ISIATasker: Task Allocation for Instant-Sensing-Instant-Actuation Mobile Crowd Sensing

Houchun Yin, Zhiwen Yu, Liang Wang, Jiangtao Wang, Lei Han, and Bin Guo

DOI 10.1109/JIOT.2021.3095160
ISSN 2327-4662

Publisher: IEEE

ISIATasker: Task Allocation for Instant-Sensing-Instant-Actuation Mobile Crowd Sensing

Houchun Yin, Zhiwen Yu, *Senior Member, IEEE*, Liang Wang, Jiangtao Wang, Lei Han, Bin Guo, *Senior Member, IEEE*

***Abstract*—Task allocation is a key issue in Mobile Crowd Sensing (MCS), which affects the sensing efficiency and quality. Previous studies focus on the allocation of tasks that have already been published to the platform, but there are some very urgent tasks that need to be executed once they were detected. Existing studies for either delay-tolerant or time-sensitive tasks have a certain time delay from task publishing to execution, so it is impossible to achieve task detection then execution seamlessly. Thus, we first define the Instant Sensing and then Instant Actuation (ISIA) problem in MCS and propose a new model to solve it. We aim to allocate POIs where ISIA tasks are most likely to be detected to workers with similar sensing types so that these tasks can be executed once they are detected. This paper presents a two-phase task allocation framework called ISIATasker. In the sensing locations clustering and sensor selection phase, we cluster independent sensing locations into several POIs and then select the optimal cooperative sensor set for each POI to assist workers in completing sensing. In the POIs allocation phase, we propose a method called PA-DDQN based on deep reinforcement learning to plan an optimal path for each worker, thus maximize the overall sensing type matching degree and POI coverage to enable instant sensing and then instant actuation. Finally, extensive experiments are conducted based on real-world datasets to demonstrate that the matching degree and POI coverage of ISIATasker outperforms other baselines.**

***Index Terms*—Mobile Crowd Sensing, task allocation, deep reinforcement learning, task urgency.**

I. Introduction

With the proliferation of smart mobile devices, Mobile Crowd Sensing (MCS) [1] [2] has become a new sensing paradigm. Many MCS application platforms have been developed for academic research, industrial production or daily life, such as CrowdOS [3], GeoCrowd [4], Gigwalk [5], CommonSense [6] and FlierMeet [7]. Workers on these MCS platforms can sense and collect real-time data with their mobile devices (e.g., air quality [8], traffic information [9]).

Task allocation is a key problem in mobile crowd sensing [10]. Existing studies can be divide into two categories [11] based on the urgency of tasks: delay-tolerant task and time-sensitive task. *Delay-tolerant* task do not need to be executed quickly. Workers can piggyback to achieve execution during their daily routes [12] [13] [14]. The MCS platform also does not need to know the privacy information of workers, so it usually doesn't invade the personal privacy while producing a lower sensing cost. However, since workers only execute tasks they just pass by, the allocation result mainly depends on the workers' daily routes, and it is difficult to ensure that tasks can be surely executed. Conversely, *time-sensitive* task need to be executed quickly, so nearby workers are required to move specifically for task execution [15] [16] [17]. MCS platform needs to recruit nearby workers who are willing to move quickly for the task, this cause a higher incentive cost while ensuring tasks can be timely executed. Summarize studies above, we can find that existing studies [18] [19] [20] focus on either delay-tolerant or time-sensitive task allocation follows several same phases, including task detection, task publishing, receiving and execution, as shown in Fig.1. Wang et al. [21] mentioned that through task detection and publishing phase, task publishers detect some problems in urban areas and report them to the MCS platform, which is called **urban context sensing**. Through task receiving and execution phase, MCS participants receive tasks and execute them according to the allocation results of the platform, which is called **urban context actuation**. Between urban context sensing and actuation, we mainly allocate the sensing tasks that have been uploaded to the platform according to some methods. Obviously, the task allocation process of MCS platform will cause some time delays from sensing to actuation. However, there is a kind of task even more urgent than time-sensitive task in the city. When these tasks are detected, workers need to **I**nstant **S**ensing and then **I**nstant **A**ctuation (**ISIA**) seamlessly, which are called ISIA tasks. To show the key concepts and ideas of ISIA task, a realistic scenario is given as follows.

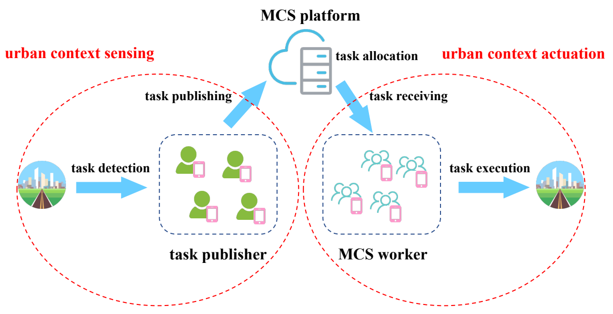


Fig. 1: Traditional task processing

Houchun Yin, Zhiwen Yu (corresponding author), Liang Wang, Lei Han and Bin Guo are with School of Computer Science, Northwestern Polytechnical University, Xi'an 710072, China. E-mail: shawn_yin@mail.nwpu.edu.cn, zhiwenyu@nwpu.edu.cn, liangwang@nwpu.edu.cn, leihan@nwpu.edu.cn, guob@nwpu.edu.cn.

Jiangtao Wang is with Tenure in the Intelligent Healthcare Centre, Coventry University, UK. E-mail: jiangtao.wang@coventry.ac.uk

There are numerous sensing locations in the city, requiring workers move to these locations for check (e.g. transmission lines, manhole covers, etc). These locations may have some urgent problems that need to be solved immediately (e.g. poles collapse, missing manhole covers, etc), which we call ISIA task. These ISIA tasks, if not be executed in time, often lead to severe consequences even the lose of lives. Therefore, detecting them and then immediately fixing them become a crucial mission. However, existing studies whether for delay-tolerant or time-sensitive mode allocate the tasks that have been published on the platform, and there is still a time delay from sensing to actuation. The ISIA task we proposed is more urgent. We hope to actuate as soon as it is detected. It means that there is no time delay from sensing to actuation. But the resulting time consumption of traditional task processing shown in Fig.1 is still not tolerable for ISIA tasks.

In order to deal with ISIA tasks, our main challenge is to recruit a group of workers to effectively cover the sensing area when ISIA tasks are undetected. We propose a new task processing mode, as shown in Fig.2. Different from previous studies, the mode we proposed moves the action phase of MCS platform to the stage before the beginning of sensing and actuation cycle, which makes the MCS platform pre-allocates workers based on the historical access data. During sensing and actuation cycle, workers directly interact with the sensors based on the allocation result, and then use their own sensing ability to execute some tasks to avoid the process of task uploading so as to enable instant sensing and then instant actuation. In this way, sensors directly transmit the real-time data to the workers without passing through the MCS platform. The short-distance transmission through Bluetooth, WiFi or other wireless communication methods will not produce higher network delay, and can ensure data transmission more stable and fast [22]. What's more, the running of allocation algorithms on MCS platform will cause some time delay. But under our mode, workers with corresponding abilities can actuate directly without the need for central controller to run algorithms during each sensing and actuation cycle, which saves the time and cost of allocation process.

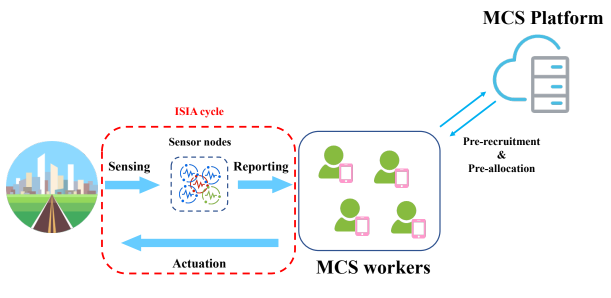


Fig. 2: ISIA task processing

Compared with the previous time sensitive problem, the ISIA problem we proposed mainly has the following challenges:

- The time tolerance of ISIA task is lower than traditional time-sensitive task. Previous studies considered sensing and actuation separately, which resulted in the lag of task execution. Our difference is that we want such tasks to be executed once they are detected, that is, to achieve the synchronization of sensing and actuation. Therefore, we need to consider sensing and actuation simultaneously, which is more complicated than before. In addition, we need to pre-recruit and pre-allocate workers in the case of undetected tasks. How to use historical data to mine the preferences of workers and the characteristics of sensing locations is also a key challenge, which directly affects the quality of sensing results.
- In order to enable instant sensing and then instant actuation, we not only need to maximize the coverage of the sensing area so as to find as many ISIA tasks as possible, but also need to maximize the matching degree between workers and the sensing types of tasks so as to make the ISIA tasks can be executed immediately to achieve a higher sensing quality. However, these two optimization objectives are conflicting and our another challenge is how to get an optimal balance.

Given the basic concepts in the analysis above, we propose a novel MCS task allocation framework, called ISIATasker, which jointly consider the sensing and actuation seamlessly. ISIATasker mainly consists of two phase. First, we propose a clustering method SPCP to cluster the densely distributed sensing locations into some POIs, each POI can be sensed by a worker. Because of the variety types of tasks, a greedy-based sensor selection method SSFP is proposed to select cooperative sensors carried by crowds, so as to make up for the lack of workers' sensing ability. Second, in order to enable instant sensing and the instant actuation, we need to maximize the POI coverage and the matching degree between workers and tasks at the same time. Therefore, a task allocation method PA-DDQN based on deep reinforcement learning is proposed, which defines a reward function to balances the two conflicting optimization objectives of POI coverage and matching degree. In summary, the main contributions of our work are as follows:

1) We are the first to propose the **ISIA** (Instant Sensing and then Instant Actuation) problem in mobile crowd sensing. Then we designed a novel MCS task allocation framework called ISIATasker, which jointly consider the sensing and actuation. While existing works focus on the allocation of tasks has been published to the platform, we are the first to consider the effective coverage of sensing area when ISIA tasks are undetected to enable an instant sensing and then instant actuation.
2) We designed a two-phase task allocation algorithm to implement ISIATasker, which consists of three key algorithms particular for solving ISIA problem. First, we propose a clustering method named SPCP based on Pearson correlation coefficient to cluster sensing locations into POIs and a greedy-based method SSFP for sensor selection. Second, in order to maximize the POI coverage and obtain a best match between workers and tasks to achieve instant sensing and then instant actuation, we proposed a method based on DDQN called PA-DDQN to allocate an optimal POI set for each worker.
3) We conduct extensive experiments based on real-world dataset: Chengdu city check-in dataset and camera

dataset. The results show that the performance of our method outperforms other baselines.

