals decrease significantly as the rescue time prolongs. For example, based on the common knowledge of earthquake relief [1], after an earthquake occurs, the survival probability of survivors is approximately 90% on the first day, but it decreases significantly to around 50%-60% on the second day. In such emergencies, rescuers require timely access to the latest information in the disaster-stricken area, as it serves as the foundation for subsequent effective rescue operations.

At present, mobile crowdsensing (MCS) [2] is an effective sensing paradigm, which has been widely used in environmental monitoring [3], public safety [4], intelligent transportation

Lei Han is with School of Computer Science and Technology, Xidian University, Xi'an, China & School of Computer Science, Northwestern Polytechnical University, Xi'an, China. E-mail: hanlei@xidian.edu.cn.

Zhiwen Yu (corresponding author) is with Harbin Engineering University, Harbin, China & Northwestern Polytechnical University, Xi'an China. E-mail: zhiwenyu@nwpu.edu.cn.

Liang Wang and Bin Guo are with School of Computer Science, Northwestern Polytechnical University, Xi'an 710072, China. E-mail: {liangwang, binguo}@nwpu.edu.cn.

Chunyu Tu and Zhiyong Yu are with College of Computer and Data Science, Fuzhou University, Fuzhou 350108, China. E-mail: {chunyutu, yuzhiyong}@fzu.edu.cn.

Weihua Shan is with Innovation Lab, Huawei Cloud Computing Technologies Co.,Ltd, Xi'an 710072, China. E-mail: ShanWeihua@huawei.com.

Manuscript received XX XX, XX; revised XX XX, XX.



Fig. 1. The Wenchuan earthquake, which resulted in a devastating toll: 67,183 deaths, 361,822 injuries, and 20,790 missing persons by 12:00 on May 27, 2008.

[5] and other fields. However, when a devastating disaster occurs, the environment in the disaster-stricken area becomes extremely complex and dangerous, which greatly limits the mobility of participants. Furthermore, since the traditional MCS relies on the participants and their mobile devices as the basic sensing unit, it is hard to work in the disaster-stricken area that require high sensing accuracy and specific sensing capabilities. With the popularization of unmanned aerial vehicles (UAVs) in recent years, UAVs play a crucial role in disaster response. UAVs, with their capabilities of rapid deployment, high mobility, and the ability to carry high-sensing sensors, can make up for the limitations of traditional MCS. Therefore, many researchers study how to apply UAVs to disaster response [6], [7], [8], [9], [10], [11], [12].

However, the existing researches have two unrealistic assumptions regarding UAVs, which hinder their practical application in disaster-stricken areas. (1) Existing researches assume that UAVs can perform sensing tasks (e.g., data collection) autonomously in the disaster-stricken area. However, the low-altitude environment of disaster-stricken areas poses numerous safety concerns, and many sensing tasks require precise maneuvering of UAVs in this challenging environment. During the execution of sensing tasks, UAVs not only have to navigate around obstacles effectively but also need to accurately detect crucial areas. Without the skilled intervention of professional personnel, it becomes extremely challenging for UAVs to autonomously carry out these sensing tasks. (2) Existing research assumes that UAVs have the capability to autonomously navigate to charging stations for recharging. However, the availability of charging stations specifically designed for UAVs is currently limited in urban areas. Moreover, after a devastating disaster, some charging stations may be damaged or rendered inoperable. Additionally, self-charging for UAVs in outdoor environments without human assistance is extremely challenging. Furthermore, the process of recharging

UAVs is time-consuming, which can significantly reduce their operational efficiency. To ensure uninterrupted performance of sensing tasks, a more efficient approach is for cars to directly replace UAV batteries instead of wasting time on UAV charging.

To address the aforementioned challenges, this paper focuses on investigating collaborative route planning for UAVs, workers, and cars to efficiently accomplish sensing tasks, as illustrated in Figure 2. The workers are responsible for the precise manipulation of UAVs at the sensing task locations, which is to overcome the limitations of UAVs in autonomous low-altitude maneuvering. Taking post-earthquake life detection as an example, UAVs can be equipped with life detection devices such as thermal imaging equipment, signal detection instruments, etc., and manipulated by skilled workers to search for survivors at the life detection task locations. Cars can swiftly replace the batteries of UAVs at designated endurance locations, which enables efficient replenishment of battery power for UAVs. In this particular application scenario, there are four questions that need to be illustrated. (1) Why do workers need the assistance of UAVs to perform the life detection tasks? Due to the complexity of the ground environment in disaster-stricken areas, the mobility of workers is severely constrained. In contrast, UAVs have significant advantages in maneuverability, allowing them to navigate hazards with flexibility and efficiently accomplish tasks at the life detection task locations. (2) Why doesn't the workers carry UAVs to perform the life detection tasks by themselves? The workers have limited mobility in disaster-stricken areas, and carrying UAVs would further restrict their mobility. Moreover, if the workers carry the UAVs, it would hinder the UAVs' ability to swiftly move between multiple life detection task locations, thus affecting their overall efficiency. (3) Why are UAVs able to autonomously fly between multiple life detection task locations? Unlike the complex low-altitude environment where life detection tasks are performed, UAVs can navigate between multiple task locations in the much simpler high-altitude environment. The high-altitude environment poses fewer challenges and obstacles for UAV flight. (4) Why can't workers themselves replace the batteries for the UAVs? Due to the limited mobility of workers in disaster-stricken areas, it would be impractical for them to carry spare batteries and hinder their mobility even further. As a result, workers are unable to fulfill the role of replacing UAV batteries.

In recent years, reinforcement learning (RL) has achieved outstanding performance in solving sequential decision-making problems [13], [14]. Therefore, this paper proposes a collaborative route planning approach for UAVs, workers, and cars in disaster response using multi-agent reinforcement learning (MARL). However, there are three challenges. (1) Traditional spatial crowdsourcing only involves two-dimensional matching of users and locations [15], [16]. 3D spatial crowdsourcing involves three-dimensional matching of users, workers and locations [17], [18]. Our problem involves four-dimensional matching of UAVs, workers, cars and location, which is more complex. (2) The attributes of UAVs, workers and cars vary greatly, such as mobility, endurance and function. In order to expedite the completion of



Fig. 2. Collaborative route planning of UAVs, workers and cars for crowdsensing.

sensing tasks, the route planning for UAVs, workers, and cars necessitates not only efficient spatio-temporal coordination but also appropriate functional alignment. (3) A large parameter scale is not conducive to model convergence. UAVs, workers, and cars exhibit heterogeneity, resulting in different state representations. Therefore, it is necessary to employ a shared neural network for all agents (i.e., UAVs, workers, and cars) to reduce the model's parameter scale. In summary, this work makes the following contributions:

(1) To the best of our knowledge, this work is the first research addressing the collaborative route planning of UAVs, workers, and cars in order to efficiently accomplish sensing tasks for crowdsensing in disaster response. In addition, we prove that the problem is NP-Hard.

(2) To tackle the aforementioned challenges, we propose a MARL-based heterogeneous multi-agent route planning algorithm called MANF-RL-RP. The algorithm has made targeted designs in terms of global-local dual information processing and model structure for heterogeneous multi-agent, making it effectively considers the collaboration among heterogeneous agents and the long-term impact of current decisions.

(3) We conducted detailed experiments based on the rich simulation data. In comparison to the baseline algorithms, namely Greedy-SC-RP and MANF-DNN-RP, MANF-RL-RP has exhibited a significant improvement in terms of task completion rate. The experimental code and data examples for this paper can be referenced from [19].

The contents of this paper are arranged as follows: Section II discusses some related works; Section III formulates the collaborative route planning of workers, cars and UAVs for crowdsensing in disaster response; Section IV models some concepts of the sequential decision-making process in this problem, and Section V implements the heterogeneous multi-agent route planning algorithms MANF-DNN-RP and MANF-RL-RP based on the concepts modeled in Section IV; then the experiments are conducted in Section VI; finally, conclusions and future work are summarized in Section VII.

## II. RELATED WORK

### A. Task allocation of traditional MCS

Task assignment of traditional Mobile Crowd Sensing (MCS) can be divided into two categories: single-task assignment and multi-tasks assignment. Single-task assignment focuses on the relationship between the spatial-temporal coverage of tasks and limited sensing resources, such as limited participants or sensing budget. For example, under the fixed sensing resources, maximize data quality or the overall utility of system [20], [21], [22], [23]. Alternatively, minimize sensing cost or the number of participants while ensuring data quality [24], [25], [26]. From single-task assignment to multi-tasks assignment, we need to take into account the following issues. From an optimization perspective, one must consider how to balance the quality of sensing data for multiple tasks while ensuring the quality of sensing data for each individual task [27], [28]. From a temporal perspective, the duration of different tasks may vary. It is necessary to consider corresponding task allocation strategies to address the varying time scales of different tasks [29]. From a spatial perspective, the spatial granularity of different tasks may also vary, and there may be inclusion relationships among them. It is necessary to address the spatial overlap between different tasks [30]. From the perspective of sensing content, different tasks may involve the same data. By assigning tasks based on data attributes, we can avoid data redundancy [31], [32]. In traditional MCS, participants and their mobile devices are considered as the fundamental sensing units. However, the mobility of participants and the sensing capabilities of mobile devices are limited.

### B. UAVs for MCS

In response to the limitations of traditional MCS, researchers have explored the integration of Unmanned Aerial Vehicles (UAVs) in MCS. UAVs possess exceptional maneuverability and can be equipped with capabilities of rapid deployment, high mobility, and the ability to carry high-sensing sensors. For example, UAVs can act as aerial base stations to assist in data transmission. Liu et al. considered to use a group of UAVs as aerial base stations to move around and collect data from multiple MCS users [10]. Liu et al. studied how to tackle the problem that a group of UAVs energy-efficiently and cooperatively collect data from low-level sensors, while charging the battery from multiple randomly deployed charging stations [11]. Liu et al. designed a fully-distributed control solution to navigate a group of UAVs, as the mobile base stations to fly around a target area, to provide long-term communication coverage for the ground mobile users [12]. In addition, UAVs can also be used to collect data. Zhou et al. considered the fixed-wing UAV-aided MCS system and investigate the corresponding joint route planning and task assignment problem from an energy efficiency perspective [6]. Liu et al. navigated a group of UAVs to move around a target area to maximize their total amount of collected data with the limited energy reserve, while geographical fairness among those point-of-interests should also be maximized [7]. Liu et al. explicitly considered to

navigate a group of UAVs in a 3-dimensional disaster work zone to maximize the amount of collected data, geographical fairness, energy efficiency, while minimizing data dropout due to limited transmission rate [8]. Liu et al. deployed UAVs in remote or hazardous areas to carry on long-term and hash tasks to achieve an optimal trade-off between maximizing the collected amount of data and coverage fairness, and minimizing the overall energy consumption of workers [9]. However, existing researches have made two unrealistic assumptions regarding UAVs, making it challenging for UAVs to be effectively utilized in disaster-stricken areas. Firstly, the assumption that UAVs can autonomously perform sensing tasks is not practical. Secondly, the assumption that UAVs can independently go to charging stations to recharge themselves is also unrealistic. In light of these challenges, we focus on studying collaborative route planning for UAVs, workers, and cars to address these issues. Workers are responsible for precise manipulation of UAVs at task locations, while cars play a crucial role in replacing UAV batteries to ensure sufficient battery life.

## III. PROBLEM DEFINITION

In this section, we begin by providing definitions for key concepts and subsequently formulate the collaborative route planning of UAVs, workers and cars for crowd-sensing in disaster response.

*Definition 1.* Discrete area set  $AREA = \{area_0, \dots, area_m, \dots\}$ .  $area_m = \langle obst_m, task_m \rangle$  represents the  $m$ -th area in  $AREA$ .  $obst_m$  is the obstacle identification of  $area_m$ . When there is an obstacle in  $area_m$ ,  $obst_m$  is marked as 1, otherwise 0.  $task_m$  is the sensing task identification of  $area_m$ . When there is a sensing task in  $area_m$ ,  $task_m$  is marked as 1, otherwise 0.

*Definition 2.* UAVs set  $UAV = \{uav_0, \dots, uav_i, \dots\}$ .  $uav_i = \langle uLoc_i^t, uRge_i^t, uPow_i^t, uCsp_i \rangle$  represents the  $i$ -th UAV in  $UAV$ .  $uLoc_i^t \in AREA$  represents the location of  $uav_i$  at moment  $t$ , ( $t \geq 0$ ).  $uRge_i^t \in [\emptyset, AREA]$  represents the range that  $uav_i$  can move within the time step  $[t, t+1]$ .  $uPow_i^t \in [0, 1]$  represents the remaining power of  $uav_i$  at the moment  $t$ .  $uCsp_i$  represents the power consumed by  $uav_i$  in a time step. We assume that  $uav_i$  consumes the same power at any time step.

*Definition 3.* Workers set  $Worker = \{wkr_0, \dots, wkr_j, \dots\}$ .  $wkr_j = \langle wLoc_j^t, wRge_j^t \rangle$  represents the  $j$ -th worker in  $Worker$ .  $wLoc_j^t \in AREA$  represents the location of  $wkr_j$  at moment  $t$ , ( $t \geq 0$ ).  $wRge_j^t \in [\emptyset, AREA]$  represents the range that  $wkr_j$  can move within the time step  $[t, t+1]$ .

*Definition 4.* Cars set  $Car = \{car_0, \dots, car_k, \dots\}$ .  $car_k = \langle cLoc_k^t, cRge_k^t \rangle$  represents the  $k$ -th car in  $Car$ .  $cLoc_k^t \in AREA$  represents the location of  $car_k$  at moment  $t$ , ( $t \geq 0$ ).  $cRge_k^t \in [\emptyset, AREA]$  represents the range that  $car_k$  can move within the time step  $[t, t+1]$ .

In real-world scenarios, UAVs, workers, and cars exhibit variations in mobility. To formulate this problem more clearly, we make the following assumption. If  $obst_m = 1$ , neither UAVs, workers nor cars can reach  $area_m$ , otherwise there is no restriction, refer to [7], [8], [33], [12]. In addition, workers

and UAVs can perform the sensing task when they meet at the sensing task location. The cars can replace the battery of UAVs when the UAVs meet the cars at designated endurance locations. In this paper, we assert that any unobstructed location can function as an endurance locations.

**Definition 5.** The routes set of UAVs,  $UTRA = \{uTra_0, \dots, uTra_i, \dots\}$ ,  $uTra_i = \{uLoc_i^0, \dots, uLoc_i^t\}$  represents the route of  $uav_i$ .

**Definition 6.** The routes set of Workers,  $WTRA = \{wTra_0, \dots, wTra_j, \dots\}$ ,  $wTra_j = \{wLoc_j^0, \dots, wLoc_j^t\}$  represents the route of  $wkr_j$ .

**Definition 7.** The routes set of Cars,  $CTRA = \{cTra_0, \dots, cTra_k, \dots\}$ ,  $cTra_k = \{cLoc_k^0, \dots, cLoc_k^t\}$  represents the route of  $car_k$ .

Before defining the problem of this paper, we need to be introduce the following constraints.

(1) When UAVs' battery is low, UAVs will stop moving, see Equation (1).

$$uRge_i^t = \emptyset, \text{ if } uPow_i^t < uCsp_i \quad (1)$$

(2) UAVs, workers, and cars cannot move to obstacles locations, see Equation (2).

$$uLoc_i^t.obst_m \neq 1 \& wLoc_j^t.obst_m \neq 1 \& cLoc_k^t.obst_m \neq 1 \quad (2)$$

(3) If UAVs meet the workers at a sensing task locations, the sensing task will be performed, see Equation (3).

$$uLoc_i^t.task_m = 0, \text{ if } uLoc_i^t = wLoc_j^t \& uLoc_i^t.task_m = 1 \quad (3)$$

(4) UAVs, workers and cars cannot move beyond the movable range within the time step  $[t, t + 1]$ , see Equation (4).

$$uLoc_i^{t+1} \in uRge_i^t \& wLoc_j^{t+1} \in wRge_j^t \& cLoc_k^{t+1} \in cRge_k^t \quad (4)$$

(5) If UAVs meet the cars, replace UAVs' battery. Otherwise, the power of UAVs will reduce or be unchanged, see Equation (5).

$$uPow_i^{t+1} = \begin{cases} 1, & \text{if } uLoc_i^{t+1} = cLoc_k^{t+1} \\ uPow_i^t - uCsp_i, & \text{if } uLoc_i^{t+1} \neq cLoc_k^{t+1} \& uPow_i^t \geq uCsp_i, \\ uPow_i^t & \text{if } uLoc_i^{t+1} \neq cLoc_k^{t+1} \& uPow_i^t < uCsp_i \end{cases} \quad (5)$$

**Problem 1** (collaborative route planning of UAVs, workers and cars for crowdsensing in disaster response): Given discrete area set  $AREA$ , UAVs set  $UAV$ , workers set  $Worker$ , cars set  $Car$ , and upper limit of sensing time  $TimeLimit$ . Determine the routes set of UAVs  $UTRA$ , the routes set of workers  $WTRA$  and the routes set of cars  $CTRA$  during  $[0, TimeLimit]$  to maximize the sensing tasks completion  $\sum_m task_m^0 - \sum_m task_m^{TimeLimit}$ . Formally,

$$\begin{aligned} & \text{confirm } UTRA, WTRA, CTRA \\ & \max \sum_m task_m^0 - \sum_m task_m^{TimeLimit} \\ & \text{s.t. constraints (1), (2), (3), (4), (5)} \end{aligned} \quad (6)$$

**Lemma 1. Problem 1** is NP-Hard.

*Proof:* Based on the problem definition, we can understand that the constraints of **Problem 1** require collaboration between workers and UAVs to complete sensing tasks, and UAVs need to encounter cars for battery replenishment. Therefore, **Problem 1** requires us to collaboratively plan the routes of workers, UAVs, and cars, with functional matching between each pair of them.

The constraints that need to be simultaneously satisfied for **Problem 1** are too complex. In order to prove that **Problem 1** is NP-Hard, we need to simplify the constraints of **Problem 1** and then prove that the simplified problem is NP-Hard. Therefore, we assume that UAVs have unlimited endurance and can autonomously perform sensing tasks. This means that we only need to plan the routes of UAVs without considering whether UAVs match workers or cars in terms of functionality. Based on this assumption, **Problem 1** can be expressed as **Problem 2**:

$$\begin{aligned} & \text{confirm } UTRA \\ & \max \sum_m task_m^0 - \sum_m task_m^{TimeLimit} \\ & \text{s.t. } uLoc_i^t.obst_m \neq 1 \\ & \quad uLoc_i^t.task_m = 0, \text{ if } uLoc_i^t.task_m = 1 \\ & \quad uLoc_i^{t+1} \in uRge_i^t \end{aligned} \quad (7)$$

**Problem 2** is a special case of **Problem 1** with simpler constraints. By proving that **Problem 2** is NP-Hard, we can then utilize problem reduction to conclude that **Problem 1** is also NP-Hard. However, it is worth noting that the assumption of simplifying **Problem 1** to **Problem 2** does not align with reality; it is merely an intermediate step used to prove the conclusion that **Problem 1** is NP-Hard.

In **Problem 2**, we only need to plan the routes of a group of UAVs without considering whether the UAVs are functionally matched with workers or cars. Then, we assume that we only need to plan the route of one UAV to perform the sensing task, indicating that there is only one UAV in the set of UAVs  $UAV$ . Based on this assumption, **Problem 2** can be expressed as **Problem 3**:

$$\begin{aligned} & \text{confirm } uTra_0 \\ & \max \sum_m task_m^0 - \sum_m task_m^{TimeLimit} \\ & \text{s.t. } uLoc_0^t.obst_m \neq 1 \\ & \quad uLoc_0^t.task_m = 0, \text{ if } uLoc_0^t.task_m = 1 \\ & \quad uLoc_0^{t+1} \in uRge_0^t \end{aligned} \quad (8)$$

**Problem 3** is a special case of **Problem 2** with simpler constraints. By proving that **Problem 3** is NP-Hard, we can then utilize problem reduction to conclude that **Problem 2** is also NP-Hard.

The objective of **Problem 3** is to find a route of UAV that maximizes the sensing tasks completion within the upper limit

of sensing time  $TimeLimit$ . In other words, **Problem 3** is a subset selection problem with time series, it is NP-Hard [34]. Therefore, **Problem 1** is NP-Hard.

#### IV. PROBLEM MODELING

We model **Problem 1** as a Markov decision process (MDP), defined as a tuple  $(\langle S, O \rangle, A, p, r, \gamma)$ .

##### A. State space

In this paper, we divide the information into two parts: global information and local information of agents (i.e., UAVs, workers and cars).

As shown in Figure 3, global information includes  $obstDist^t$ ,  $taskDist^t$ ,  $urgeDist^t$ ,  $workDist^t$  and  $carDist^t$ .  $obstDist^t$  represents the location distribution of obstacles at the moment  $t$ .  $taskDist^t$  represents the location distribution of sensing tasks at the moment  $t$ .  $urgeDist^t$  represents the urgency distribution to replace UAVs' battery at the moment  $t$ , which measures the cumulative urgency of replacing batteries for all UAVs in different locations.  $workDist^t$  represents the workers distribution at the moment  $t$ .  $carDist^t$  represents the cars distribution at the moment  $t$ .

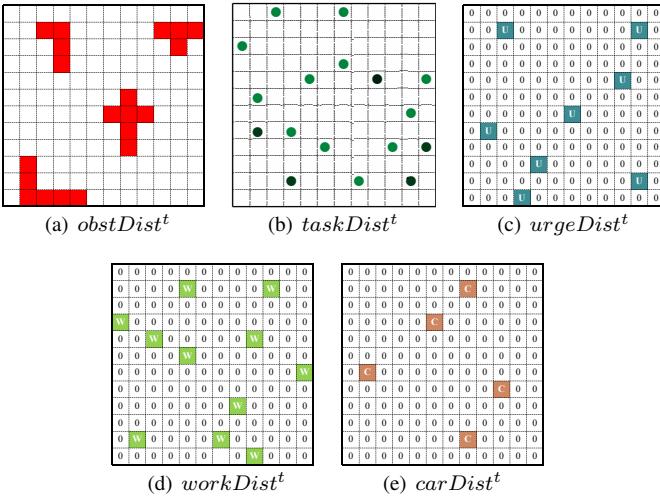


Fig. 3. Global information at the moment  $t$ .

As shown in Figure 4, local information of agents includes  $agentLoc_{ijk}^t$ ,  $agentArriv_{ijk}^t$ ,  $urge_{ijk}^t$  and  $agentID_{ijk}$ , note that  $(ijk = i/j/k)$ .  $agentLoc_{ijk}^t$  represents the location of the  $ijk$ -th agent at the moment  $t$ .  $agentArriv_{ijk}^t$  represents the areas that the  $ijk$ -th agent can reach within the time step  $[t, t + 1]$ . In real-world environments, the optional range  $agentArriv_{ijk}^t$  of all agents (UAVs, workers and cars) can be predefined based on the actual situation, and the optional range  $agentArriv_{ijk}^t$  has already eliminated unreachable locations.  $urge_{ijk}^t$  represents the urgency of the  $ijk$ -th agent to replace its battery at the moment  $t$ .  $agentID_{ijk}$  represents the ID numbers of the  $ijk$ -th agent.  $agentID_{ijk}$  is implemented based on one-hot encoding. For example, if the encoding at the  $ijk$ -th position of Fig.4 (d) is 1, it represents the  $ijk$ -th agent.

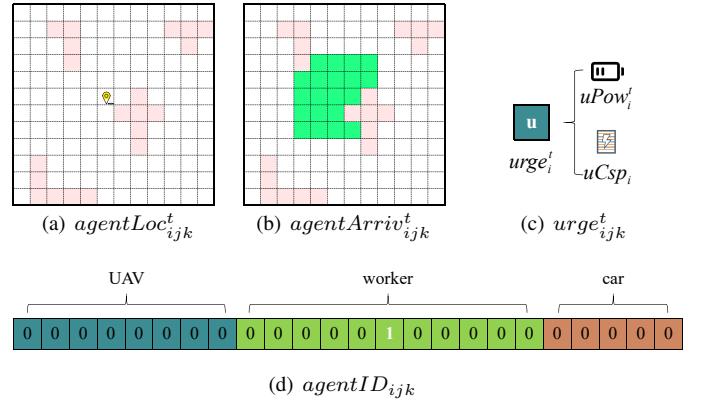


Fig. 4. Local information of the  $ijk$ -th agent at the moment  $t$ .

In this paper, the evaluation of  $urge_{ijk}^t$  needs to meet the following three conditions.

(1) The more the power of the  $ijk$ -th agent, the less its urgency. The urgency has a practical physical meaning. When  $uav_i$  has infinite power,  $uav_i$  would not need to replace its battery, and its urgency should be 0. Finally, we think that workers and cars have infinite power. Therefore, the urgency of workers and cars should be 0.

(2) The larger the power of the  $ijk$ -th agent, the less sensitive the urgency of  $uav_i$ . Therefore, as the power of  $uav_i$  increases, the decreasing speed of its urgency becomes small.