II. Related Works

Many MCS platforms (e.g., [3], [6], [7]) have been developed and widely used in recent years. For example, the WeSense application in CrowdOS [3] platform is a well-known MCS platform, which has attracted a lot of users to publish or execute tasks on it. As a core part of many MCS platforms, task allocation has attracted much attention and been widely studied. According to the urgency of MCS tasks, the existing studies can be divided into two categories [11]: *delay-tolerant* or *time-sensitive* task allocation.

Some studies focus on *delay-tolerant* tasks [13] [14] [23]. Since there is no requirement for such tasks to be executed quickly, we only need to assign workers who can finally execute these tasks during their daily routes. For example, Zhang et al. [12] proposed a greedy-based piggyback crowdsensing framework called CrowdRecruiter, which aims to recruit the minimum number of workers to minimize total incentive cost while still meeting coverage constraint. Xiong et al. [13] propose a novel framework, CrowdTasker, for piggyback mobile crowdsensing to achieve near-maximal coverage quality without exceeding incentive budget. Wang et al. [14] propose a user recruitment strategy based on semi-Markov model to calculate the utility of workers for tasks, so as to recruit the optimal K users to complete tasks. Song et al. [23] convert QoI-Aware multitask-Oriented participant selection problem to a nonlinear knapsack problem and then propose a dynamic participant selection (DPS) strategy to solve it.

Other studies focus on time-sensitive tasks [15] [16] [24]. These tasks are quite urgent and need to be executed quickly, which requires nearby workers to move to the sensing area in the shortest time for task execution. For example, Yucel et al. [24] propose a stable matching theory for MCS task allocation, so as to get a stable matching between two groups of entities according to workers' preference. This paper defines two stable conditions for user satisfaction, and verifies their matching stability for time-sensitive tasks. Yang et al. [15] proposed a distributed worker selection framework. They use a POI-based mobility prediction model to predict the probability of a task completed by a worker and then propose a greedy-based offline algorithm, which selects the most appropriate group of workers to maximize the probability of tasks to be completed. Li et al. [16] focus on the dynamic participant selection problem for large-scale heterogeneous sensing tasks. The optimization objective is to minimize the sensing cost under coverage constraints. They proposed an online method with caching mechanism for task whose sensing time is earlier than arrival time. [20] first define Min-Max Task (MMT) planning problem in MCS systems, considering time-sensitive and heterogeneity of sensing tasks. To address MMT problem, they propose a Memetic based Bidirectional General Variable Neighborhood algorithm. Wang et al. [25] consider a scenario where a mobile crowdsourcing task is too complex but can be divided into a number of easier subtasks, which have interdependency between them. They investigate an important problem, namely task graph scheduling in mobile crowdsourcing (TGS-MC), aim to minimize the task completion time and overall idle time with the consideration of worker reliability. [26] propose an effective prediction-based participant recruitment framework and divide participants into two categories: PAYG and PAYM with different incentive method, and aim to minimize the total cost.

These studies all allocate tasks that have already been published to the platform. But when ISIA tasks are detected, we need to enable instant sensing and then instant actuation, and the mode of publishing and receiving tasks through MCS platform is not applicable. In this paper, we propose a novel task allocation framework called ISIATasker. The work differs from studies above in the following three aspects: 1) Difference in basic problem. We first propose the instant sensing and then instant actuation (**ISIA**) problem in MCS to jointly consider sensing and actuation, and design a two-phase framework to solve this problem. 2)Difference in definition of optimization problem. Different from previous studies, we focus on the problem of POIs allocation when ISIA tasks are undetected. In order to achieve ISIA, we need to maximize both POI coverage and worker-task sensing type matching degree. And we define a reward function to transform the original problem into maximizing the total reward. 3) Difference in core algorithm. Some recent researches [27] [28] use deep learning methods to solve the related problems in Mobile Crowd Sensing, and can get a better result than traditional methods. Inspired by these papers, we propose a DDQN-based method called PA-DDQN, which allocates an optimal POI set for each worker to maximize the quality of instant sensing and then instant actuation to address ISIA problem.

III. Problem Formulation

In this section, we formulate the ISIA problem. Some main notations are illustrated in TABLE I.

TABLE I: Main Notation Used Throughout the Paper

Symbol	**Meaning**
L	Sensing location set
S	Sensor set
W	Worker set
LC	POI set
β	Number of sensing data types
$\Gamma(l_i)$ $\Gamma(s_i)$	Sensing data type of l_i and s_i
$Sc(s_i)$	Sensor s_i's sensing coverage
$\Delta(l_i)$	Number of sensors required for location l_i
Cov_w	Wireless coverage of mobile device
$\mathbf{X}_{\boldsymbol{w_i}}$	Worker w_i's sensing ability vector
$\mathbf{X}_{\boldsymbol{lc_j}}$	POI lc_j's sensing requirement vector
$x(lc_j)$	Number of sensing location in POI lc_j
$d(w_i, lc_j)$	Distance between worker w_i and POI lc_j
$D(w_i, LC)$	Total distance of worker w_i
$LC(w_i)$	Worker w_i's sensing and actuation path
Pr	All location-sensor pairs
Pr_v	Valid location-sensor pairs
$t(lc_j)$	Time consumption at lc_j
$T(w_i)$	Total time consumption of worker w_i
t_a	Time of sensing at a sensing location
t_b	Time of actuation at a sensing location
$P(w_i, LC)$	Total matching degree of worker w_i and POI set LC
$\lvert LC_{\text{cov}}\rvert$	Number of POIs covered by workers

Different from previous studies such as [14] [24] [29], we focus on the the allocation of POIs when ISIA tasks are not detected. There are a large number of sensing locations $L = \{l_1, l_2, ...l_n\}$ in the urban area. These sensing locations have β types of sensing data (e.g., traffic conditions, air quality) and a probability of the problem (ISIA task) occurrences. The sensing type of l_i is defined as $\Gamma(l_i)$. In addition, there are some sensors $S = \{s_1, s_2, ...s_q\}$ carried by crowds in the city. The sensing type of s_i is defined as $\Gamma(s_i)$ while sensing coverage is defined as $Sc(s_i)$. When s_i is within the wireless coverage of mobile devices carried by worker, it can transmit sensing information of current location to worker for judging whether there is an ISIA task. The number of monitoring sensors required for each sensing location is $\Delta(l_i)$. The set of workers is denoted as $W = \{w_1, w_2, ...w_p\}$. The wireless coverage of the mobile device carried by each worker is Cov_w. Each worker w_i has a maximum traveling time constraint T_i and may have up to β types of sensing ability. The multi-ability constraint of a worker is expressed as vector $\mathbf{X}_{\boldsymbol{w_i}} = [\chi_{1w_i}, \chi_{2w_i}, \cdots \chi_{\beta w_i}]$. $\chi_{jw_i} = 0$ means the worker doesn't have the jth ability.

In our problem, we aim to assign a worker with relevant ability to execute ISIA task. So we cluster L into new POI set $LC = \{lc_1, lc_2, ...lc_m\}$ according to the types and spatial distribution of sensing locations. Then we assign each POI a worker whose ability best match the types of possible ISIA tasks, so that when tasks occur within the POI, nearby workers can actuate immediately. Each lc_i contain $x(lc_i)$ sensing locations $\{l_1, l_2, ...l_x\}$, and $\sum_{i=1}^{m} x(lc_i) = n$. By counting the number of sensing locations of different types, the requirement vector of each POI is expressed as $\mathbf{X}_{\boldsymbol{lc_i}} = [\chi_{1lc_i}, \chi_{2lc_i}, \cdots \chi_{\beta lc_i}]$. $\chi_{jlc_i} = 0$ means the POI doesn't have the jth requirements. For worker w_i, the distance to next POI lc_j is denoted by $d(w_i, lc_j)$. We use $LC(w_i) = \{lc_1, lc_2, ...lc_\gamma\}$ to denote the order of γ locations checked by each worker, so the total distance can be formulated as $D(w_i, LC) = \sum_{j=1}^{\gamma} d(w_i, lc_j)$. We define an time slice from the end of check at the previous location to the end of check at the next location lc_j as a cycle. This process consuming time period $t(lc_j)$. The total time consumption $T(w_i)$ of w_i to execute $LC(w_i)$ can be formulated as $T(w_i) = \sum_{j=1}^{\gamma} t(lc_j)$. Obviously, $t(lc_j)$ is composed of moving time(MT) period, sensing time(ST) period and actuation time(AT) period. Moving time is determined by the distance $d(w_i, lc_j)$, sensing time is determined by the number of sensing locations in lc_j, and actuation time is determined by the number of urgent tasks in lc_j. If no urgent tasks are found, then no actuation time is required. For simplicity, we suppose that moving speed of each worker is v, checking a sensing location consumes a constant time period t_a, and executing an urgent task consumes a constant time period t_b. Furthermore, we need to select an appropriate sensor set for each POI to assist worker for sensing. We select the optimal sensor set in the area of all sensing locations $\{l_1, l_2, ...l_x\}$ in lc_i and denote all location-sensor pairs as $Pr = lc_i \times S = \{(l_j, s_k) | l_j \in lc_i, s_k \in S\}$. Based on the type of sensing locations and sensors, the valid location-sensor pairs are selected from Pr and denoted as $Pr_v = lc_i \times S = \{(l_j, s_k) | l_j \in lc_i, s_k \in S, and\ \Gamma(s_k) = \Gamma(l_j)\}$.

One of our optimization objective is to allocate a suitable set of POIs to each worker, so as to maximize the worker-task matching degree to enable ISIA. Specifically, for each lc_j the probability that the possible task requirements coincide with a worker's ability is denoted as $p(w_i, lc_j)$. Therefore, for γ POIs, the total matching degree of worker w_i is $P(w_i, LC) = \frac{p(w_i,lc_1)+p(w_i,lc_2)+...+p(w_i,lc_\gamma)}{\gamma}$. Besides, in order to find more tasks as soon as possible, our another optimization objective is to maximize the POI coverage. In other words, we focus on how to maximize the total number of POIs denoted as $LC_{cov}(LC_{cov} \subseteq LC)$ covered by workers, while keeping the workers' moving distance under certain constraint. The optimization problem can be formulated as follows:

$$\boldsymbol{give}\ w_i \in W, lc_j \in LC, l_q \in L, s_k \in S \tag{1a}$$

$$\boldsymbol{Maximize} \sum_{i=1}^{n} P(w_i, LC) \tag{1b}$$

$$\boldsymbol{Maximize}\ |LC_{cov}| \tag{1c}$$

$$\boldsymbol{s.t.} \sum_{l_q \in L} |(l_q, s_k)| \leq 1, \forall s_k \in S \tag{2a}$$

$$\sum_{w_i \in W} |(w_i, lc_j)| \leq 1, \forall lc_j \in LC \tag{2b}$$

$$t(lc_j) = \frac{d(w_i, lc_j)}{v} + x(lc_j) \times t_a + \sigma(lc_j) \times t_b \tag{2c}$$

$$T(w_i) = \sum_{j=1}^{\gamma} t(lc_j) = \frac{D(w_i, LC)}{v} + x(LC) \times t_a + \sigma(LC) \times t_b \tag{2d}$$

$$T(w_i) \leq T_i, \forall w_i \in W \tag{2e}$$

Here, the objective function in 1b is the total matching degree between workers and POIs. The objective function in 1c is the coverage of POIs. Constraint 2a denotes that each sensor s_k can sensing no more than one sensing location. Constraint 2b denotes that each POI should be effectively covered by no more than one worker. Constraint 2c denotes that a unit check time period is composed of moving time, sensing time and actuation time. Constraint 2d denotes the total time consumption of w_i to execute $LC(w_i)$. Constraint 2e denotes that the total time consumption of worker w_i to actuate should not exceed T_i.