(3) The urgency of different agents can be added and the urgency of different areas can be compared. Therefore, there should be an upper limit with the urgency of  $uav_i$ .

Based on the above three conditions, we define the functional relationship among its current power  $uPow_i^t$ , power consumption  $uCsp_i$  in a time step and urgency  $urge_i^t$ , see Equation (9). "floor()" means round down function.

$$urge_i^t = \frac{1}{e^{\text{floor}(uPow_i^t/uCsp_i)}} \quad (9)$$

So, we can obtain,

$$\begin{aligned} \Delta urge_i^t &= \frac{\partial urge_i^t}{\partial \text{floor}(uPow_i^t/uCsp_i)} = \frac{\partial \frac{1}{e^{\text{floor}(uPow_i^t/uCsp_i)}}}{\partial \text{floor}(uPow_i^t/uCsp_i)} \\ &= -\left(\frac{1}{e^{\text{floor}(uPow_i^t/uCsp_i)}}\right) < 0 \end{aligned}$$

Besides,  $\text{floor}(uPow_i^t/uCsp_i) \propto uPow_i^t$

So,  $urge_i^t$  and  $uPow_i^t$  are inversely proportional. When  $uPow_i^t = +\infty$ ,  $urge_i^t$  is the smallest, which is 0. Condition (1) is satisfied.

$$\begin{aligned} \frac{\partial \Delta urge_i^t}{\partial \text{floor}(uPow_i^t/uCsp_i)} &= \frac{\partial \left(-\frac{1}{e^{\text{floor}(uPow_i^t/uCsp_i)}}\right)}{\partial \text{floor}(uPow_i^t/uCsp_i)} \\ \text{Besides, } &\frac{1}{e^{\text{floor}(uPow_i^t/uCsp_i)}} > 0 \end{aligned}$$

So,  $\Delta urge_i^t$  and  $uPow_i^t$  are proportional, and  $\Delta urge_i^t < 0$ . Condition (2) is satisfied.

When  $uPow_i^t = 0$ ,  $urge_i^t$  is the largest, which is 1. Condition (3) is satisfied.

Based on the global information and local information of agents, we can construct global state  $S = \{s^0, \dots, s^t, \dots\}$  and local state of agents  $O = \{\{o_0^0, \dots, o_{ijk}^0, \dots\}, \dots, \{o_0^t, \dots, o_{ijk}^t, \dots\}, \dots\}$ , refer to Equation (10) and Equation (11) for details.

$$s^t = \{obstDist^t, taskDist^t, urgeDist^t, workDist^t, carDist^t\} \quad (10)$$

$$o_{ijk}^t = \{s^t, agentLoc_{ijk}^t, agentID_{ijk}, urge_{ijk}^t\} \quad (11)$$

$agentArriv_{ijk}^t$  is used to filter the unreachable locations for the  $ijk$ -th agent at the moment  $t$ .

### B. Action space

Action set  $A = \{\{a_0^0, \dots, a_{ijk}^0, \dots\}, \dots, \{a_0^t, \dots, a_{ijk}^t, \dots\}, \dots\}$ .  $a_{ijk}^t = uLoc_i^t/wLoc_j^t/cLoc_k^t$  represents the location that the  $ijk$ -th agent will reach within the time step  $[t, t+1]$ . Based on the state space, we know that  $a_{ijk}^t$  is only affected by  $o_{ijk}^t$  and  $agentArriv_{ijk}^t$ . Therefore,  $\{a_0^0, \dots, a_{ijk}^t\}$  are independent with each other.

### C. State transition

$\langle S, O \rangle \times A \times \langle S, O \rangle \rightarrow p, (p \in [0, 1])$  represents the probability distribution of a state transition  $p(\{s^{t+1}, o_0^{t+1}, \dots, o_{ijk}^{t+1}\} | \{s^t, o_0^t, \dots, o_{ijk}^t, \dots\}, \{a_0^t, \dots, a_{ijk}^t, \dots\})$  in which the current state is  $\{s^t, o_0^t, \dots, o_{ijk}^t, \dots\}$ . When action  $\{a_0^t, \dots, a_{ijk}^t, \dots\}$  is chosen, the state is transitioned to a new state  $\{s^{t+1}, o_0^{t+1}, \dots, o_{ijk}^{t+1}, \dots\}$

**Lemma 2.**  $\{\{o_0^0, \dots, o_{ijk}^0, \dots\}, \dots, \{o_0^t, \dots, o_{ijk}^t, \dots\}, \dots\}$  satisfies the Markov property.

*Proof:* To prove Lemma 2., we need to prove Equation (12).

$$\begin{aligned} & \forall \{o_0^t, \dots, o_{ijk}^t, \dots\}, \\ & p\{\{o_0^t, \dots, o_{ijk}^t, \dots\} | \{o_0^{t-1}, \dots, o_{ijk}^{t-1}, \dots\}, \dots, \{o_0^0, \dots, o_{ijk}^0, \dots\}\} \\ & = p\{\{o_0^t, \dots, o_{ijk}^t, \dots\} | \{o_0^{t-1}, \dots, o_{ijk}^{t-1}, \dots\}\} \end{aligned} \quad (12)$$

According to state space and action space, we know  $\{o_0^t, \dots, o_{ijk}^t, \dots\} - \{o_0^{t-1}, \dots, o_{ijk}^{t-1}, \dots\} = \{a_0^{t-1}, \dots, a_{ijk}^{t-1}, \dots\}$ .

Besides,  $\{a_{ijk}^0, \dots, a_{ijk}^t\}$  are independent with each other. So, Equation (13) are independent with each other.

$$\begin{aligned} & \{\{o_0^1, \dots, o_{ijk}^1, \dots\} - \{o_0^0, \dots, o_{ijk}^0, \dots\}, \dots, \\ & \{o_0^t, \dots, o_{ijk}^t, \dots\} - \{o_0^{t-1}, \dots, o_{ijk}^{t-1}, \dots\}, \dots\} \end{aligned} \quad (13)$$

So,  $\{\{o_0^0, \dots, o_{ijk}^0, \dots\}, \dots, \{o_0^t, \dots, o_{ijk}^t, \dots\}, \dots\}$  has independent incrementality.

Combining with the definition of conditional probability, it can be seen as Equation (14).

Similarly, we can prove Equation (15).

Next, we can prove Equation (16).

Therefore,  $\{\{o_0^0, \dots, o_{ijk}^0, \dots\}, \dots, \{o_0^t, \dots, o_{ijk}^t, \dots\}, \dots\}$  satisfies the Markov property.

$$\begin{aligned} & p\{\{o_0^t, \dots, o_{ijk}^t, \dots\} | \{o_0^{t-1}, \dots, o_{ijk}^{t-1}, \dots\}, \dots, \{o_0^0, \dots, o_{ijk}^0, \dots\}\} \\ & = \frac{p\{\{\dots, o_{ijk}^t, \dots\}, \{\dots, o_{ijk}^{t-1}, \dots\}, \dots, \{\dots, o_{ijk}^0, \dots\}\}}{p\{\{\dots, o_{ijk}^{t-1}, \dots\}, \dots, \{\dots, o_{ijk}^0, \dots\}\}} \\ & = \frac{p\{\{\dots, o_{ijk}^t, \dots\} - \{\dots, o_{ijk}^{t-1}, \dots\}, \dots\}}{p\{\{\dots, o_{ijk}^{t-1}, \dots\} - \{\dots, o_{ijk}^{t-2}, \dots\}, \dots\}} \\ & = \frac{p\{\{\dots, o_{ijk}^t, \dots\} - \{\dots, o_{ijk}^{t-1}, \dots\}\}, \dots}{p\{\{\dots, o_{ijk}^{t-1}, \dots\} - \{\dots, o_{ijk}^{t-2}, \dots\}\}, \dots} \\ & = p\{\{o_0^t, \dots, o_{ijk}^t, \dots\} - \{o_0^{t-1}, \dots, o_{ijk}^{t-1}, \dots\}\} \end{aligned} \quad (14)$$

$$\begin{aligned} & p\{\{o_0^t, \dots, o_{ijk}^t, \dots\} | \{o_0^{t-1}, \dots, o_{ijk}^{t-1}, \dots\}\} \\ & = p\{\{o_0^t, \dots, o_{ijk}^t, \dots\} - \{o_0^{t-1}, \dots, o_{ijk}^{t-1}, \dots\}\} \end{aligned} \quad (15)$$

### D. Reward function

$\langle S, O \rangle \times A \rightarrow r$  represents the expected immediate reward received after the state is transitioned from  $\{s^{t-1}, o_0^{t-1}, \dots, o_{ijk}^{t-1}, \dots\}$  to  $\{s^t, o_0^t, \dots, o_{ijk}^t, \dots\}$ , due to taking the action  $\{a_0^{t-1}, \dots, a_{ijk}^{t-1}, \dots\}$ .

In **Problem 1**, the objective of workers and UAVs is to maximize the number of completed sensing tasks, while the objective of cars is to minimize the urgency of UAVs' power. Therefore, the expected immediate reward  $r^t$  should include two parts, see Equation (19). The first part is the sensing tasks completion  $taskCpt^t$  within the time step  $[t, t + 1]$ , see Equation (17). The second part is the reduced urgency  $\sum_i mtigU_i^t$ , which is due to the cars replace the batteries of UAVs at the moment  $t$ ,  $mtigU_i^t$  see Equation (18). "α" and "β" are used to measure the weights of two parts, where  $\alpha + \beta = 1$ . Equation (18) is used to calculate the reward for i-th UAV  $uav_i$  to alleviate urgency in three situations. (1) When  $uav_i$  does not meet any cars,  $uav_i$  obtains a reward of 0. (2) When  $uav_i$  meets the k-th car  $car_k$  and the remaining battery of  $uav_i$  is insufficient to support its flight,  $uav_i$  obtains a reward of  $\frac{1}{\text{floor}(\frac{1}{e} / uCsp_i)}$ . At this moment,  $uav_i$  no longer has flight capability and can only wait for the car, so its power will no longer continue to decrease, its urgency is the highest. (3) When  $uav_i$  meets  $car_k$  and the remaining battery of  $uav_i$  can still maintain its normal flight,  $uav_i$  obtains a reward of  $\frac{1}{\text{floor}((uPow_i^{t-1} - uCsp_i) / uCsp_i)} - \frac{1}{\text{floor}(\frac{1}{e} / uCsp_i)}$ . At this moment,  $uav_i$  still has flight capability and can go to a certain place to meet  $car_k$  to replace its battery, so its power will continue to decrease.

$$taskCpt^t = \sum_m task_m^{t-1} - \sum_m task_m^t \quad (17)$$

$$\begin{aligned} p\{\{o_0^t, \dots, o_{ijk}^t, \dots\} | \{o_0^{t-1}, \dots, o_{ijk}^{t-1}, \dots\}, \dots, \{o_0^0, \dots, o_{ijk}^0, \dots\}\} \\ = p\{\{o_0^t, \dots, o_{ijk}^t, \dots\} | \{o_0^{t-1}, \dots, o_{ijk}^{t-1}, \dots\}\} \end{aligned} \quad (16)$$

$$mtigU_i^t = \begin{cases} 0, & \text{if } uLoc_i^t \neq cLoc_k^t \\ 1 - \frac{1}{e^{\lfloor \frac{1}{uCsp_i} \rfloor}}, & \text{if } uLoc_i^t = cLoc_k^t \& uPow_i^{t-1} < uCsp_i \\ \frac{1}{e^{\lfloor \frac{(uPow_i^{t-1} - uCsp_i)/uCsp_i \rfloor}{uCsp_i} \rfloor}} - \frac{1}{e^{\lfloor \frac{1}{uCsp_i} \rfloor}}, & \text{if } uLoc_i^t = cLoc_k^t \& uPow_i^{t-1} \geq uCsp_i \end{cases}, \quad (18)$$

$$r^t = \alpha \times taskCpt^t + \beta \times \sum_i mtigU_i^t \quad (19)$$

The traditional expected immediate reward  $r^t$  see Equation (20). When any agent chooses an action outside its optional range  $agentArriv_{ijk}^t$ ,  $r^t$  is set to -10 (negative reward), which punishes current impossible action.

$$r^t = \begin{cases} \alpha \times taskCpt^t + \beta \times \sum_i mtigU_i^t, & \text{if } \forall a_{ijk}^t \in agentArriv_{ijk}^t \\ -10, & \text{else} \end{cases} \quad (20)$$

At the moment  $t$ , when an action is chosen randomly, the probability that we get a positive reward is  $pro^t = \prod_i \frac{uRge_i^t}{AREA} \times \prod_j \frac{wRge_j^t}{AREA} \times \prod_k \frac{cRge_k^t}{AREA}$ . In actual scenarios,  $pro^t$  will be extremely small, which leads to the sparse reward. Training models based on the sparse reward is difficult [35]. So, how do we filter out non-feasible actions to avoid negative rewards? During the training process, we use direct logical checks to ensure that each agent's actions are only generated within its optional range  $agentArriv_{ijk}^t$ . It directly avoids the selection of non-feasible actions, and effectively eliminates the occurrence of negative rewards. Therefore, this paper should use  $agentArriv_{ijk}^t$  to filter the non-optimal actions and calculate the expected immediate reward based on Equation (19). It's important to mention that directly adding  $taskCpt^t$  and  $\sum_i mtigU_i^t$  in numerical value is not explainable in terms of actual physical meaning. However, in order to estimate the cumulative reward for all types of agents within a single time step, this approach becomes necessary, as outlined in **Algorithm 1** for details. When we consider multiple time steps, it can be simplified into the Equation (17), as outlined in **Algorithm 2**.