IV. Task Allocation Strategy

A. ISIATasker overview design

As shown in Fig.3. The design of ISIATasker mainly includes two phase. In the *sensing locations clustering and sensor selection phase*, we first cluster a large number of sensing locations and then take each clustering result as a POI, which has a possibility of ISIA tasks. The purpose of clustering is to make the probability distribution of tasks detected at each POI more consistent with the ability distribution of workers. Therefore, we have introduced the Pearson correlation coefficient into clustering process to measure the similarity of tasks and workers. Furthermore, for each POI, we need to select an optimal cooperative sensor set to assist workers for effective sensing. Workers who receive sensing information from these sensors can achieve ISIA. We defined the utility and redundancy of each sensor set, and adopt a descent greedy-based method for sensor selection. By circularly deleting sensors with highest redundancy, we finally retain sensors that have a long-term stable matching with the sensing locations as the result set, so that we can use the historical information of these sensors to guide the next phase of POI allocation. Finally, in the *POIs allocation phase*, we need to maximize total POI coverage and matching degree simultaneously. Based on the historical access data, we propose a deep reinforcement learning method called PA-DDQN to maximize the POI coverage and matching degree for instant sensing and then instant actuation.

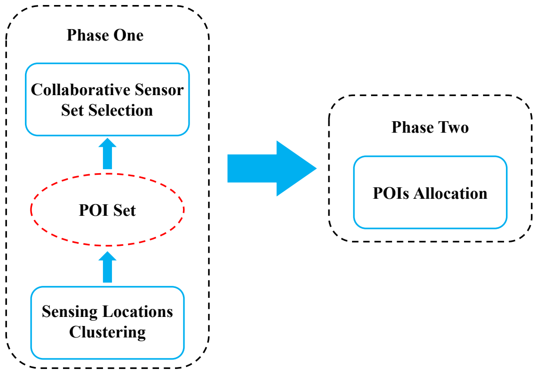

Phase One
Collaborative Sensor Set Selection
POI Set
Sensing Locations Clustering
Phase Two
POIs Allocation

Fig. 3: ISIATasker framework

B. Sensing location clustering and sensor selection

For sensing locations in urban area, some studies adopt the method of equal-grid division[30] to divide them into small areas, and then allocate the tasks within each grid. This method surely can obtain the locally optimal result, but it also has some disadvantages. The distribution of possible ISIA tasks in the sensing area is not uniform, for example, the task distribution near POI is usually denser. The equal-grid division method will lead to too few tasks in some areas and too many tasks in others, thus reducing the quality of task allocation. By contrast, we adopt an improved method based on K-means to cluster several sensing locations into parent POIs according to their types and spacial distribution.

As shown in Fig. 4, when the distribution of sensing locations has obvious spatial aggregation characteristics, we can use the simple K-means method to cluster these sensing locations and there is no overlap between these clustering results. However, in another case, the distribution of sensing locations is relatively dense, while worker has limited sensing ability. We need to divide these area into multiple POIs. In Fig.5(a), simple K-means divides these sensing locations into two categories according to the spatial distribution. However, this method does not consider the heterogeneity of sensing locations, which leads to the mismatch between workers' ability and the composition of POIs. We found that in this case, there are some overlapping areas, and the sensing locations in these overlapping areas have a variety of clustering choices. Therefore, as shown in Fig. 5(b), we optimize the clustering process by measuring the correlation between the type of sensing locations and the ability of workers, so as to maximize the matching degree between the composition of clustering results and workers' ability.

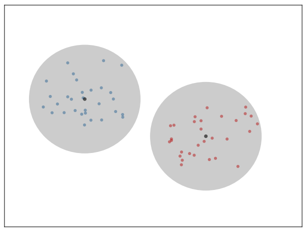


Fig. 4: Two POIs without overlap

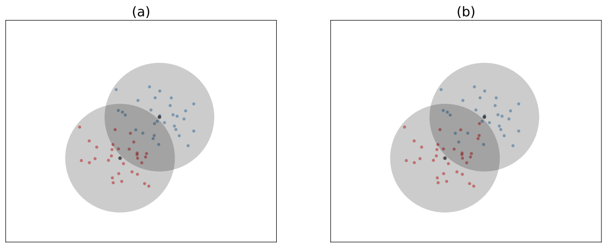


Fig. 5: Two overlapping POIs

The sensing ability vector of each worker w_i can be expressed as:

$$\mathbf{X}_{\boldsymbol{w_i}} = [\chi_{1w_i}, \chi_{2w_i}, \ldots \chi_{\beta w_i}] \tag{3}$$

Furthermore, we can estimate the similarity between the composition of a POI lc_j and the ability of worker w_i by the Pearson correlation coefficient:

$$r_{w_i}^{lc_j} = \frac{\sum_{k=1}^{\beta}(\chi_{kw_i} - \overline{\chi_{w_i}})(\chi_{klc_j} - \overline{\chi_{lc_j}})}{\left[\sum_{k=1}^{\beta}(\chi_{kw_i} - \overline{\chi_{w_i}})^2 \sum_{k=1}^{\beta}(\chi_{klc_j} - \overline{\chi_{lc_j}})^2\right]^{\frac{1}{2}}} \tag{4}$$

Where $\overline{\chi_{w_i}} = \frac{1}{\beta}\sum_{k=1}^{\beta}\chi_{kw_i}$, $\overline{\chi_{lc_j}} = \frac{1}{\beta}\sum_{k=1}^{\beta}\chi_{klc_j}$. The absolute value of $r_{w_i}^{lc_j}$ is proportional to the similarity between POI and worker. Higher value of $r_{w_i}^{lc_j}$ represent the higher matching degree of POI lc_j and w_i. Then we introduce the Pearson correlation coefficient into the clustering process and cluster the sensing locations by balancing the spatial distribution and similarity. The total similarity between a clustered POI lc_j and worker set W is defined as Equation 5a, and the matching gain is defined as Equation 5b:

Algorithm 1 Sensing location clustering based on Pearson correlation coefficient (**SPCP**)

```
Input:
    sensing locations set $L = \{l_1, l_2, ...l_n\}$;
Output:
    POIs set $LC = \{lc_1, lc_2, ...lc_m\}$;
1: Initialize, $I = 0$; Randomly select $k$ locations from $L$ as the initial clustering center $L^{(0)} = (l_1^{(0)}, ..., l_k^{(0)})$;
2: Cluster the set $L$. For class centers $L^I = (l_1^{(I)}, ..., l_k^{(I)})$, $l_j^I$ is the center of class $lc_j$;
3: for each $l_i \in L$ do
4:     $lc_{res} = lc_{closest}$, $maxIncre = 0$;
5:     for each $l_j^{(I)} \in L^{(I)}$ do
6:         Calculate the distance from $l_i$ to the center $l_j^{(I)}$ of each class $lc_j$;
7:         Calculate the current correlation coefficient $r_W^{lc_j}$;
8:         if $distance(l_i, l_j^{(I)}) \leq Cov_w$ then
9:             $Incre = \left| r_W^{lc_j \cup l_i} \right| - \left| r_W^{lc_j} \right|$
10:        end if
11:        if $Incre \geq maxIncre$ then
12:            $maxIncre = Incre$
13:            $lc_{res} = lc_j$
14:        end if
15:    end for
16:    Classify $l_i$ into $lc_{res}$
17: end for
18: Calculate the mean in the current classes $LC$;
19: Update class center $L^{(I+1)} = (l_1^{(I+1)}, ..., l_k^{(I+1)})$;
20: if $Iterative\ unconvergence$ then
21:     $I = I + 1$, go back to 2;
22: end if
23: return $LC$
```

$$r_W^{lc_j} = \sum_{w_i \in W} r_{w_i}^{lc_j} \tag{5a}$$

$$Incre = \left| r_W^{lc_j \cup l_i} \right| - \left| r_W^{lc_j} \right| \tag{5b}$$

Finally, our goal is to maximize the similarity between each $w_i \in W$ and $lc_j \in LC$, so that enable the composition of POIs best match the ability of workers. The pseudocode of the above sensing location clustering process based on K-means is presented in Algorithm 1.