**Lemma 3.** In **Problem 1**, UAVs, workers and cars are cooperative.

*Proof:* Workers need to manipulate UAVs to perform the sensing tasks. For the sensing tasks completion  $taskCpt^t$ , UAVs and workers are cooperative.

So,  $taskCpt^t \propto \{uLoc_0^t, \dots, uLoc_i^t, \dots\}$  and  $taskCpt^t \propto \{wLoc_0^t, \dots, wLoc_j^t, \dots\}$ .

The purpose of cars is to relieve the urgency of the UAVs' power as much as possible. For the reduced urgency  $\sum_i mtigU_i^t$ ,  $\sum_i mtigU_i^t \propto \{cLoc_0^t, \dots, cLoc_k^t, \dots\}$

Besides,  $\{uPow_0^t, \dots, uPow_i^t, \dots\}$  is proportional to the number of working UAVs, e.g.,  $\{uPow_0^t, \dots, uPow_i^t, \dots\} \propto \sum_i mtigU_i^t$ .

And,  $\{uLoc_0^t, \dots, uLoc_i^t, \dots\} \propto \{uPow_0^t, \dots, uPow_i^t, \dots\}$ .  
So,  $taskCpt^t \propto \{cLoc_0^t, \dots, cLoc_k^t, \dots\}$ .

Therefore, in **Problem 1**, UAVs, workers and cars are cooperative.

## V. METHODOLOGY

In this section, we will introduce the method for collaborative route planning of workers, cars and UAVs for crowdsensing in disaster response. Addressing our problem is confronted with at least three challenges. (1) Our problem introduces the challenge of four-dimensional matching, encompassing UAVs, workers, cars, and locations, thereby increasing complexity. (2) The diverse attributes of UAVs, workers, and cars, including mobility, endurance, and function, pose a challenge. Achieving efficient spatio-temporal coordination and ensuring functional alignment are crucial for accelerating the completion of sensing tasks. (3) Dealing with a large parameter scale can impede model convergence. The heterogeneity among UAVs, workers, and cars results in distinct state representations, necessitating the utilization of a shared neural network for all agents to reduce the overall model parameter scale.

To effectively address the aforementioned three major challenges, we have undertaken the following efforts. First, we design a heterogeneous multi-agent network framework (MANF) to address the **Problem 1**. It is worth highlighting that in MANF, each agent is responsible for controlling either a single UAV, a worker, or a car. Then, we proceed to implement heterogeneous multi-agent route planning algorithms, namely MANF-DNN-RP and MANF-RL-RP, using the MANF framework. MANF-DNN-RP leverages deep learning techniques and incorporates the latest research on UAVs' route planning [7], [11]. On the other hand, MANF-RL-RP is based on MARL and draws inspiration from the QMIX algorithm [36]. The MANF-DNN-RP algorithm only focuses on how to obtain the optimal action under the current state, while ignores the long-term impact on the subsequent state transitions and reward. However, the MANF-RL-RP algorithm can take into account the long-term impact on subsequent state transitions and reward [37], [38].

### A. MANF

Referring to the QMIX algorithm, the MANF consists of two main parts. One is the agent network, which outputs the value  $Q_{ijk}^t(a_{ijk}^t)$  for a single agent, while the mixing network takes  $Q_{ijk}^t(a_{ijk}^t)$  as input and outputs a joint value  $Q_{tot}^t(a^t)$ . To maintain consistency between the centralized policy and the decentralized agent policies (e.g., monotonicity), the network parameter weight and offset of the mixing network are calculated through the hypernetworks network [39]. The mixing network weight must be greater

than 0, and there is no requirement for mixing network offset. Based on the above description and combined with the research content of this paper, MANF is shown in Figure 5, and there are two points worth noting. (1) We divide the information into two parts, global information and local information of agents. Global information  $\{obstDist^t, taskDist^t, urgeDist^t, workDist^t, carDist^t\}$  needs to extract spatial features based on convolutional neural networks and share them with all agents.  $\{agentLoc_{ijk}^t, agentID_{ijk}, urge_{ijk}^t\}$  in the local information is used to construct the input of the agent network combining with the global state  $s^t$ .  $agentArriv_{ijk}^t$  in the local information is used to filter the non-optimal actions to avoid negative reward. (2) Due to the decision-making process  $\{\{o_0^0, \dots, o_{ijk}^0, \dots\}, \dots, \{o_0^t, \dots, o_{ijk}^t, \dots\}, \dots\}$  of a single agent satisfies the Markov property, referring to **Lemma 2**, we do not need to extract the time series features of agents in the agent network. In addition, since UAVs, workers, and cars are cooperative in **Problem 1**, referring to **Lemma 3**, the relationship between the joint actions value  $Q_{tot}^t(a^t)$  and the agents' action value  $\{Q_0^t(a_0^t), \dots, Q_{ijk}^t(a_{ijk}^t), \dots\}$  satisfies monotonicity. Therefore, in order to ensure the consistency of the joint strategy and the decentralized strategy, the weight of the Mixing Network needs to be non-negative [36].

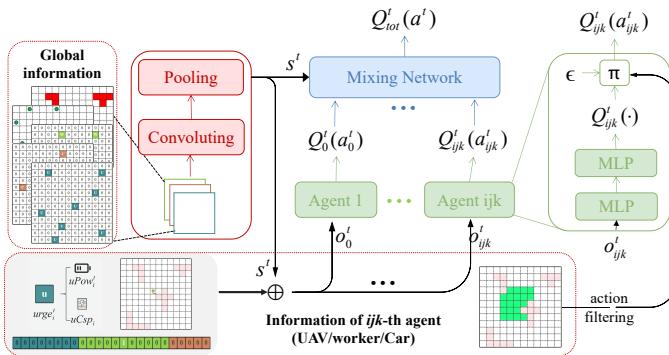


Fig. 5. Heterogeneous multi-agent network framework (MANF).

### B. MANF-DNN-RP

Combined with MANF, we implement the heterogeneous multi-agent route planning algorithm MANF-DNN-RP based on deep learning, as shown in **Algorithm 1**. The core idea of the MANF-DNN-RP algorithm is as follows. First, we calculate the expected immediate reward  $r^t$  for choosing the actions  $\{a_0^t, \dots, a_{ijk}^t, \dots\}$  under the current state  $\{o_0^t, \dots, o_{ijk}^t, \dots\}$  within the time step  $[t, t + 1]$ . The expected immediate reward  $r^t$  consists of two parts, the sensing tasks completion  $taskCpt^t$  and the reduced urgency  $\sum_i mtigU_i^t$  within the time step  $[t, t + 1]$ , refer to Equation (19). The whole process is shown in **Algorithm 1**, lines 5-16. Then, we can accurately represent the ternary mapping relationship  $\langle \{o_0^t, \dots, o_{ijk}^t, \dots\}, \{a_0^t, \dots, a_{ijk}^t, \dots\}, r^t \rangle$ . MANF-DNN-RP can be trained based on the ternary mapping relationship to accurately output the expected immediate reward  $r^t$  after choosing the actions  $\{a_0^t, \dots, a_{ijk}^t, \dots\}$  for the current state  $\{o_0^t, \dots, o_{ijk}^t, \dots\}$ , as shown in **Algorithm 1**, lines 19-20.

Finally, we can compare the expected immediate reward for different actions under the current state  $\{o_0^t, \dots, o_{ijk}^t, \dots\}$  based on MANF-DNN-RP, and choose an actions with the largest immediate reward. Repeat the above action selection process until reaching the target moment.

---

### Algorithm 1 : MANF-DNN-RP

---

**Input:** AREA, UAV, Worker, Car, TimeLimit  
**Output:** Spatial Convolutional Network cnnSpace, Agent Evaluation Network evalAgent, Mixing Network evalMixing

- 1: Initialize cnnSpace, evalAgent, evalMixing and experience pool  $D$  with size  $M$ ;
- 2: **while** cnnSpace, evalAgent and evalMixing do not converge **do**
- 3:     Index UAVs, workers, and cars with  $\{agentID_{ijk}\}$ ;
- 4:     **for**  $t = 0 \rightarrow TimeLimit$  **do**
- 5:         Get  $obstDist^t$  and  $taskDist^t$  based on AREA;
- 6:         Get  $\{urge_{ijk}^t\}$  and  $urgeDist^t$  based on UAV;  
 $//urge_{ijk}^t$  of workers and cars are set to 0.
- 7:         Get  $workDist^t$  based on Worker;
- 8:         Get  $carDist^t$  based on Car;
- 9:         Get  $\{agentLoc_{ijk}^t\}$  and  $\{agentArriv_{ijk}^t\}$  based on UAV, Worker and Car;
- 10:          $s^t = \text{cnnSpace}(obstDist^t, taskDist^t, \bullet, \bullet, \bullet);$   
 $// \bullet = urgeDist^t / workDist^t / carDist^t.$
- 11:          $\{o_{ijk}^t\} = \{\{s^t, agentLoc_{ijk}^t, agentID_{ijk}, urge_{ijk}^t\}\};$
- 12:          $\{Q_{ijk}^t(\cdot)\} = \{\text{evalAgent}(o_{ijk}^t)\};$
- 13:         Get  $\{a_{ijk}^t\}$  combining with  $\{Q_{ijk}^t(\cdot)\}$  and  $\{agentArriv_{ijk}^t\}$  based on  $\varepsilon - \text{greedy}$ ;
- 14:         Update AREA, UAV, Worker and Car based on  $\{a_{ijk}^t\}$ ;
- 15:         Get  $r^t$  based on Equation (19);
- 16:          $\{s^t, \{o_{ijk}^t\}, \{a_{ijk}^t\}, r^t\}$  to  $D$ ;
- 17:     **end for**
- 18:     **if**  $\text{length}(D) \geq M$  **then**
- 19:         Randomly sample training data  $trainData$  in  $D$ ;
- 20:          $L(\theta) = r^t - \text{evalMixing}(s^t, \{Q_{ijk}^t(o_{ijk}^t, a_{ijk}^t)\});$
- 21:         Update cnnSpace, evalAgent and evalMixing based on  $L(\theta)$ ;
- 22:     **end if**
- 23: **end while**

---

### C. MANF-RL-RP

The MANF-DNN-RP algorithm only focuses on how to obtain the optimal action  $\{a_0^t, \dots, a_{ijk}^t, \dots\}$  under the current state  $\{o_0^t, \dots, o_{ijk}^t, \dots\}$  within the time step  $[t, t + 1]$ , while ignores the long-term impact on the subsequent state transitions and reward. Therefore, we implement a heterogeneous multi-agent route planning algorithm MANF-RL-RP based on MARL, which can take into account the long-term impact of  $\{o_0^t, \dots, o_{ijk}^t, \dots\}$ . Based on Equation (19), we can estimate the total expected immediate reward  $G^t$  within the time step  $[t, TimeLimit]$ .  $G^t$  represents the cumulative sum of immediate reward during the time step  $[t, TimeLimit]$ , referring to Equation (21).  $\gamma$  is the discount factor, where a higher value indicates a greater emphasis on future immediate reward,

**Algorithm 2 : MANF-RL-RP**


---

**Input:** AREA, UAV, Worker, Car, TimeLimit,  $\gamma$   
**Output:** Spatial Convolutional Network cnnSpace, Agent Evaluation Network evalAgent, Mixing Network evalMixing

- 1: Initialize cnnSpace, evalAgent, evalMixing, experience pool  $D$  with size  $M$ ;
- 2: **while** cnnSpace, evalAgent and evalMixing do not converge **do**
- 3:     Index UAVs, workers, and cars with  $\{agentID_{ijk}\}$ ;
- 4:     **for**  $t = 0 \rightarrow TimeLimit$  **do**
- 5:         Get  $s^t$ ,  $\{o_{ijk}^t\}$  and  $\{agentArriv_{ijk}^t\}$ , referring to lines 5-13 in **Algorithm 1**;
- 6:          $\{Q_{ijk}^t(\cdot)\} = \{\text{evalAgent}(o_{ijk}^t)\}$ ;
- 7:         Get  $\{a_{ijk}^t\}$  combining with  $\{Q_{ijk}^t(\cdot)\}$  and  $\{agentArriv_{ijk}^t\}$  based on  $\varepsilon$ -greedy;
- 8:         Update AREA, UAV, Worker and Car based on  $\{a_{ijk}^t\}$ ;
- 9:         Get  $r^t = taskCpt^t$  based on Equation (17);
- 10:         Get  $s^{t+1}$ ,  $\{o_{ijk}^{t+1}\}$  and  $\{agentArriv_{ijk}^{t+1}\}$ ;
- 11:         **if**  $t == TimeLimit$  **then**
- 12:              $te^t = 0$ ;
- 13:         **else**
- 14:              $te^t = 1$ ;
- 15:         **end if**
- 16:         Add  $s^t$ ,  $s^{t+1}$ ,  $\{o_{ijk}^t\}$ ,  $\{o_{ijk}^{t+1}\}$ ,  $\{agentArriv_{ijk}^{t+1}\}$ ,  $\{a_{ijk}^t\}$ ,  $r^t$ ,  $te^t$  to  $D$ ;
- 17:     **end for**
- 18:     **if**  $\text{length}(D) \geq M$  **then**
- 19:         Randomly sample training data  $trainData$  in  $D$ ;
- 20:          $\{mQ_{ijk}^{t+1}\} = \{\max(\text{tgtAgent}(o_{ijk}^{t+1}, \bullet))\}$ ;  
//  $\bullet = agentArriv_{ijk}^{t+1}$ .
- 21:          $L(\theta) = r^t + \gamma \cdot te^t \cdot \text{evalMixing}(s^{t+1}, \{mQ_{ijk}^{t+1}\}) - \text{evalMixing}(s^t, \{Q_{ijk}^t(o_{ijk}^t, a_{ijk}^t)\})$ ;
- 22:         Update cnnSpace, evalAgent and evalMixing based on  $L(\theta)$ ;
- 23:     **end if**
- 24: **end while**

---

while a lower value indicates a greater emphasis on immediate reward in the near term.

$$\begin{aligned} G^t &= r^t + \gamma r^{t+1} + \dots + \gamma^{TimeLimit-t} r^{TimeLimit} \\ &= (taskCpt^t + \sum_i mtigU_i^t) + \dots + \\ &\quad \gamma^{TimeLimit-t} (taskCpt^{TimeLimit} + \sum_i mtigU_i^{TimeLimit}) \end{aligned} \quad (21)$$

However, the impact of the cars on the task completion rate is delayed, referring to **Lemma 3**. Replacing the UAVs' batteries with the cars at the current moment can reduce the occurrence of the UAVs halting operations in future moments due to insufficient power. In other words,  $\sum_i mtigU_i^t$  will be reflected on  $\{taskCpt^{t+1}, \dots, taskCpt^{TimeLimit}\}$ . In addition, the optimization objective of **Problem 1** is to maximize the task completion  $\sum_m task_m^0 - \sum_m task_m^{TimeLimit}$ , which is inconsistent with the total expected immediate reward  $G^t$  in

Equation (21). Therefore, if we compute the immediate reward based on Equation (19),  $\{\sum_i mtigU_i^t, \dots, \sum_i mtigU_i^{TimeLimit}\}$  may affect the ability of agents to make optimal decisions in MANF-RL-RP algorithm. Therefore, we can calculate immediate reward based on Equation (17), and further simplify the total expected immediate reward  $G^t$  to  $G_{task}^t$ , see Equation (22), which is fitter the optimization objective in **Problem 1**. Finally, we can refer to the optimization process of standard reinforcement learning algorithms for the MANF-RL-RP algorithm, as shown in **Algorithm 2**. Additionally, it is essential to point out that setting up a target network and an evaluation network can address the issue of training instability in MANF-RL-RP. During the training process of MANF-RL-RP, using a single neural network can lead to two problems: Firstly, the target values are estimated based on the single neural network, and these estimated values may have biases. Secondly, updating the network leads to modifications in the estimated values, thereby exacerbating the disparity between the target values and the estimated values. To address these problems, MANF-RL-RP incorporates a target network and an evaluation network. The target network is periodically updated based on the evaluation network to slow down the rate of target value changes. In contrast, MANF-DNN-RP utilizes target values that are derived from objective real-world environments, ensuring stability and freedom from bias.

In addition, there are two points worth noting. (1) In the QMIX algorithm, the Q-value is decomposed into local Q-values for individual agents and a global mixed Q-value. This design is effective in scenarios with a fixed number of agents. Therefore, the MANF-RL-RP algorithm can only be applied to situations with a fixed number of UAVs, workers and cars. (2) The the MANF-RL-RP algorithm is trained based on pre-annotated static environments. Hence, it can only be applied to collaborative route planning of UAVs, workers and cars for crowdsensing in static environments.

$$G_{task}^t = taskCpt^t + \dots + \gamma^{TimeLimit-t} taskCpt^{TimeLimit} \quad (22)$$

**VI. EVALUATION****A. Data set**

We conduct experimental evaluations based on simulated data, which includes discrete area set AREA, UAVs set UAV, workers set Worker, and cars set Car. To conduct experimental evaluations more objectively and comprehensively, the following points are worth noting in the simulated data, refer to [33], [12], [7], [40]. It is worth noting that, in order to ensure the authenticity of the simulated data, we referenced assumptions about experimental data from existing representative works to cover data characteristics in disaster scenarios as comprehensively as possible. The effectiveness of the method still needs further verification in disaster scenarios.

(1) The changes in the objective world caused by disasters often lack regularity. However, influenced by human activities, the distribution of sensing tasks often has its own uniqueness. Therefore, we make the following assumptions about the

distribution of obstacles locations and sensing tasks locations. The locations of the obstacles satisfy a random distribution. The locations of the sensing tasks satisfy random distribution or check-in empirical distribution, as shown in Figure 6. Note: The check-in data records the human position in the real-world environment. The check-in empirical distribution is simulated based on check-in data [41], which represents a density distribution of human in the geographic locations.

(2) The agents (e.g., UAVs, workers and cars) used to perform sensing tasks can be centrally deployed by special department or spontaneously participated by existing participants in the environment. Therefore, we make the following assumptions about the initial agents locations. The initial locations of all agents satisfy the following three distributions, as indicated in reference [32]. (1) The initial locations of all agents are same; (2) The initial locations of all agents satisfy random distribution, as shown in Figure 6(a); (3) The initial locations of all agents satisfy the check-in empirical distribution, as shown in Figure 6(b).

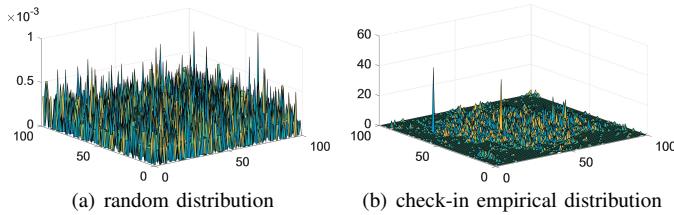


Fig. 6. Geographical distributions of agents and sensing tasks.

(3) The agents performing sensing tasks can have the same mobility (e.g., centrally deployed agents have the same configuration) or different mobilities (e.g., spontaneously participating agents have varied configurations). Therefore, the

permissible range of movement can be generated in two ways. (1) all agents have the same fixed movable radius; (2) the movable radius of each agent is randomly generated within a specified interval. It should be noted that the permissible ranges of movement for agents are determined based on the movable radius and environmental information, excluding areas with obstacles.

(4) UAVs with the same configuration have similar power. However, UAVs with different configurations will have significantly different power. Therefore, the power consumed by UAVs in a time step are generated in two ways. (1) The powers of UAVs are same; (2) The powers of UAVs are randomly set. Please note that we assume the UAV's fully charged battery is 1 kW·h.

Based on the above requirements, we simulated 8 sets of data, as shown in Table I. In addition, based on the simulated data, we set up 3 groups of experiments, as shown in Table II. There are three points worth noting. (1) The determination of spatial granularity needs to consider the mobility and sensing capabilities of agents e.g., UAVs, workers, and vehicles. In a unit of time, agents should be able to move to the target area and complete the sensing tasks. Assuming the size of each grid is  $1 \text{ km}^2$ , the experimental areas in this paper range from  $144 \text{ km}^2$  to  $400 \text{ km}^2$ . This is roughly equivalent to the urban area size of a medium to large city. (2) The scenario studied in this paper is static. When a disaster occurs, we only need to model the post-disaster scenario based on past knowledge, including the obstacles locations and sensing tasks locations. (3) In times of disaster, there is an optimal time window for executing rescue operations. During this period, rescue efforts may be more feasible and effective, contributing to saving lives or mitigating the impact of the disaster. For

TABLE I  
SIMULATED DATA

data ID	initial locations	agent number	area number	tasks number (distribution)	obstacle number	movable radius (KM)	power (KW/h)
(1)	same	10/25/5	16*16	120(random)	20	8/3/5	0.3
(2)	random	10/25/5	16*16	120(random)	20	[7,9]/[2,4]/[4,6]	[0.2,0.4]
(3)	check-in	10/25/5	16*16	120(random)	20	[7,9]/[2,4]/[4,6]	[0.2,0.4]
(4)	check-in	10/25/5	16*16	120(check-in)	20	[7,9]/[2,4]/[4,6]	[0.2,0.4]
(5)	random	(8/20/4),(10/25/5),(12/30/6)	16*16	120(random)	20	[7,9]/[2,4]/[4,6]	[0.2,0.4]
(6)	random	10/25/5	(12*12),(16*16),(20*20)	120(random)	20	[7,9]/[2,4]/[4,6]	[0.2,0.4]
(7)	random	10/25/5	16*16	100/120/140(random)	20	[7,9]/[2,4]/[4,6]	[0.2,0.4]
(8)	random	10/25/5	16*16	120(random)	20/40/60	[7,9]/[2,4]/[4,6]	[0.2,0.4]

TABLE II  
EXPERIMENTAL GROUP SETTINGS

group ID	data ID	TimeLimit	algorithm (abbreviation)
(1)	(1) / (2) / (3) / (4)	9	MANF-DNN-RP
(2)	(1) / (2) / (3) / (4)	9	MANF-DNN-RP-temp / MANF-DNN-RP / MANF-RL-RP-temp / MANF-RL-RP
(3)	(1) / (2) / (3) / (4)	6 / 9 / 12	Greedy-SC-RP / MANF-DNN-RP / MANF-RL-RP
(4)	(5) / (6) / (7) / (8)	9	Greedy-SC-RP / MANF-DNN-RP / MANF-RL-RP

example, based on the common knowledge of earthquake relief [1], after an earthquake occurs, the survival probability of survivors is approximately 90% on the first day, but it decreases significantly to around 50%-60% on the second day. Therefore, urgent sensing tasks such as life detection need to be completed as quickly as possible within the initial hours to provide necessary assistance for subsequent rescue operations. Assuming that the time required to complete one sensing task is one hour, setting *TimeLimit* as 6/9/12 in this paper essentially ensures the completion of sensing tasks in a short time, thereby allowing ample time for subsequent rescue operations.

### B. Experiment Setup

Since neural networks are not the focus of our research, we use the same neural network in the control experiments. Study [42] has shown that a layer of the neural network can fit any function. Therefore, all methods in this paper use neural networks with one hidden layer, and the hidden layer nodes are 10 times as large as the input layer nodes. In addition, the spatial convolutional network *cnnSpace* only contains one convolutional layer and one average pooling layer. All activation functions are *relu()* in this paper. The other experimental hyperparameters are shown in Table III. It is significant to note that the earlier the sensing tasks are completed, the better. Therefore, according to experience, we set  $\gamma$  to a smaller value to pay attention to the sensing tasks completion at the current moment as much as possible.

TABLE III  
EXPERIMENTAL HYPERPARAMETERS

learning rate	$\gamma$	$P$	$M$	$\varepsilon$
0.0001	0.7	200	5000	1/0.1/32
batch size	optimizer	/	in_channels	out_channels
32	RMSprop	/	5	10
kernel_size	stride	padding	dilation	Pool_size
3	1	1	1	2

### C. Baselines and evaluation indicator

Baselines are as follows.