After clustering process, we need to select an optimal cooperative sensor set for each POI to assist worker for effective sensing. The goal of sensor selection is to ensure that locations can be effectively covered by minimum set of sensors while maximizing the sensing quality. Each sensing location has a coverage threshold limit of $\Delta(l_i)$. In order to measure the sensing quality of the selected sensor set, the utility of each location-sensor pair is defined as follows:

$$\begin{aligned} &U(l_j, s_k) = \omega(l_j, s_k)/d(l_j, s_k), && if\ d(l_j, s_k) \leq Sc(s_k) \\ &U(l_j, s_k) = 0, && if\ d(l_j, s_k) \geq Sc(s_k) \end{aligned} \tag{6}$$

Algorithm 2 Sensor set selection for each POI (**SSFP**)

```
Input:
    A POI $lc_i = \{l_1, l_2, ..., l_x\}$; Sensor set $S = \{s_1, s_2, ...s_m\}$;
Output:
    The optimal set of sensor $S_{res} = \{s_1, s_2, ...s_k\}$;
1: Initialize the location-sensor pool $Pr_v = \{(l_j, s_k) | l_j \in LC_i, s_k \in S, and\ \Gamma(s_k) = \Gamma(l_j)\}$;
2: for each $l_j$ in $lc_i$ do
3:     Calculate the number of pairs $PrNum(l_j)$.
4: end for
5: while each $PrNum(l_i^j) > K$ do
6:     Set $maxRe = 0$
7:     for each $s_l \in S/S_{res}$ do
8:         if $Re(lc_i) - Re(S/\{s_l\}) > maxRe$ then
9:             $maxRe = Re(lc_i) - Re(S/\{s_l\})$
10:            $OptimalS \leftarrow s_l$
11:        end if
12:    end for
13:    $Pr_v = Pr_v - Pr(l_i^j, OptimalS)$
14:    $PrNum(l_j) = PrNum(l_j) - 1$
15: end while
16: for each $s_l \in Pr\ in\ Pr_v$ do
17:     $S_{res} = S_{res} \cup \{s_l\}$
18: end for
19: return $S_{res}$
```

ω is weight factor, represents the number of occurrences of the matching pair in the past. Thus, our objective at this phase is to allocate an optimal set of sensors for each POI to maximize the sensing quality of the task:

$$U(lc_i, S) = \sum_{j=1}^{x} \sum_{s_k \in S_K} U(l_j, s_k) \tag{7}$$

S_K are K sensors to sensing l_j with the highest utility ($K \geq \Delta(l_j)$). Based on the analysis above, we introduce an intermediate variable called redundancy to determine whether a location-sensor pair should be removed from the result set. For a sensing location, the redundancy of its corresponding sensor set is formulated as:

$$Re(l_j) = \frac{\sum_{s_k \in S_K} U(l_j, s_k)}{U(l_j, S)} \tag{8}$$

Thus, we formulated the total redundancy of POI lc_i as:

$$Re(lc_i) = \sum_{j=1}^{n} Re(l_j) \tag{9}$$

If removing a location-sensor pair reduces the most total redundancy while the sensor group still meets the coverage requirements, we should remove this pair from the pool. Finally, a descent greedy method is proposed for sensor selection, as shown in Algorithm 2. This method initializes a location-sensor pool that contains all valid pairs Pr_v. Based on which we remove a set of pairs to minimize the overall utility reduction.

C. POIs allocation

In this section, we need to allocate an optimal POI set for each worker. When a worker passes through a POI, some ISIA tasks may be detected. If the worker's ability don't meet the requirements of the task, the actuation will fail. Therefore, our goal is to maximize the total matching degree by making workers' ability best match the requirements of POIs. In addition, we also need to maximize POI coverage to ensure that as many ISIA tasks as possible are detected.

Each sensing location has a small probability of problem. Therefore, the ISIA task ut_i may occurs in one or more sensing locations $\{l_1, ..., l_e\}$ simultaneously. There are β single type of problem $\{\Gamma_1, \Gamma_2, ..., \Gamma_\beta\}$. Based on the history information of sensors deployed in the sensing area, the probability of a single type Γ_i task can be formulated as

$$P(\Gamma_i) = \frac{\sum_{j=1}^{q}\sum_{k=1}^{e}\{Count(s_j, l_k, O)|\Gamma(l_k) = \Gamma_i\}}{\sum_{j=1}^{q}\sum_{k=1}^{e}\{Count(s_j, l_k)|\Gamma(l_k) = \Gamma_i\}} \tag{10}$$

Where $Count(s_j, l_k)$ denotes the total times of sensors sense locations $\{l_1, ..., l_e\}$, $Count(ut_i, s_j, l_k)$ denotes the times that all sensors detects problem occurred in $\{l_1, ..., l_e\}$ simultaneously. For ε sensing type in a POI, there may be $E = 2^\varepsilon$ type of urgent task ut_i. In the clustering process, we control the value of $\varepsilon \leq 10$. When there is at least one sensing location have problem, we assume that the POI has a task that requires the worker to actuation instantly. We use vector $\mathbf{X}_{\boldsymbol{ut_i}} = [\chi_{1ut_i}, \chi_{2ut_i}, ...\chi_{\beta ut_i}]$ to denote the type of ut_i, and the probability of ut_i can be formulated as:

$$P(ut_i) = \prod_{j=1}^{\beta}[P(\Gamma_j) \times \chi_{jut_i} + (1 - P(\Gamma_j)) \times (1 - \chi_{jut_i})] \tag{11}$$

Finally, the probability that the possible sensing requirements coincide with a worker's ability $p(w_i, lc_j)$ can be calculated from Equation 12.

$$p(w_i, lc_j) = \sum_{k=1}^{E}\{\frac{Num(\chi_{aut_k} \leq \chi_{aw_i})}{\beta}p(ut_k)|\chi_{aut_k} \in \mathrm{X}_{ut_k}\} \tag{12}$$

POI coverage is another major factor to consider in this problem. We need to maximize the coverage of POIs under limited time constraint. The time consumption $T(w_i)$ of workers is mainly composed of three parts: moving time, sensing time and actuation time, as shown in Equation 13.

$$\begin{aligned} T(w_i) &= \sum_{j=1}^{\gamma} t(lc_j) \\ &= \frac{D(w_i, LC)}{v} + x(LC) \times t_a + \sigma(LC) \times t_b \end{aligned} \tag{13}$$

The moving time and sensing time are fixed values, which can be calculated according to the distance of the POIs and the number of sensing locations. The execution time is related to the number of tasks detected in the POI, but the number of tasks detected at different times is usually uncertain. Therefore, we use the mean time to represent the actuation time. For an POI, the mean vector of various tasks that may be detected can be formulated as Equation 14, which reflects the density of ISIA tasks. So the mean actuation time $E(AT(lc_j))$ for lc_j is formulated in Equation 15.

$$\begin{aligned} \mathbf{X}_{\boldsymbol{ut}} &= \sum_{j=1}^{E} P(ut_j|lc_i)\mathbf{X}_{\boldsymbol{ut_j}} \\ &= \sum_{j=1}^{E}[P(ut_j|lc_i)\chi_{1ut_j}, ..., P(ut_j|lc_i)\chi_{\beta ut_j}] \\ &= [\chi_{1ut}, \chi_{2ut}, ..., \chi_{\beta ut}] \end{aligned} \tag{14}$$

$$\begin{aligned} \sigma(lc_j) &= \sum_{i=1}^{\beta}\chi_{iut}, \\ E(AT(lc_j)) &= \sigma(lc_j) \times t_b, \forall lc_j \in LC \end{aligned} \tag{15}$$

The main challenge in our problem relates to the two contradictory optimization objectives. This is because the distance between each POI with a high matching degree may be relatively far, which leads to workers' more moving time consumption. However, under the constraint of maximum time consumption, workers can only check a limited number of POIs, which cannot guarantee the maximum coverage. On the contrary, the allocation solution that achieve the maximum coverage of POIs usually cannot have a relatively higher degree of matching between worker and possible tasks. Therefore, it is impossible for an optimal solution to satisfy these two optimization objectives simultaneously.

To get relatively optimal solution of the ISIA problem, we designed a reward function $\mathbb{R}$ based on the relationship between the two optimization objectives to solve ISIA problem, as shown in Equation 16.

$$\mathbb{R} = \sum_{w_i \in W}(\eta_1 \cdot \sum_{lc_j \in LC(w_i)} p(w_i, lc_j) - \eta_2 \cdot T(w_i)) \tag{16}$$

$\mathbb{R}$ is mainly composed of expected profit and cost, which represents the expected rewards that workers can get from the POIs check process. When there is a higher matching degree between workers and tasks, we can obtain higher expected profits, this is because these workers have a higher probability of timely actuation. η_1 represents the unit profit of the unit matching. Moreover, the cost includes moving time cost, sensing time cost and actuation time cost. η_2 represents the cost per unit time. In the following method, η_1 and η_2's value will be updated automatically with the training process until an optimal solution is obtained. Next, we prove that the POIs allocation problem we proposed is NP-hard.

Theorem 1. *The POIs allocation problem is NP-hard.*

Proof. *In orienteering problem [31], we need to plan a path for worker to visit a number of locations. The objective of orienteering problem is to maximize the points coverage under a maximum moving distance limit. This orienteering problem* *is already proved to be NP-hard. Next, we assume that there is only one worker in our instance as the worker in the orienteering problem. Then, the worker has a predetermined maximum time consumption. The sensing time and the actuation time is set to 0. Obviously, the worker's time consumption is directly proportional to the moving distance, which is mapped to the maximum moving distance in the orienteering problem. We set the unit cost of workers to 0, thus the reward $\mathbb{R}$ equals the total expected profit according to Equation 16, which is also mapped to the points in the orienteering problem. Obviously, the instance of our problem is exactly an orienteering problem, which is also NP-hard.*

Next, we aim to maximize total reward $\mathbb{R}$. In this paper, we try to address this problem by using reinforcement learning (RL) approach. The goal of RL is to find an optimal strategy that allows an agent to receive most rewards from the environment. Different from some meta-heuristic methods, the training of RL method is based on a large number of historical experience data. In fact, it takes into account the past, current and future situation of agent and can relatively get a closer result to the global optimal solution. Q-learning is a value-based algorithm in reinforcement learning, which has been widely used. It predicts the action should be taken at any state by establishing a Q-table. The algorithm updates the Q-table according to Q-function, which is shown as follows:

$$Q(s,a) = Q(s,a) + \alpha(r + \gamma \max_{a' \in \mathcal{A}} Q(s',a') - Q(s,a)) \quad (17)$$

Where $Q(s,a)$ represents the expected reward of taking action a at state s, α is the learning rate, γ is the reward decay coefficient and r is the reward received after taking action a from state s to s'. However, in our problem, the state space and action space are too large to include all the actions and states with a Q-table, so Q-learning algorithm is not applicable. We propose a method based on double deep Q-network (DDQN) to solve our problem. The DDQN uses two DNNs, which includes a policy network and a target network to help search the optimal strategy. We use $\mathcal{S}$ and $\mathcal{A}$ to denote the state space and action space. A state $\mathcal{S}$ contains the current allocation result of POIs and workers in the urban area. An action $\mathcal{A}$ means we assigns a POI to a worker. Every time performing an action, we can get a reward and reach the next state. We adopt the state representation in [28] and define state $s \in \mathcal{S}$ as a $2m$ dimensional vector, where each of the first m elements indicates the worker assigned to each POI. The remaining m elements represent the order number of workers passing by POIs. An action $a \in \mathcal{A}$ indicates we assign a POI $lc_j \in LC$ to worker $w_i \in W$. It can be divided into valid action and invalid action, in which valid action means that there will be no sensing conflict between workers. In addition, we also add a termination status $done$ to represent whether the current state is a termination state. When the system has completed the allocation of all workers, we set $done$ to 1. In addition, when the current state is invalid, we also set $done$ to 1 and terminate current episode of training.