(1) We evaluated the most relevant work [43], [18]. Given that our research problem differs from existing works, we have made modifications to these approaches in order to address our specific problem, resulting in the development of the Greedy-SC-RP algorithm. The Greedy-SC-RP algorithm employs a greedy approach to sequentially plan routes for UAVs, workers, and cars. The implementation process of the algorithm can be outlined in the following three steps.

First, calculate the sum of Euclidean distances  $Dis_{ii}^t$  between different locations  $locOpt_{ii}^t$  within the optional range  $agentArriv_i^t$  of the UAV  $uav_i^t$  and all locations of sensing tasks  $task_m$ , ( $m = 1, 2, \dots$ ), as shown in Equation (23). The function "ED()" denotes the calculation of the Euclidean distance between two locations. Select the location with the minimum Euclidean distance  $Dis_{ii}^t$  as the location for the UAV

$uav_i^{t+1}$  within the time step  $[t, t+1]$ . Repeat the above process until the next location for all UAVs are computed.

$$Dis_{ii}^t = \sum_m ED(locOpt_{ii}^t, task_m), locOpt_{ii}^t \in agentArriv_i^t \quad (23)$$

Then, calculate the sum of Euclidean distances  $Dis_{jj}^t$  between different locations  $locOpt_{jj}^t$  within the optional range  $agentArriv_j^t$  of the worker  $wkr_j^t$  and all locations of sensing tasks  $task_m$ , ( $m = 1, 2, \dots$ ), see Equation (24). Select the location with the minimum Euclidean distance  $Dis_{jj}^t$  as the location for the worker  $wkr_j^{t+1}$  within the time step  $[t, t+1]$ . Repeat the above process until the next location for all workers are computed.

$$Dis_{jj}^t = \sum_m ED(locOpt_{jj}^t, task_m), locOpt_{jj}^t \in agentArriv_j^t \quad (24)$$

Finally, calculate the sum of Euclidean distances  $Dis_{kk}^t$  between different locations  $locOpt_{kk}^t$  within the optional range  $agentArriv_k^t$  of the car  $car_k^t$  and locations of UAVs  $uav_i^{t+1}$ , ( $i = 1, 2, \dots$ ) at moment  $t + 1$ , see Equation (25). Select the location with the minimum Euclidean distance  $Dis_{kk}^t$  as the location for the car  $car_k^{t+1}$  within the time step  $[t, t+1]$ . Repeat the above process until the next location for all cars are computed.

$$Dis_{kk}^t = \sum_i ED(locOpt_{kk}^t, uav_i^{t+1}), locOpt_{kk}^t \in agentArriv_k^t \quad (25)$$

(2) MANF-DNN-RP-temp: Remove the spatial convolutional network *cnnSpace* in the MANF-DNN-RP algorithm. MANF-DNN-RP refers to the most relevant and up-to-date work on the implementation of drone or vehicle scheduling [7], [11].

(3) MANF-RL-RP-temp: Change the expected immediate reward from  $taskCpt^t$  to  $taskCpt^t + \sum_i mtigU_i^t$  in the MANF-RL-RP algorithm. The core idea of MANF-RL-RP is similar to some of the latest research work, where the focus is on training multiple agents to collaboratively work in complex environments, aiming to maximize the cumulative rewards they obtain [44], [45], [46]. However, the methods proposed in these works are challenging to apply to address the problem in this paper. These works primarily deal with the scheduling of multiple homogeneous agents and do not involve the collaborative scheduling of heterogeneous agents.

According to **Problem 1**, we use the task completion rate within the time step  $[0, TimeLimit]$  as evaluation indicator, see Equation (26).

$$taskCptRate = \frac{\sum_m task_m^0 - \sum_m task_m^{TimeLimit}}{\sum_m task_m^0} \quad (26)$$

### D. Experiment Results

1) *Measuring the weights " $\alpha$ " and " $\beta$ "*: The experimental setting refers to group (1) in Table II. The experimental results

are shown in Table IV. Overall, when " $\alpha=0.5/\beta=0.5$ ", the MANF-DNN-RP algorithm performs better in our experimental scenario. Therefore, this paper sets " $\alpha=0.5/\beta=0.5$ " for experimentation. In fact, the setting of " $\alpha$ " and " $\beta$ " may be related to various factors, such as the ratio of agents (e.g., UAVs, workers, and cars), the distribution of tasks, the consumption of UAV battery in unit moment, etc. Under different conditional assumptions, we may need to assign different weights to " $\alpha$ " and " $\beta$ ". On the other hand, the MANF-RL-RP algorithm does not need to consider the setting of " $\alpha$ " and " $\beta$ ", and it has better applicability.

TABLE IV  
THE TASK COMPLETION RATE OF THE MANF-DNN-RP ALGORITHM  
UNDER DIFFERENT WEIGHT SETTINGS.

dataID	$\alpha=0.4/\beta=0.6$	$\alpha=0.45/\beta=0.55$	$\alpha=0.5/\beta=0.5$	$\alpha=0.55/\beta=0.45$	$\alpha=0.6/\beta=0.4$
(1)	0.4417	0.4833	0.4917	0.4750	0.4500
(2)	0.4667	0.5083	0.5083	0.5000	0.4750
(3)	0.5333	0.5917	0.5833	0.5667	0.5750
(4)	0.5250	0.5500	0.5583	0.5417	0.5333

2) *Verifying the improvement of methods:* The experimental setting refers to group (2) in Table II. The experimental results are shown in Table V. Compared to MANF-DNN-RP-temp, the utilization of MANF-DNN-RP, which expands the spatial convolutional network cnnSpace to extract spatial features of global information, leads to a substantial enhancement in the average task completion rate by 2.50%. Introducing spatial convolutional neural networks effectively extracts and preserves the two-dimensional spatial distribution features of the original global variables. On the other hand, algorithms that do not utilize spatial convolutional networks forcibly transform the two-dimensional distribution of global variables into one-dimensional vectors, making it difficult to capture and retain the inherent two-dimensional distribution characteristics of the global variables. When comparing MANF-RL-RP-temp to MANF-RL-RP, the simplification of the expected immediate reward from  $taskCpt^t + \sum_i mtigU_i^t$  to  $taskCpt^t$  brings about a significant increase in the average task completion rate by 12.71%. The impact of the cars on the task completion rate is delayed, referring to **Lemma 3**. Replacing the UAVs' batteries with the cars at the current moment can reduce the occurrence of the UAVs halting operations in future moments due to insufficient power. In other words,  $\sum_i mtigU_i^t$  will be reflected on  $\{taskCpt^{t+1}, \dots, taskCpt^{TimeLimit}\}$ . In addition, the optimization objective of **Problem 1** is to maximize the task completion  $\sum_m task_m^0 - \sum_m task_m^{TimeLimit}$ , which is inconsistent with the total expected immediate reward  $G^t$  in Equation (21). Therefore, if we compute the immediate reward based on  $taskCpt^t + \sum_i mtigU_i^t$ ,  $\{\sum_i mtigU_i^t, \dots, \sum_i mtigU_i^{TimeLimit}\}$  may affect the ability of agents to make optimal decisions in MANF-RL-RP algorithm. Therefore, we can calculate immediate reward based on  $taskCpt^t$ .

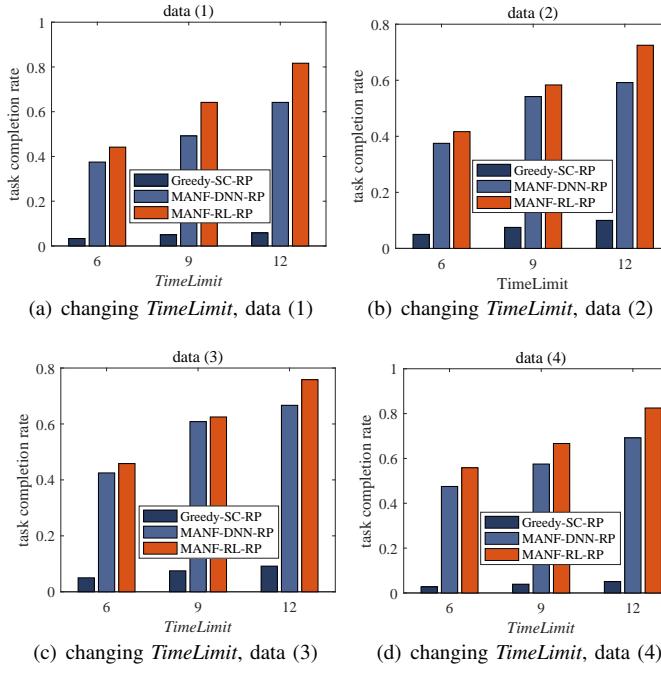
3) *Changing TimeLimit under the different data distribution:* The experimental setting refers to group (3) in Table

TABLE V  
PERFORMANCE COMPARISON BASED ON DIFFERENT DATA

dataID	MANF-DNN-RP-temp	MANF-DNN-RP	MANF-RL-RP-temp	MANF-RL-RP
(1)	0.4667	0.4917	0.4500	0.6417
(2)	0.5083	0.5417	0.4250	0.5833
(3)	0.5833	0.6083	0.5417	0.6250
(4)	0.5583	0.5750	0.5917	0.6667

II. The experimental results are shown in Figure 7. With the increase of upper limit of sensing time  $TimeLimit$ , the task completion rate of all methods is gradually increasing. The increase in  $TimeLimit$  indicates that UAVs, workers, and cars can spend more time executing the sensing tasks, and more time results in higher task completion rate. In addition, under different  $TimeLimit$ , the task completion rate of MANF-RL-RP is significantly higher than that of MANF-DNN-RP and Greedy-SC-RP. When  $TimeLimit$  is 6, the task completion rate is increased by 5.63% and 42.86% on average, respectively. When  $TimeLimit$  is 9, the task completion rate is increased by 7.50% and 56.94% on average, respectively. When  $TimeLimit$  is 12, the task completion rate is increased by 13.33% and 70.60% on average, respectively. Since Greedy-SC-RP greedily plans routes for agents in turn (i.e., UAVs, workers and cars), it is difficult to capture the collaboration between agents, and does not perform well. Next, MANF-DNN-RP is essentially a greedy idea, which gives priority to the best cooperation of all agents at the current moment, regardless of the long-term impact of the current choice on subsequent decisions. MANF-RL-RP is implemented based on reinforcement learning, which can take into account the long-term impact of current choices on subsequent decisions. For a more detailed analysis, please refer to Section VI-E.

4) *Changing the number of agents, areas, sensing tasks or obstacles:* The experimental setting refers to group (4) in Table II. The experimental results are shown in Figure 8. With the increase of agents (i.e., UAVs, workers and cars), the task completion rate of all methods is gradually increasing, as shown in Figure 8(a). The increase of agents indicates that there are more UAVs, workers and cars to perform sensing tasks simultaneously, which can complete more sensing tasks per unit time, finally leading to a higher task completion rate. With the increase of the number of areas, the task completion rate of all methods is gradually decreasing, as shown in Figure 8(b). The increase of areas leads to a more sparse distribution of sensing tasks, and the mobility of agents is limited, which inevitably leads to lower task completion rates. With the increase of the number of sensing tasks, the task completion rate of all methods is gradually decreasing, as shown in Figure 8(c). There is an upper limit to the number of sensing tasks that a given number of agents can complete within a limited time. With the increase of the number of obstacles, there is a slight increase in the task completion rate of all methods, as shown in Figure 8(d). The increase of obstacles reduces the sparsity of sensing task distribution. The high density distribution of sensing tasks is conducive to increasing the task completion

Fig. 7. Changing *TimeLimit* under the different data distribution.

rate.

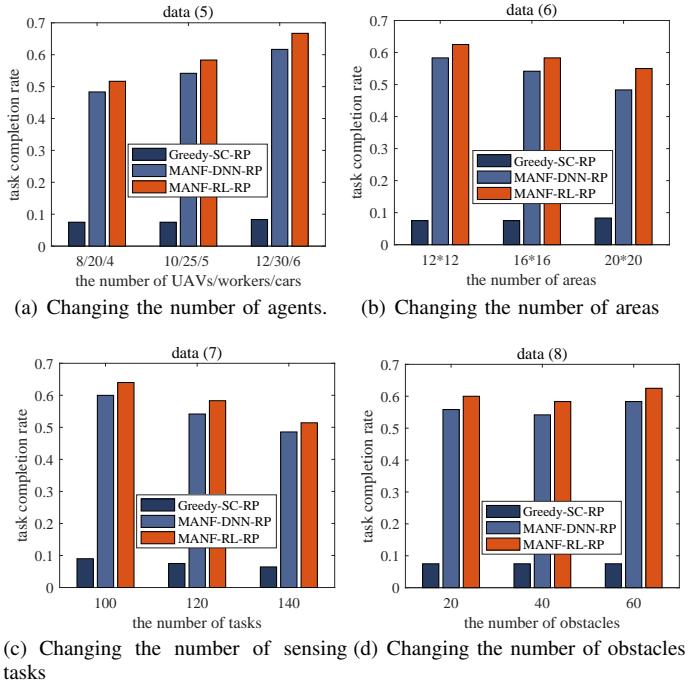


Fig. 8. Changing the number of agents, areas, sensing tasks or obstacles

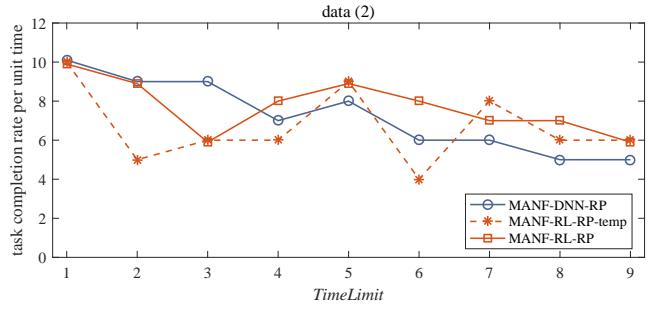
## E. Analysis and Discussion

Since Greedy-SC-RP greedily plans routes for agents in turn (i.e., UAVs, workers and cars), it is difficult to capture the collaboration between agents, and does not perform well. Compared with MANF-DNN-RP-temp, MANF-DNN-RP expands the spatial convolutional network *cnnSpace* to extract spatial features of global information, and performs better

[11], [12], [7], [8], [9]. Next, we will compare MANF-RL-RP, MANF-DNN-RP and MANF-RL-RP-temp to illustrate the advantages of MANF-RL-RP.

1) *MANF-RL-RP VS. MANF-DNN-RP*: To explain the difference between MANF-RL-RP and MANF-DNN-RP more clearly, we need to calculate the task completion rate per unit time *taskCptRatePer<sup>t</sup>*, see Equation (27).

$$\text{taskCptRatePer}^t = \sum_m \text{task}_m^t - \sum_m \text{task}_m^{t+1}, t \in [1, \text{TimeLimit}] \quad (27)$$

Fig. 9. *taskCpt<sup>t</sup>* at different moments, *TimeLimit* = 9.

MANF-RL-RP performs better than MANF-DNN-RP mainly for two reasons. (1) MANF-DNN-RP is essentially a greedy idea, which gives priority to the best cooperation of all agents at the current moment, regardless of the long-term impact of the current choice on subsequent decisions. MANF-RL-RP is implemented based on reinforcement learning, which can take into account the long-term impact of current choices on subsequent decisions. As shown in Figure 9, the task completion rate per unit time *taskCptRatePer<sup>t</sup>* of MANF-DNN-RP is slightly higher than *taskCptRatePer<sup>t</sup>* of MANF-RL-RP at  $t \in [1, 3]$ , and then significantly worse than *taskCptRatePer<sup>t</sup>* of MANF-RL-RP at  $t \in [4, 9]$ . (2) However, MANF-RL-RP-temp is also implemented based on reinforcement learning. As shown in Figure 9, Why is the task completion rate per unit time *taskCptRatePer<sup>t</sup>* of MANF-RL-RP-temp almost always worse than *taskCptRatePer<sup>t</sup>* of MANF-RL-RP? To take into account the promotion of cars on carrying out sensing tasks, MANF-DNN-RP uses  $\text{taskCpt}^t + \sum_i m_{\text{tig}} U_i^t$  as the greed indicator, which is not perfect fit with the optimization goal of **Problem 1**. MANF-RL-RP uses  $\text{taskCpt}^t$  to estimate the expected immediate reward, which can well fit the optimization goal of **Problem 1**. For detailed explanation, please refer to Section VI-E2.

2) *MANF-RL-RP VS. MANF-RL-RP-temp*: Table VI records the remaining power of UAVs at different moments, in which cyan mark indicates that the batteries of the UAVs have been replaced. Based on Table VI, we can get two differences between MANF-RL-RP-temp and MANF-RL-RP. (1) When the batteries of UAVs are replaced, the power of UAVs in MANF-RL-RP-temp is usually higher than that in MANF-RL-RP. (2) The battery replacement frequency of UAVs in MANF-RL-RP-temp (i.e., 42 times) is higher than that of UAVs in MANF-RL-RP (i.e., 30 times). We should replace the batteries of UAVs without affecting performing

the sensing tasks, rather than replacing their batteries when the power of UAVs is still high. In addition, frequently meeting cars to replace the batteries of UAVs may seriously affect the efficiency of UAVs in performing sensing tasks. Therefore, it is better to choose  $\text{taskCpt}^t$  (i.e., MANF-RL-RP) to calculate the expected immediate reward than  $\text{taskCpt}^t + \sum_i m_{\text{tig}} U_i^t$  (i.e., MANF-RL-RP-temp) in **Problem 1**.

*3) Computational complexity of MANF-DNN-RP and MANF-RL-RP:* Based on data (2), we conducted a comprehensive analysis of the training process for MANF-DNN-RP and MANF-RL-RP. Figure 10 depicts the variations in the loss value and task completion rate as the number of data iterations increases during model training. Based on Figure 10, we can observe two distinct characteristics in the curve. (1) Figure 10(a) illustrates that within the first 4000 iterations, the training loss values of the two methods had already reached a lower level. However, as depicted in Figure 10(b), the convergence speed of the two models slows down after this point without reaching a convergence point. The reason behind this is that, at this stage, the two models have not fully completed the detection of the environmental space. They are exclusively trained using local detection results, which limits their ability to make optimal decisions in response to the overall environment. (2) Based on the observations from Figure 10(b), it is evident that MANF-RL-RP achieves faster convergence and demonstrates enhanced stability compared to MANF-DNN-RP. Specifically, MANF-DNN-RP exhibits higher levels of fluctuation, whereas MANF-RL-RP exhibits comparatively lower fluctuations once converged. However, in relation to **Algorithm 1** and **Algorithm 2**, MANF-DNN-RP and MANF-RL-RP have the same time complexity. What factors contribute to this discrepancy? The reason behind this is that, during the training process, the divergence in training data contributes to the observed disparities between the two models. Because the MANF-DNN-RP model primarily emphasizes immediate rewards, the variations in rewards per unit of time are not substantial. As a result, the distinguishability of training data labels becomes less evident, presenting a challenge for model fitting. In contrast, MANF-RL-RP places emphasis on long-term decision-making benefits and demonstrates notable variations across different durations of the decision-making process. As a result, the distinguishability of training data labels becomes relatively prominent, facilitating smoother model fitting. Consequently, MANF-RL-RP achieves faster convergence and greater stability after convergence, in contrast to MANF-DNN-RP.

## VII. CONCLUSION

Devastating disasters (e.g., earthquake) are extremely destructive. Efficiently obtaining the up-to-date information in the disaster-stricken area is the key to successful disaster response. UAVs, workers and cars can collaborate to complete the sensing tasks (e.g., data collection) in disaster-stricken areas. In this paper, we explicitly consider planning the routes of a group of agents (i.e., UAVs, workers, and cars) to maximize the task completion rate. We propose a heterogeneous multi-agent route planning algorithm MANF-RL-RP, which has the

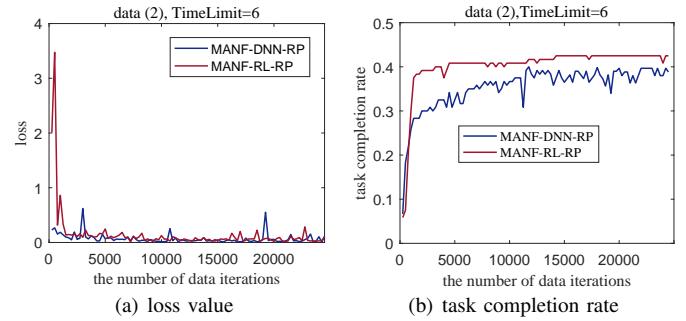


Fig. 10. The variations of loss value and task completion rate as the number of data iterations increases.

following design. (a) Global-local dual information processing. First, we mine the spatial features of global information based on convolutional neural networks (CNN) and share them with all agents to reduce the model training cost. Then, we divide the local information of agents into two parts: state information and filtering information. State information is used to guide the agents to make sequential decision. Filtering information is used to filter the non-optimal actions to address the issue of sparse rewards in the sequential decision-making process. (b) Model structure for heterogeneous multi-agent. We fill in the missing information of workers and cars to use the same data structure to represent the state of UAVs, workers, and cars, then share the same neural network parameter to reduce model parameter scale. Furthermore, we design a reasonable reward function and prove that UAVs, workers, and cars have cooperative relationships, which can guide model training well. In addition, we prove that the sequential decision-making process of agents has the Markov property, which simplifies the agent network structure. Finally, we conducted detailed experiments based on the rich simulation data. In comparison to the baseline algorithms, namely Greedy-SC-RP and MANF-DNN-RP, MANF-RL-RP has exhibited a significant improvement in terms of task completion rate. Under different *TimeLimit*, the task completion rate of MANF-RL-RP is significantly higher than that of MANF-DNN-RP and Greedy-SC-RP. When *TimeLimit* is 6, the task completion rate is increased by 5.63% and 42.86% on average, respectively. When *TimeLimit* is 9, the task completion rate is increased by 7.50% and 56.94% on average, respectively. When *TimeLimit* is 12, the task completion rate is increased by 13.33% and 70.60% on average, respectively.

## ACKNOWLEDGMENT

This work was supported in part by the National Natural Science Foundation of China (No. 61960206008, No. 62032020, No. U2001207, No.62332014) and Natural Science Foundation of Shaanxi Province for Distinguished Young Scholars (No. 2023-JC-JQ-54). This work was also sponsored by the Innovation Foundation for Doctor Dissertation of Northwestern Polytechnical University(CX2022017).

## REFERENCES

- [1] <https://en.wikipedia.org/wiki/Earthquake>, 2023.

TABLE VI  
POWER OF UAVS AT DIFFERENT MOMENTS,  $TimeLimit = 9$ .

method (data (2))		MANF-RL-RP-temp									MANF-RL-RP								
UAV's ID	moment $t$	1	2	3	4	5	6	7	8	9	1	2	3	4	5	6	7	8	9
1	1	0.71	1	1	0.71	0.42	1	0.71	0.42	1	0.71	1	0.71	1	1	1	0.71	1	0.71
2	2	0.64	1	0.64	1	0.64	1	1	0.64	1	0.64	0.28	0.28	0.28	0.28	1	0.64	1	0.64
3	3	0.65	1	0.65	1	0.65	1	0.65	1	0.65	0.65	0.30	1	0.65	0.30	0.30	1	0.65	0.30
4	4	0.67	1	0.67	1	0.67	1	1	1	0.67	0.67	0.34	1	0.67	0.34	1	1	0.67	1
5	5	0.61	1	0.61	1	0.61	1	0.61	1	0.61	0.61	0.22	0.22	1	0.61	0.22	0.22	1	0.61
6	6	0.74	1	0.74	0.48	1	1	0.74	0.48	1	0.74	0.48	0.22	0.22	0.22	1	0.74	0.48	0.22
7	7	0.69	1	1	0.69	0.38	1	0.69	0.38	1	0.69	0.38	1	0.69	1	0.69	1	0.69	1
8	8	0.66	1	0.66	1	0.66	1	0.66	1	1	0.66	1	1	1	0.66	1	0.66	0.32	0.32
9	9	0.65	1	0.65	1	0.65	1	0.65	1	0.65	0.65	0.30	0.30	1	0.65	0.30	0.30	1	0.65
10	10	0.70	1	0.70	0.40	1	0.70	0.40	1	0.70	1	1	0.70	0.40	1	0.70	1	0.70	0.40