We show the pseudocode of POIs allocation algorithm in Algorithm 3. Line 1 initializes a replay memory D with a certain capacity of N to store samples. Line 2-3 initialize the policy network and target network respectively. In the training process, we randomly select a batch of state transition samples from the replay memory to train the policy network. Line 4 initializes a variable $\mathbb{R}$ to represent the reward value of the current global optimal solution. From line 5 we started a total of M episodes of training process. Lines 6-9 update the current state of each worker and the set of valid actions that can be selected in the next step. Then we set the elements

Algorithm 3 DDQN based POIs allocation (**PA-DDQN**)

```
Input:
    POIs set $LC = \{lc_1, lc_2, ..., lc_m\}$; Worker set $W = \{w_1, w_2, ..., w_p\}$; Number of episodes $M$, capacity of replay memory $N$, probability of random selection $\varepsilon$, learning rate $\alpha$, discount factor $\gamma$, period to update target network $C$.
Output:
    Result path $LC(w_i) : \forall w_i \in W$; maximum reward $R$.
1:  Initialize replay memory $\mathcal{D}$ to capacity $N$.
2:  Initialize action-value function $Q$ with random weights $\theta$.
3:  Initialize target action-value function $\hat{Q}$ with weights $\theta^- = \theta$.
4:  $\mathbb{R} = 0$
5:  for episode = 1, $M$ do
6:      Reset workers locations $loc_W$.
7:      Initialize valid action set $\mathcal{A}_{valid}$.
8:      Initialize all elements in initial state vector $\boldsymbol{s_1}$ to zero.
9:      $r = 0$
10:     for $t = 1, T$ do
11:         Generate a random number $rand$ within [0,1].
12:         if $rand \le \varepsilon$ then
13:             Randomly select an action $a_t$ from $\mathcal{A}_{valid}$.
14:         else
15:             Select the action $a_t = \arg\max_a Q(s_t, a; \theta)$.
16:         end if
17:         if $a \in \mathcal{A}_{valid}$ then
18:             Get the one-step cost $r_{cost}$ and profit $r_{profit}$.
19:             $r = r + (r_{profit} - r_{cost})$, $done = 0$
20:         else
21:             $done = 1$, break
22:         end if
23:         Take action $a_t$ and get next state $s_{t+1}$.
24:         Store current transition $(s_t, a_t, r, done, s_{t+1})$ in $D$.
25:         Sample random minibatch of transitions $(s_j, a_j, r_j, done, s_{j+1})$ from $D$.
26:         Update policy network $Q$ via learning step.
27:         Every $C$ steps reset target network $\hat{Q} = Q$.
28:         Update current status of points and workers.
29:         Update the valid action set $\mathcal{A}_{valid}$ in current state.
30:     end for
31:     if $r > \mathbb{R}$ then
32:         $\mathbb{R} = r$
33:         Record the allocation result $LC(w_i) : \forall w_i \in W$.
34:     end if
35: end for
36: return $LC(w_i) : \forall w_i \in W$; $\mathbb{R}$.
```

of all vectors and the reward value in the initial state to 0. We start the state transition and network training from Line 10. Lines 11-16 show that we adopt the ε-greedy strategy to select the next state. If the current state is not a termination state, we randomly select the next action from the set of valid actions with the probability of ε, or get the next action with the probability of 1-ε through the policy network. Lines 17-22 show that if the next action is a valid action, we need to calculate one step reward, and then set the termination status to 0. Otherwise, we will set the termination status to 1 and terminate the training ahead of time. Line 23 means we take the current action and transit to the next state. In line 24 we store the current samples into the replay memory, where each sample vector is composed of the state s_t, the action a_t, the one step reward r, the termination status $done$ and the next state s_{t+1}. In lines 25-27 we train the network and update the parameters. We randomly select a batch of samples from the replay memory for training, and then update the policy network. Besides, we update the target network every C steps, which can prevent the over fitting of the network. Lines 28-29 update the state of each worker and the current valid action set. Finally, in Lines 31-34, we update the maximum $\mathbb{R}$ value and its corresponding task allocation strategy. After M episodes of training, we take the optimal strategy as the final allocation result.

V. Experimental Evaluation

In this section, we evaluate the effectiveness of our proposed algorithms through extensive experiments on real-world check-in dataset and camera location dataset in Chengdu city. We first introduce the dataset and the experimental settings. Then, some experimental purposes and baseline are proposed. Finally, we compare the algorithm in this paper with several baselines and analyze experimental results.

A. Datasets and Experimental Settings

In order to conduct experiments with real-world sensing location distribution, we combined a real-world check-in dataset and a real-world camera location dataset in Chengdu city. As shown in Fig.6.

Check-in dataset. It contains 490,000 check-in records in Chengdu, Sichuan, China. Each check-in data includes address, longitude, latitude and check-in type (for example, school, bank, restaurants, etc). These check-in points well reflect the natural distribution of sensing locations in the city. We divide these data into independent areas using equal grids, and select 50 areas with densely distributed points for experiments.

Camera location dataset. It contains the location of surveillance cameras in Chengdu, Sichuan, China, including camera ID, description, longitude and latitude. We map these points to the corresponding grid as sensors in these areas. Combined with the check-in dataset, we can get the approximate true distribution of sensing locations and sensors in the same region.

Then we present some settings of our experiments. First of all, we divide the whole urban area of Chengdu into several $5\text{km} \times 5\text{km}$ squares, and each square is regarded as an independent sensing area. Several areas with dense POI distribution were selected for the experiment. Limited by workers' sensing ability, the number of sensing locations in each POI needs to be controlled within a certain range, so $x(lc_i)$ varies from 10 to 50, and the number of sensing locations in different POIs can be different. For each sensing location, we generate a random number from 1 to β as its sensing type. Similarly, for each sensor point, we also randomly define a sensing type. For m POIs, we recruit p workers for sensing and actuation. m varies from 500 to 1000, and p varies from 10 to 50. Within each sensing cycle (e.g. 08:00-10:00), each POI is considered to be completed once it is checked by one worker. Here we use the Manhattan Distance [32] to measure the travel distance between two POIs. Each worker has a certain maximum time limit T_i, ranging from 1 to 2 hour. Each worker's movement speed v is fixed (i.e. $1.1m/s$). Workers should complete sensing within no longer than T_i. The unit cost per unit time of all workers is set to 1, and the profit a worker can obtain at an POI is proportional to the matching degree between the two, varies from 0 to 100.

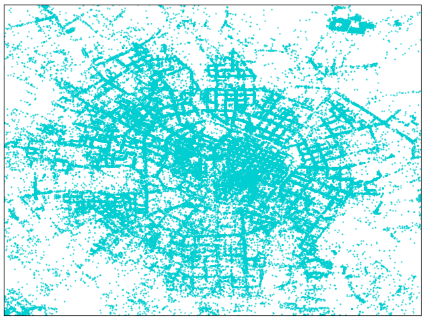


Fig. 6: The spatial distribution of sensing/sensor locations in Chengdu city

In addition, in the PA-DDQN algorithm, the capacity of replay memory is set to 10000, and the number of episodes is set to 7000. Probability of random selection ε is start with 0.1 and gradually increases to 0.9. The discount factor is set to 0.99, learning rate is set to 0.01, and the size of minibatch sampled from $\mathcal{D}$ is set to 128.

B. Experimental Purposes and Baselines

The goal of our experiments is to compare the performance of ISIATasker and other baseline methods under different situations, such as different number of tasks, different number of workers, different maximum sensing time limit, different number of clusters and so on. The performance comparison metrics contain total reward, POI coverage and matching degree. We provide the following baseline methods for comparative studies.

- *Random Allocation (RA)* : This method randomly selects an unallocated POI for a worker in each allocation cycle, while ensuring the result meets the current time limit. Since the order of workers may have an impact on the final allocation result, we randomly generate the worker orders and carry out random allocation several times (e.g.

50 times). We take the allocation strategy with the highest reward value as the final result of RA.

- *Weight-based Greedy Allocation (WGA)* : For the multi-objective optimization problem mentioned in this paper, we need to quantify the value of each allocation action. A simple idea is to use linear weighting method. The main idea is to ignore the different units and ranges between different objective functions. By setting corresponding weights for different objective functions, we can use a comprehensive utility function to represent the overall optimization objective. The linear weighting function is as follows:

$$r = \omega \times p(w_i, lc_j) + (1-\omega) \times \frac{t_{\max} - t(lc_j)}{t_{\max} - t_{\min}} \quad (18)$$

In this way, we can set different weights for different optimization objectives according to the actual situation, so as to get the optimal result. Obviously, when $\omega = 1$, the original multi-objective optimization problem is transformed into a single objective optimization problem based on matching degree. When $\omega = 0$, the original multi-objective optimization problem is transformed into a single objective optimization problem based on time cost. In our experiments, we set ω as 0, 0.5, and 1 respectively. Then we choose the worker-POI pair with the largest value r in each allocation cycle by greedy method to get the final result.

- *Reward-based Greedy Allocation (RGA)* : In this paper, we define a reward function $\mathbb{R}$ and transform the multi-objective optimization problem into a problem which aim to maximize total reward. Therefore, we propose a greedy algorithm to maximize the reward function $\mathbb{R}$ as the baseline. We greedily select the POI lc_j with the largest reward r for worker w_i in each allocation cycle. When the time consumption of all workers reaches the maximum limit, we take the allocation strategy as the final result.
- *DDQN based on Simple K-means (DDQN-SK)* : The clustering of sensing locations has a very important influence on the final optimization result. We add Pearson correlation coefficient in the clustering process to measure the similarity. So that the sensing type distribution of possible tasks within the POIs more consistent with the ability distribution of workers. In order to verify the effect of introducing Pearson correlation coefficient, we proposed DDQN algorithm based on simple K-means, which only performs simple spacial K-means clustering of sensing locations.

C. Experimental Results

In this subsection, we conduct the experiments and compare the performance of our method with other baselines. When varying one parameter, the other parameters are set to fixed values.

1) Different Number of Training Episodes: First of all, we compare the overall reward, matching degree, coverage ratio of different methods under different training episodes. We fix the number of POIs as 600, the number of workers as 60, the value of β as 15, and the number of sensing locations in each POI range from 10 to 20.

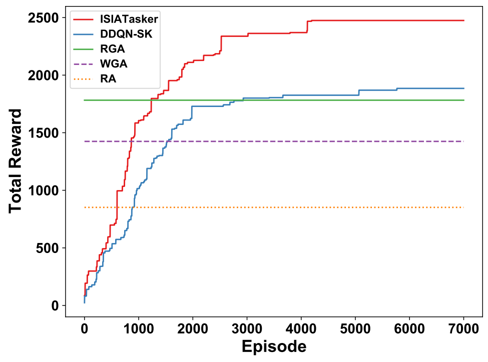


Fig. 7: Total Reward comparison under learning process

As shown in Fig. 7, with the increase of training episode, the total reward $\mathbb{R}$ of the DDQN-based methods (ISIATasker and DDQN-SK) is gradually increasing. The total reward of ISIATasker starts from a very low value. Through continuous learning, it outperforms the greedy method at about 1000th episode, and gets a better solution than any other baselines. After learning about 6000 episodes, ISIATasker finally converges to an approximate optimal value. DDQN-SK shows similar effect in the training process, but because the correlation between sensing types and workers' ability is not considered in the clustering process, this method can not achieve the same matching effect as ISIATasker, which results in a lower total reward. However, although the clustering process is less effective than other baselines, DDQN-SK still learns the approximate optimal solution in its solution space, and the total reward $\mathbb{R}$ exceeds RGA at around 3000th episode. In addition, other baselines (RGA, WGA, RA) are one-time algorithms. When the experimental data are fixed, each solution obtained by the algorithm is a fixed value and does not need the process of learning. Therefore, we illustrate it as a straight line in our figure. It can be seen from the Fig. 7 that the result of RGA is worse than ISIATasker, but similar to DDQN-SK. The result of WGA is worse than RGA, but better than RA.

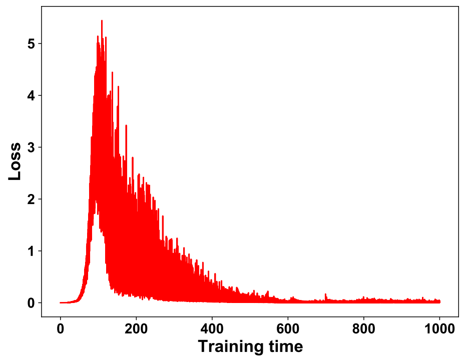


Fig. 8: Convergence of PA-DDQN loss function

The convergence curve of the loss function is shown in Fig. 8. At the beginning, the loss increases rapidly, this is because the experience replay memory has not collected enough training samples. When a sufficient number of samples are collected in the experience replay memory, we can see that the loss begin to decrease and finally converges to a low value. This shows that our network finally get a good training effect.

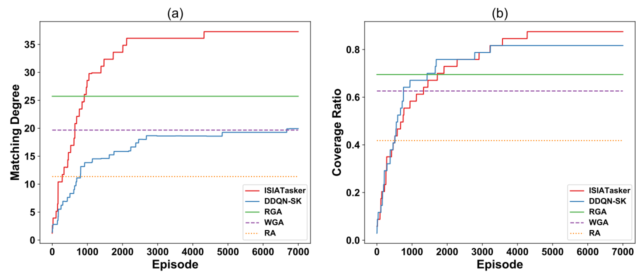


Fig. 9: Matching degree(left) and coverage ratio(right) comparison under learning process

Our optimization objectives are to maximize matching and the POIs coverage, as shown in Equation 1b and 1c. Therefore, we analyze the results from matching degree and coverage ratio. In Fig. 9(a), we can see that ISIATasker can converge quickly, and it can find higher matching degree compared with other baselines. But DDQN-SK algorithm only clusters sensing locations by simple K-means, the matching degree of the final result is worse than RGA and WGA based on greedy, only better than RA. In Fig. 9(b), we only consider the coverage ratio. It can be seen that ISIATasker and DDQN-SK get approximately the same effect, which is better than other baselines. This is because RL algorithm, which takes single time cost as optimization objective, can find better solution than greedy method without considering the influence of sensing type heterogeneity.

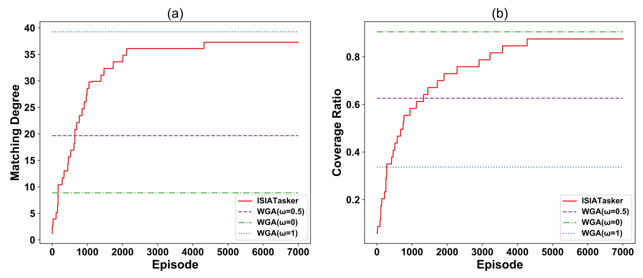


Fig. 10: Comparison between ISIATasker and WGA under learning process

ISIATasker is actually a kind of ”trade off” method, which aim to find a balance between maximizing matching degree and maximizing coverage ratio. Therefore, our method can’t get the optimal solution of each optimization objective. On the contrary, we only need to ensure that the results are approximate optimal solutions for both two objectives. As shown in Fig. 10, we set the weight ω in WGA to different values and compare it with ISIATasker. When the value of ω is 0, WGA only takes the coverage ratio as the single objective. When the value of ω is 1, WGA only takes the matching degree as the single objective. Thus, the original problem is transformed into two single optimization problems with different optimization objectives. In Fig. 10(a), WGA (ω = 0) only gets a very low matching degree. The matching degree of WGA (ω = 0.5) is higher than WGA (ω = 0) but lower than ISIATasker and WGA (ω = 1). ISIATasker has a higher matching degree, which is only slightly lower than WGA (ω = 1). From the perspective of coverage ratio, as shown in Fig. 10(b), the experimental results are just opposite to Fig. 10(a). WGA (ω = 1) only gets very low coverage ratio. The coverage ratio of WGA (ω = 0.5) is higher than WGA (ω = 1) but lower than ISIATasker and WGA (ω = 0). ISIATasker has a higher coverage ratio, which is only slightly lower than WGA (ω = 0). Generally speaking, both WGA (ω = 0) and WGA (ω = 1) optimize a single objective, which can not get a good balance between the two optimization objectives. ISIATasker achieves better results for both matching degree and coverage ratio and obtains a relatively optimal solution, which is better than other baselines.

2) Different Number of POIs: Next we evaluate the performance of ISIATasker and other baselines by setting different number of POIs. We will analyze the total reward, matching degree and coverage ratio, and explain the change trend of relevant experimental results. We fix the number of workers as 60, the value of β as 15, and the number of sensing locations in each POI range from 10 to 20.

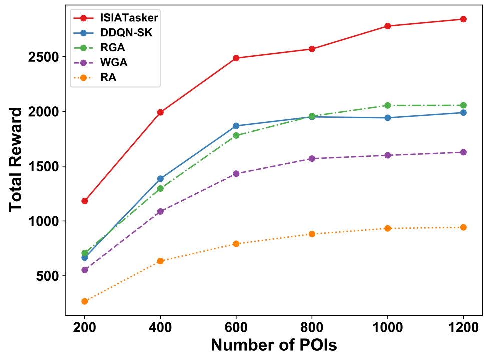


Fig. 11: Total reward comparison under different number of POIs

In Fig. 11, we compare the total reward of different methods under different number of POIs. When the number of POIs increases, there are more options with higher profits and lower costs for each worker, so the total reward of different methods are also increase. ISIATasker tends to select these POIs which make the total reward $\mathbb{R}$ higher and outperforms DDQN-SK, RGA, WGA and RA by achieving a higher total reward, respectively. However, it is worth noting that although ISIATasker outperforms other baselines in all settings, its significance becomes less obvious with the increase of POIs. This is because although the number of POIs is increasing, the ability of workers is limited, and they can only maximize the total reward within a certain time limit. When the POI reaches a larger scale, the probability of new point that can make workers more profitable is relatively low.

Fig. 12 analyzes the experimental results from matching degree and coverage ratio. In Fig. 12(a), we can see that with the increases of POIs number, the matching degree has a relatively slow growth trend. This is because the types of some new POIs better match the ability of workers than before.

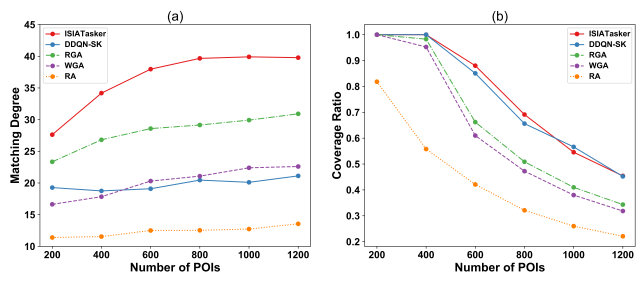


Fig. 12: Matching degree(left) and coverage ratio(right) comparison under different number of POIs

Our algorithm will assign these new POIs to workers first. In addition, with the increases of the number of POIs, the significance of the growth trend of our method becomes less obvious, and the reason is the same as that mentioned above. Next to ISIATasker in performance is RGA based on greedy. DDQN-SK is worse than the former two in matching degree. It has similar performance with WGA and is better than RA. As shown in Fig. 12(b), we analyze the experimental results from coverage ratio. It can be seen that with the increase of POIs number, the coverage ratio of all methods has a significant downward trend. This is because the number of workers is fixed, so the total ability is also limited. Workers can only cover as many POIs as possible under the limitation of their ability. When there are few POIs, such as 200 or 400, the ability of workers in the area can basically meet the total requirements. Therefore, ISIATasker and DDQN-SK can achieve a high coverage. It is also worth noting that the performance of DDQN-SK is worse than ISIATasker in matching degree, but it can achieve the same performance as ISIATasker in coverage ratio.

3) Different Number of Workers: In this part, we set different number of workers to get the experimental results under different conditions. Then the performance of ISIATasker and other baselines is evaluated from total reward, matching degree and coverage ratio. We fix the number of POIs as 600, the value of β as 15, and the number of sensing locations in each POI range from 10 to 20.

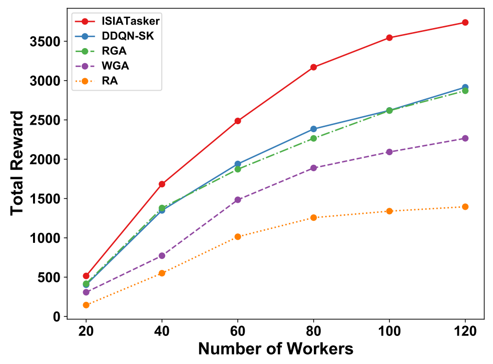


Fig. 13: Total reward comparison under different number of workers

As shown in Fig. 13, we compare the change of different methods' total reward under different number of workers. It can be seen from the figure that the total reward of all methods have a increasing trend. This is because the increase enables the platform to recruit more workers. In the case of a certain number of POIs, more workers can cover a wider sensing area. Therefore, compared with other parameters, the increase of the total reward brought by the increase of workers is more obvious. However, we can see that there is a slight slowdown in the growth of total reward. We find that each worker may give up some POIs with little profit but great cost, so the total reward will not show a linear growth. Compared with other baselines, ISIATasker achieves higher total reward. And this advantage becomes more obvious with the increase of the number of workers. Moreover, DDQN-SK and RGA have similar performance, which is better than WGA and RA.