- [2] R. K. Ganti, Y. Fan, and H. Lei, "Mobile crowdsensing: current state and future challenges," *IEEE Communications Magazine*, vol. 49, no. 11, pp. 32–39, 2011.
- [3] P. Dutta, P. M. Aoki, N. Kumar, A. M. Mainwaring, and A. Woodruff, "Common sense: Participatory urban sensing using a network of handheld air quality monitors," in *International Conference on Embedded Networked Sensor Systems*, 2009.
- [4] R. Lee, S. Wakamiya, and K. Sumiya, "Discovery of unusual regional social activities using geo-tagged microblogs," *World Wide Web-internet & Web Information Systems*, vol. 14, no. 4, pp. 321–349, 2011.
- [5] None, "How long to wait? predicting bus arrival time with mobile phone based participatory sensing," *IEEE Transactions on Mobile Computing*, vol. 13, no. 6, pp. 1228–1241, 2014.
- [6] Z. Zhou, J. Feng, B. Gu, B. Ai, S. Mumtaz, J. Rodriguez, and M. Guizani, "When mobile crowd sensing meets uav: Energy-efficient task assignment and route planning," *IEEE Transactions on Communications*, vol. 66, no. 11, pp. 5526–5538, 2018.
- [7] C. H. Liu, Z. Chen, and Y. Zhan, "Energy-efficient distributed mobile crowd sensing: A deep learning approach," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 6, pp. 1262–1276, 2019.
- [8] H. Wang, C. H. Liu, Z. Dai, J. Tang, and G. Wang, "Energy-efficient 3d vehicular crowdsourcing for disaster response by distributed deep reinforcement learning," in *Proceedings of the 27th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2021, pp. 3679–3687.
- [9] C. H. Liu, Y. Zhao, Z. Dai, Y. Yuan, G. Wang, D. Wu, and K. K. Leung, "Curiosity-driven energy-efficient worker scheduling in vehicular crowdsourcing: A deep reinforcement learning approach," in *2020 IEEE 36th International Conference on Data Engineering (ICDE)*. IEEE, 2020, pp. 25–36.
- [10] Z. Dai, C. H. Liu, Y. Ye, R. Han, Y. Yuan, G. Wang, and J. Tang, "Aoi-minimal uav crowdsensing by model-based graph convolutional reinforcement learning," in *IEEE INFOCOM 2022-IEEE Conference on Computer Communications*. IEEE, 2022, pp. 1029–1038.
- [11] C. H. Liu, C. Piao, and J. Tang, "Energy-efficient uav crowdsensing with multiple charging stations by deep learning," in *IEEE INFOCOM 2020-IEEE Conference on Computer Communications*. IEEE, 2020, pp. 199–208.
- [12] C. H. Liu, X. Ma, X. Gao, and J. Tang, "Distributed energy-efficient multi-uav navigation for long-term communication coverage by deep reinforcement learning," *IEEE Transactions on Mobile Computing*, vol. 19, no. 6, pp. 1274–1285, 2019.
- [13] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton *et al.*, "Mastering the game of go without human knowledge," *nature*, vol. 550, no. 7676, pp. 354–359, 2017.
- [14] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski *et al.*, "Human-level control through deep reinforcement learning," *nature*, vol. 518, no. 7540, pp. 529–533, 2015.
- [15] Y. Zhao and Q. Han, "Spatial crowdsourcing: current state and future directions," *IEEE communications magazine*, vol. 54, no. 7, pp. 102–107, 2016.
- [16] Y. Tong, J. She, B. Ding, L. Wang, and L. Chen, "Online mobile micro-task allocation in spatial crowdsourcing," in *2016 IEEE 32Nd international conference on data engineering (ICDE)*. IEEE, 2016, pp. 49–60.
- [17] B. Li, Y. Cheng, Y. Yuan, G. Wang, and L. Chen, "Simultaneous arrival matching for new spatial crowdsourcing platforms," in *Proceedings of the 29th International Conference on International Joint Conferences on Artificial Intelligence*, 2021, pp. 1279–1287.
- [18] ———, "Three-dimensional stable matching problem for spatial crowdsourcing platforms," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2019, pp. 1643–1653.
- [19] <https://github.com/CocaColaZero/MANF-for-Collaborative-Route-Planning.git>, 2023.
- [20] S. Reddy, K. Shilton, J. Burke, D. Estrin, M. Hansen, and M. Srivastava, "Using context annotated mobility profiles to recruit data collectors in participatory sensing," in *International Symposium on Location-and Context-Awareness*. Springer, 2009, pp. 52–69.
- [21] D. Zhang, H. Xiong, L. Wang, and G. Chen, "Crowdrecruiter: Selecting participants for piggyback crowdsensing under probabilistic coverage constraint," in *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, 2014, pp. 703–714.
- [22] H. Xiong, D. Zhang, G. Chen, L. Wang, V. Gauthier, and L. E. Barnes, "icrowd: Near-optimal task allocation for piggyback crowdsensing," *IEEE Transactions on Mobile Computing*, vol. 15, no. 8, pp. 2010–2022, 2015.
- [23] Z. Song, B. Zhang, C. H. Liu, A. V. Vasilakos, J. Ma, and W. Wang, "Qoi-aware energy-efficient participant selection," in *2014 Eleventh Annual IEEE International Conference on sensing, communication, and networking (SECON)*. IEEE, 2014, pp. 248–256.
- [24] M. Karaliopoulos, O. Telelis, and I. Koutsopoulos, "User recruitment for mobile crowdsensing over opportunistic networks," in *2015 IEEE Conference on Computer Communications (INFOCOM)*. IEEE, 2015, pp. 2254–2262.
- [25] Z. Yu, J. Zhou, W. Guo, L. Guo, and Z. Yu, "Participant selection for t-sweep k-coverage crowd sensing tasks," *World Wide Web*, vol. 21, no. 3, pp. 741–758, 2018.
- [26] L. Wang, Z. Yu, Q. Han, B. Guo, and H. Xiong, "Multi-objective optimization based allocation of heterogeneous spatial crowdsourcing tasks," *IEEE Transactions on Mobile Computing*, vol. 17, no. 7, pp. 1637–1650, 2017.
- [27] M. Zhang, P. Yang, C. Tian, S. Tang, X. Gao, B. Wang, and F. Xiao, "Quality-aware sensing coverage in budget-constrained mobile crowdsensing networks," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 9, pp. 7698–7707, 2015.
- [28] J. Wang, Y. Wang, D. Zhang, F. Wang, H. Xiong, C. Chen, Q. Lv, and Z. Qiu, "Multi-task allocation in mobile crowd sensing with individual task quality assurance," *IEEE Transactions on Mobile Computing*, vol. 17, no. 9, pp. 2101–2113, 2018.
- [29] H. Li, T. Li, and Y. Wang, "Dynamic participant recruitment of mobile crowd sensing for heterogeneous sensing tasks," in *2015 IEEE 12th International Conference on Mobile Ad Hoc and Sensor Systems*. IEEE, 2015, pp. 136–144.

- [30] L. Wang, Z. Yu, D. Zhang, B. Guo, and C. H. Liu, "Heterogeneous multi-task assignment in mobile crowdsensing using spatiotemporal correlation," *IEEE Transactions on Mobile Computing*, vol. 18, no. 1, pp. 84–97, 2018.
- [31] E. Wang, Y. Yang, and K. Lou, "User selection utilizing data properties in mobile crowdsensing," *Information Sciences*, vol. 490, pp. 210–226, 2019.
- [32] L. Han, Z. Yu, Z. Yu, L. Wang, H. Yin, and B. Guo, "Online organizing large-scale heterogeneous tasks and multi-skilled participants in mobile crowdsensing," *IEEE Transactions on Mobile Computing*, 2021.
- [33] C. H. Liu, Z. Dai, H. Yang, and J. Tang, "Multi-task-oriented vehicular crowdsensing: A deep learning approach," in *IEEE INFOCOM 2020-IEEE Conference on Computer Communications*. IEEE, 2020, pp. 1123–1132.
- [34] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher, "An analysis of approximations for maximizing submodular set functions—i," *Mathematical programming*, vol. 14, no. 1, pp. 265–294, 1978.
- [35] J. Hare, "Dealing with sparse rewards in reinforcement learning," *arXiv preprint arXiv:1910.09281*, 2019.
- [36] T. Rashid, M. Samvelyan, C. Schroeder, G. Farquhar, J. Foerster, and S. Whiteson, "Qmix: Monotonic value function factorisation for deep multi-agent reinforcement learning," in *International conference on machine learning*. PMLR, 2018, pp. 4295–4304.
- [37] D. Bertsekas and J. N. Tsitsiklis, *Neuro-dynamic programming*. Athena Scientific, 1996.
- [38] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.
- [39] D. Ha, A. Dai, and Q. V. Le, "Hypernetworks," *arXiv preprint arXiv:1609.09106*, 2016.
- [40] L. Han, Z. Yu, Z. Yu, L. Wang, H. Yin, and B. Guo, "Online organizing large-scale heterogeneous tasks and multi-skilled participants in mobile crowdsensing," *IEEE Transactions on Mobile Computing*, pp. 1–1, 2021.
- [41] E. Cho, S. A. Myers, and J. Leskovec, "Friendship and mobility: user movement in location-based social networks," in *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Diego, CA, USA, August 21-24, 2011*, 2011.
- [42] G. Cybenko, "Approximation by superpositions of a sigmoidal function," *Mathematics of Control, Signals and Systems*, vol. 2, no. 4, pp. 303–314, 1989.
- [43] L. Wang, D. Yang, Z. Yu, F. Xiong, L. Han, S. Pan, and B. Guo, "Task scheduling in three-dimensional spatial crowdsourcing: A social welfare perspective," *IEEE Transactions on Mobile Computing*, 2022.
- [44] Y. Ye, C. H. Liu, Z. Dai, J. Zhao, Y. Yuan, G. Wang, and J. Tang, "Exploring both individuality and cooperation for air-ground spatial crowdsourcing by multi-agent deep reinforcement learning," in *2023 IEEE 39th International Conference on Data Engineering (ICDE)*. IEEE, 2023, pp. 205–217.
- [45] Y. Zhao, K. Wu, Z. Xu, Z. Che, Q. Lu, J. Tang, and C. H. Liu, "Cadre: A cascade deep reinforcement learning framework for vision-based autonomous urban driving," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 3, 2022, pp. 3481–3489.
- [46] Y. Wang, C. H. Liu, C. Piao, Y. Yuan, R. Han, G. Wang, and J. Tang, "Human-drone collaborative spatial crowdsourcing by memory-augmented and distributed multi-agent deep reinforcement learning," in *2022 IEEE 38th International Conference on Data Engineering (ICDE)*. IEEE, 2022, pp. 459–471.



**Lei Han** received the Ph.D. degree in computer science from Northwestern Polytechnical University, Xi'an, China, in 2023. He currently holds the position of Postdoctoral Researcher at Xidian University. His research interests include ubiquitous computing, mobile crowd sensing, and data mining.



**Chunyu Tu** Chunyu Tu received the master's degree in computer technology from Fuzhou university, Fuzhou, China. He is currently pursuing the Ph.D. degree in computer science and technology. His research interests include pervasive computing and mobile crowd sensing.



**Zhiwen Yu** received the M.E. and Ph.D. degrees from Northwestern Polytechnical University, Xi'an, China in 2003 and 2005, respectively. He is a professor in the School of Computer Science, Northwestern Polytechnical University. He visited the Institute of Information and Communication in Singapore from 2004 to 2005. From 2006 to 2009, he was a postdoctoral researcher at Nagoya University and a special researcher at Kyoto University in Japan. From November 2009 to October 2010, he was funded by the German Humboldt Foundation and went to the University of Mannheim in Germany for collaborative research. His current research interests include pervasive computing, mobile crowdsensing, internet of things, and intelligent information technology.



**Zhiyong Yu** received the M.E. and Ph.D. degrees in computer science and technology from Northwestern Polytechnical University, Xi'an, China in 2007 and 2011, respectively. He is an associate professor in the College of Mathematics and Computer Science, Fuzhou University, Fuzhou, China. He was also a visiting student at Kyoto University, Kyoto, Japan, from 2007 to 2009 and a visiting researcher at the Institut Mines-Telecom, TELECOM SudParis, Evry, France, from 2012 to 2013. His current research interests include pervasive computing, mobile social networks, and mobile crowd sensing.



**Weihua Shan** received the bachelor's degree in Computer Science and Technology from Northwestern Polytechnical University, Xi'an, China in June 2005. He is currently the deputy chief expert of Huawei cloud technology. He is engaged in research on cutting-edge technologies related to cloud computing, including Cloud-edge-device content distribution and scheduling, cloud physical engine, metaverse simulation, and multi-agent interaction.



**Liang Wang** received the Ph.D. degree in computer science from Shenyang Institute of Automation (SIA), Chinese Academy of Sciences, Shenyang, China, in 2014. He is currently an Associate Professor with Northwestern Polytechnical University, Xi'an, China. His research interests include ubiquitous computing, mobile crowd sensing, and data mining.



**Bin Guo** received the Ph.D. degree in computer science from Keio University, Minato, Japan, in 2009. He was a Postdoctoral Researcher with the Institut TELECOM SudParis, Essonne, France. He is currently a Professor with Northwestern Polytechnical University, Xi'an, China. His research interests include ubiquitous computing, mobile crowd sensing, and HCI.