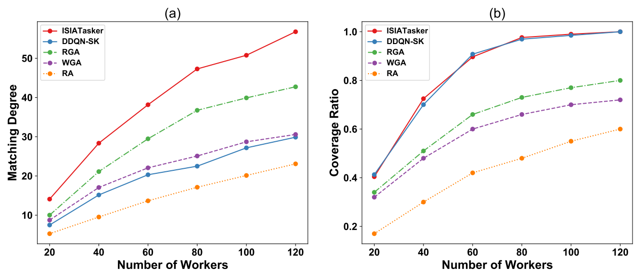


Fig. 14: Matching degree(left) and coverage ratio(right) comparison under different number of workers

Fig. 14 shows the experimental performance of matching degree and coverage ratio. As shown in Fig. 14(a), with the increase of the number of workers, the total matching degree has an obvious linear growth trend, which can be explained from the following two aspects. On the one hand, the platform can recruit more workers, so as to cover more POIs and improve the overall matching degree significantly. On the other hand, it means that workers have more composition of muti-ability, which can better match with heterogeneous POIs. In this case, ISIATasker is significantly better than other baselines in matching degree. Fig. 14(b) reports the change of coverage ratio. It can be seen from the figure that all methods has an obvious increasing trend. When the number of workers is enough, such as at 100 and 120, RL methods (ISIATasker and DDQN-SK) can achieve the approximate full coverage of POIs faster than other methods. Only from the perspective of coverage ratio, ISIATasker and DDQN-SK have the same performance, which is better than the baselines based on greedy (RGA and WGA) and random (RA). This is because ISIATasker and DDQN-SK have the same solution process for coverage ratio.

4) Different Values of β: Next, we analyze the influence of different values of β on the experimental results. We fixed the number of POIs to 600, the number of workers to 60, and the number of sensing locations in each POI range from 10 to 20.

As shown in Fig. 15, the total reward of all methods decreases with the increase of β. This shows that when the number of workers and POIs is fixed, more heterogeneous sensing type will lead to less total reward value of the allocation result. Although the downward trend of ISIATasker is obvious, its total reward is still higher than other baselines.

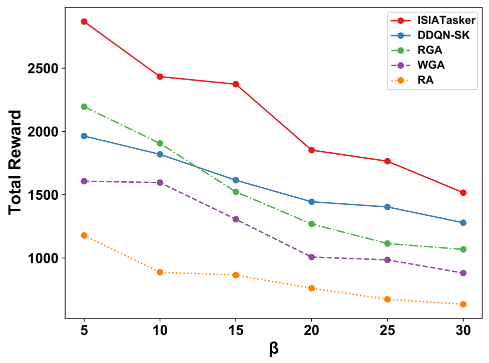


Fig. 15: Total reward comparison under different value of β

In order to find out the reasons behind this change, we analyzed the relationship between the two evaluation indexes of matching degree and coverage ratio with β. As shown in Fig. 16(a), similar to the previous methods, the matching degree of all methods also shows a downward trend. Obviously, the change of β has a direct impact on the matching degree. However, in Fig. 16(b), we can see that with the increase of β, the coverage ratio of all methods are stable floating in a fixed interval, and there is no obvious upward or downward trend. This shows that the change of β does not affect the coverage ratio of the final results. Obviously, the reason for the decrease of total reward and matching degree is that the increase of heterogeneous sensing types leads to greater differences between workers and POIs, which makes it more difficult for our methods to find the best matching between workers and POIs.

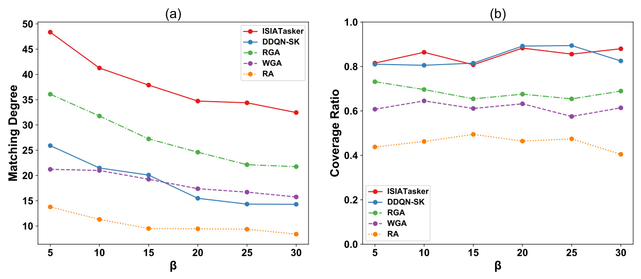


Fig. 16: Matching degree(left) and coverage ratio(right) comparison under different value of β

5) Different of Other Parameters: As shown in Fig. 17, we evaluate the similarity between the POIs assigned to different workers. Lighter color blocks indicate the lower similarity between two POIs, and darker color blocks indicate a higher similarity. $a0$, $a1$, $a2$, $a3$, $a4$ and $a5$ represent the six POIs assigned to worker a in the result, and $b0$, $b1$, $b2$, $b3$, $b4$, $b5$ represent the six POIs assigned to worker b. It is obvious from the figure that the similarity between POIs belonging to the same worker is higher. For example, the darker color blocks are concentrated in $[a0, a1, a2, a3, a4, a5]$ and $[b0, b1, b2, b3, b4, b5]$. However, the similarity between the POIs of worker a and worker b is relatively low. The experimental results show that our method tends to select the POIs which are more suitable for each worker, so as to make full use of the worker's ability to enable instant sensing and then instant actuation.

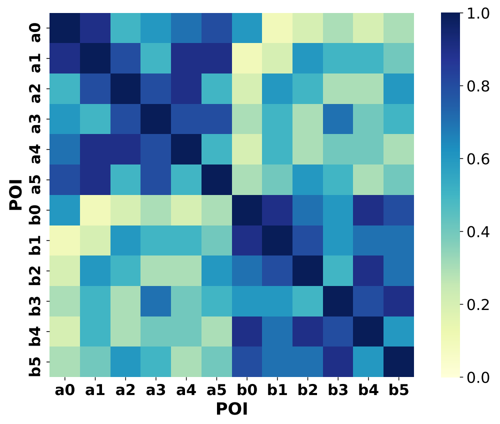


Fig. 17: POIs Similarity

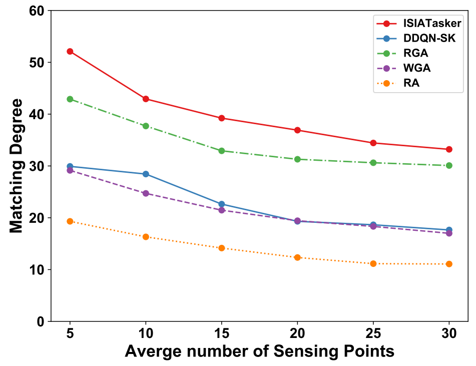


Fig. 18: Matching degree comparison under different average number of sensing locations

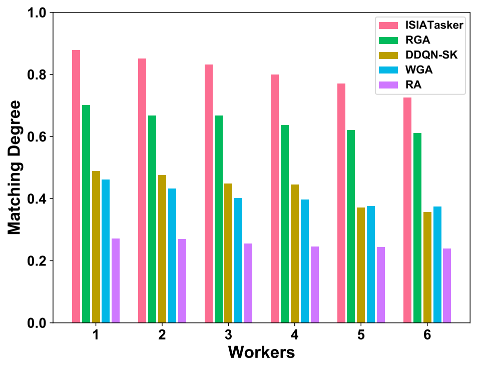


Fig. 19: Matching degree comparison for different workers

Fig. 18 evaluates the performance of different methods on matching degree under various average number of sensing locations in one POI. We can see that the matching degree of all methods decreases with the increase of the average number of sensing locations. This is because more sensing locations make the composition of POI more complex, which increases the difficulty of the best matching. In this case, the performance of ISIATasker is obviously better than other baselines. In addition, we randomly selected six workers in the experimental results and evaluated the difference of matching degree obtained by different methods, as shown in Fig. 19. Obviously, the matching degree of ISIATasker is significantly higher than that of other baselines.

D. Algorithm Complexity and Performance Analysis

In this section, we analyze the time complexity of our proposed algorithms. The SPCP method clusters a large number of discrete sensing locations into several POIs with similar sensing types. The running time complexity of SPCP for clustering will be $O(\beta \times n \times k)$, in which β is the dimension of sensing type vector, n is the number of sensing locations, and k is the number of clustering categories. The sensor selection method SSFP in each POI after clustering adopts an iterative process, which continuously removes the combination with highest redundancy from the pool to reduce the sensing redundancy. The running time complexity of SSFP is $O(|lc_i| \times (2\,|S| - K))$. In the worst case, the running time complexity is $O(2 \times |lc_i| \times |S|)$.

In order to deal with the high complexity of the allocation problem under ISIA constraints, we use PA-DDQN based on deep reinforcement learning to find efficient solutions. As shown in algorithm 3, PA-DDQN is an offline task allocation method, and the model training process will produce some time consumption. However, because the MCS platform pre-trains the model and pre-recruits workers based on historical data before the start of sensing and actuation cycle, which means PA-DDQN will not be executed during the sensing cycle, so the running time of this method will not make the time consumption of sensing and actuation longer. Therefore, this model can be used for ISIA tasks. PA-DDQN uses two deep neural networks (DNNS) to help search the optimal strategy, which is used to learn the best operation (allocate action) and the reward of all possible operations in each state. We can choose different numbers and dimensions of hidden layers according to the complexity of the ISIA problem. Different from many meta-heuristic methods whose solution quality depends on the fine tuning of parameters, this method takes into account the past, current and future possible situations, so we can get the result which is closer to the global optimal solution of NP-hard problem, The experimental results also show that our PA-DDQN method has a better performance.

VI. Limitation and Discussion

This section discusses other issues that are not addressed in this work due to various constraints, which we plan to investigate in our future work.

The unpredictability and fail execution of tasks. Our algorithms are executed before the sensing and actuation cycle, so it only guarantees the maximum coverage of POIs that may have some sensing tasks, while ensuring the maximum matching between workers and tasks based on historical data, which is completed by offline allocation. However, in the process of sensing and actuation, due to the uncontrollability of workers and the unpredictability of ISIA tasks, our MCS system is likely to encounter the situation of task execution failure. We need to study how to deal with these failed tasks, and extend the task allocation algorithm from offline phase to online phase to reallocate workers and tasks dynamically, so as to further improve the execution success rate and sensing quality of tasks.

Lack of historical access data. Under the scene of sparse mobile crowd sensing, the historical access data in some areas are usually scarce, but our PA-DDQN algorithm needs to train the model based on these historical data, which makes our mode unable to mine enough information through the sparse historical data. This will lead to a low quality of allocation results and will greatly reduce the success rate and sensing quality of tasks. Therefore, we need to further study the cold start problem of this method to ensure the performance of the algorithm in the case of sparse historical data.

The unpredictability of workers' route. Our current study is based on the fact that all workers have a fixed route, and all workers have the willingness to be dispatched by the platform. However, in many scenarios, the workers recruited by the platform do not have enough willingness to execute the allocated tasks, and the route of workers is difficult to predict. Therefore, in order to get a more detailed description of workers' personal preferences to improve the quality of allocation, we need to consider the prediction of workers' mobile routes to get the accurate position of each worker at the beginning of each sensing cycle, based on which we can ultimately improve the quality of the sensing results.

More complex task quality metric. In this paper, we measure the sensing quality of tasks only base on historical access data. In fact, there are many factors that affect the sensing quality, including characteristics of sensing tasks, temporal and spatial distribution, sensing requirements, as well as capabilities of workers and IOT devices. All of these will lead to different quality of sensing results. Therefore, we need to measure the sensing quality from at least two aspects, one is to measure quality according to the different types of sensing tasks, the other is to measure quality according to the sensing terminal itself. Building a practical task quality model is a complex problem, so we try to study whether we can extend more complex measurement methods in this scenario, which will be a new research work in the future.

VII. Conclusion

In this paper, We propose a new MCS task allocation framework called ISIATasker, which jointly consider the sensing and actuation to enable an instant sensing and then instant actuation. We designed a two-phase task allocation algorithm to implement ISIATasker, which consists of three key methods particular for solving ISIA problem. First, we propose a method called SPCP based on Pearson correlation coefficient to cluster sensing locations into several POIs and a descent greedy method SSFP for cooperative sensor set selection. Second, in order to maximize POI coverage and worker-task matching degree, we proposed a DDQN-based method called PA-DDQN to allocate an optimal POI set for each worker. We conduct extensive experiments based on real-world dateset: Chengdu check-in dataset and camera dataset. The results show that the performance of our method outperforms other baselines.

Acknowledgment

This work was supported in part by the National Key R&D Program of China (No. 2019YFB2102200), the National Science Fund for Distinguished Young Scholars (No. 61725205), and the National Natural Science Foundation of China (No. 61960206008, U2001207).

References

[1] R. K. Ganti, F. Ye, and H. Lei, "Mobile crowdsensing: current state and future challenges," *IEEE communications Magazine*, vol. 49, no. 11, pp. 32–39, 2011.

[2] B. Guo, Z. Wang, Z. Yu, Y. Wang, N. Y. Yen, R. Huang, and X. Zhou, "Mobile crowd sensing and computing: The review of an emerging human-powered sensing paradigm," *ACM computing surveys (CSUR)*, vol. 48, no. 1, pp. 1–31, 2015.

[3] Y. Liu, Z. Yu, B. Guo, Q. Han, J. Su, and J. Liao, "Crowdos: a ubiquitous operating system for crowdsourcing and mobile crowd sensing," *IEEE Transactions on Mobile Computing*, 2020.

[4] L. Kazemi and C. Shahabi, "Geocrowd: enabling query answering with spatial crowdsourcing," in *Proceedings of the 20th international conference on advances in geographic information systems*, 2012, pp. 189–198.

[5] S. Perez, "Gigwalk, an amazon turk for real world work, gets an upgraded backend for businesses'," 2013.

[6] P. Dutta, P. M. Aoki, N. Kumar, A. Mainwaring, C. Myers, W. Willett, and A. Woodruff, "Common sense: participatory urban sensing using a network of handheld air quality monitors," in *Proceedings of the 7th ACM conference on embedded networked sensor systems*, 2009, pp. 349–350.

[7] B. Guo, H. Chen, Z. Yu, X. Xie, S. Huangfu, and D. Zhang, "Fliermeet: a mobile crowdsensing system for cross-space public information reposting, tagging, and sharing," *IEEE Transactions on Mobile Computing*, vol. 14, no. 10, pp. 2020–2033, 2014.

[8] Y. Zheng, F. Liu, and H.-P. Hsieh, "U-air: When urban air quality inference meets big data," in *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2013, pp. 1436–1444.

[9] V. Coric and M. Gruteser, "Crowdsensing maps of on-street parking spaces," in *2013 IEEE International Conference on Distributed Computing in Sensor Systems*. IEEE, 2013, pp. 115–122.

[10] J. Wang, L. Wang, Y. Wang, D. Zhang, and L. Kong, "Task allocation in mobile crowd sensing: State-of-the-art and future opportunities," *IEEE Internet of Things Journal*, vol. 5, no. 5, pp. 3747–3757, 2018.

[11] B. Guo, Y. Liu, W. Wu, Z. Yu, and Q. Han, "Activecrowd: A framework for optimized multitask allocation in mobile crowdsensing systems," *IEEE Transactions on Human-Machine Systems*, vol. 47, no. 3, pp. 392–403, 2016.

[12] 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.

[13] H. Xiong, D. Zhang, G. Chen, L. Wang, and V. Gauthier, "Crowdtasker: Maximizing coverage quality in piggyback crowdsensing under budget constraint," in *2015 IEEE International Conference on Pervasive Computing and Communications (PerCom)*. IEEE, 2015, pp. 55–62.

[14] E. Wang, Y. Yang, J. Wu, K. Lou, D. Luan, and H. Wang, "User recruitment system for efficient photo collection in mobile crowdsensing," *IEEE Transactions on Human-Machine Systems*, vol. 50, no. 1, pp. 1–12, 2019.

[15] Y. Yang, W. Liu, E. Wang, and J. Wu, "A prediction-based user selection framework for heterogeneous mobile crowdsensing," *IEEE Transactions on Mobile Computing*, vol. 18, no. 11, pp. 2460–2473, 2018.

[16] H. Li, T. Li, W. Wang, and Y. Wang, "Dynamic participant selection for large-scale mobile crowd sensing," *IEEE Transactions on Mobile Computing*, vol. 18, no. 12, pp. 2842–2855, 2018.

[17] 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.

[18] C. Lai and X. Zhang, "Duration-sensitive task allocation for mobile crowd sensing," *IEEE Systems Journal*, vol. 14, no. 3, pp. 4430–4441, 2020.

[19] C. Dai, X. Wang, K. Liu, D. Qi, W. Lin, and P. Zhou, "Stable task assignment for mobile crowdsensing with budget constraint," *IEEE Transactions on Mobile Computing*, 2020.

[20] H. Wang, D. Zhao, H. Ma, and L. Ding, "Min-max planning of time-sensitive and heterogeneous tasks in mobile crowd sensing," in *2018 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2018, pp. 1–7.

[21] J. Wang, Y. Wang, D. Zhang, Q. Lv, and C. Chen, "Crowd-powered sensing and actuation in smart cities: Current issues and future directions," *IEEE Wireless Communications*, vol. 26, no. 2, pp. 86–92, 2019.

[22] M. Xiao, J. Wu, L. Huang, Y. Wang, and C. Liu, "Multi-task assignment for crowdsensing in mobile social networks," in *2015 IEEE Conference on Computer Communications (INFOCOM)*. IEEE, 2015, pp. 2227–2235.

[23] Z. Song, C. H. Liu, J. Wu, J. Ma, and W. Wang, "Qoi-aware multitask-oriented dynamic participant selection with budget constraints," *IEEE Transactions on Vehicular Technology*, vol. 63, no. 9, pp. 4618–4632, 2014.

[24] F. Yucel, M. Yuksel, and E. Bulut, "Qos-based budget constrained stable task assignment in mobile crowdsensing," *IEEE Transactions on Mobile Computing*, 2020.

[25] L. Wang, Z. Yu, Q. Han, D. Yang, S. Pan, Y. Yao, and D. Zhang, "Compact scheduling for task graph oriented mobile crowdsourcing," *IEEE Transactions on Mobile Computing*, 2020.

[26] E. Wang, Y. Yang, J. Wu, W. Liu, and X. Wang, "An efficient prediction-based user recruitment for mobile crowdsensing," *IEEE Transactions on Mobile Computing*, vol. 17, no. 1, pp. 16–28, 2017.

[27] E. Wang, M. Zhang, X. Cheng, Y. Yang, W. Liu, H. Yu, L. Wang, and J. Zhang, "Deep learning-enabled sparse industrial crowdsensing and prediction," *IEEE Transactions on Industrial Informatics*, 2020.

[28] X. Tao and A. S. Hafid, "Deepsensing: A novel mobile crowdsensing framework with double deep q-network and prioritized experience replay," *IEEE Internet of Things Journal*, vol. 7, no. 12, pp. 11 547–11 558, 2020.

[29] Y. Liu, B. Guo, Y. Wang, W. Wu, Z. Yu, and D. Zhang, "Taskme: Multi-task allocation in mobile crowd sensing," in *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, 2016, pp. 403–414.

[30] Y. Jing, B. Guo, Z. Wang, V. O. Li, J. C. Lam, and Z. Yu, "Crowdtracker: Optimized urban moving object tracking using mobile crowd sensing," *IEEE Internet of Things Journal*, vol. 5, no. 5, pp. 3452–3463, 2017.

[31] B. L. Golden, L. Levy, and R. Vohra, "The orienteering problem," *Naval Research Logistics (NRL)*, vol. 34, no. 3, pp. 307–318, 1987.

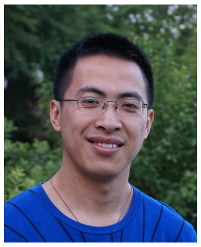
[32] G. Reinelt, "A traveling salesman problem library," *Journal of the Operational Research Society*, vol. 11, no. 1, pp. 19–21, 1992.

Houchun Yin received the bachelor's degree in Computer Science and Technology from Northwestern Polytechnical University, Xi'an, China in June 2019. He is currently a master candidate of computer application technology from Northwestern Polytechnical University, Xi'an, China. His current research interest is pervasive computing and mobile crowdsensing.

Zhiwen Yu received the Ph.D. degree in computer science from Northwestern Polytechnical University, Xi'an, China, in 2005. He is currently a Professor and the Dean of the School of Computer Science, Northwestern Polytechnical University, Xi'an, China. He was an Alexander Von Humboldt Fellow with Mannheim University, Germany, and a Research Fellow with Kyoto University, Kyoto, Japan. His research interests include ubiquitous computing, mobile crowdsensing and HCI.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2021.3095160, IEEE Internet of Things Journal

Liang Wang received his 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.



Jiangtao Wang received the PhD degree from Peking University, China, in 2015. He is currently an Associate Professor with Tenure in the Intelligent Healthcare Centre, Coventry University, UK. Before that, he was a lecturer with the School of Computing and Communications at Lancaster University, UK. His research interest includes mobile and pervasive computing, crowdsensing/crowdsourcing, and IoT.

Lei Han received the bachelor's degree and master's degree in engineering of computer science from Fuzhou University, Fuzhou, China in 2017 and 2020. He is currently a PH.D candidate of computer application technology from Northwestern Polytechnical University, Xi'an, China. His current research interest is pervasive computing and mobile crowdsensing.

Bin Guo received the PhD degree in computer science from Keio University, Japan, in 2009, and then was a postdoc researcher with Institut Tel ecom SudParis, France. He is a professor with Northwestern Polytechnical University, Xi'an, China. His research interests include ubiquitous computing, mobile crowd sensing, and HCI.