AoI-guaranteed Incentive Mechanism for Mobile Crowdsensing with Freshness Concerns

Yin Xu, *Student Member, IEEE*, Mingjun Xiao, *Member, IEEE*, Yu Zhu,
Jie Wu, *Fellow, IEEE*, Sheng Zhang, *Member, IEEE*, and Jinrui Zhou

Abstract—With the explosive spread of smart mobile devices, Mobile CrowdSensing (MCS) has been becoming a promising paradigm, by which a platform can coordinate a group of workers to complete large-scale data collection tasks using their mobile devices. In this paper, we investigate the incentive mechanism design in MCS systems, taking the freshness of collected data and social benefits into consideration. First, the Age of Information (AoI) metric is introduced to measure the freshness of data. Then, we model the incentive mechanism design with AoI guarantees as an incomplete information two-stage Stackelberg game with multiple constraints. Next, we consider the scenario that all participants share the public utility function parameters of the Stackelberg game. By deriving the optimal remuneration paid by the platform and the optimal data update frequency for each worker, and proving the existence of a unique Stackelberg equilibrium, we propose an AoI-guaranteed Incentive Mechanism (AIM) that enables the platform and all workers to maximize their utilities simultaneously. Furthermore, we extend AIM to a general scenario where each participant has no prior knowledge of the utility function parameters of the game. By resorting to the Deep Reinforcement Learning (DRL) technique and modeling the two-stage Stackelberg game as a Markov decision process, we propose a DRL-based Incentive Mechanism (DIM) with AoI guarantees, which makes each participant effectively seek its optimal strategy through trial and error. Meanwhile, the system can guarantee that the AoI values of all data uploaded to the platform are not larger than a given threshold. Finally, numerical experiments on real-world traces are conducted to validate the efficacy and efficiency of AIM and DIM.

Index Terms—Mobile Crowdsensing, Incentive Mechanism, Age of Information, Stackelberg Game, Deep Reinforcement Learning

1 Introduction

With the unprecedented prevalence of high-performance mobile devices and wireless communication networks, Mobile CrowdSensing (MCS) has become an attractive paradigm for collecting sensing data [2]–[5]. A typical MCS system is comprised of a cloud platform and a collection of mobile users (a.k.a., workers), through which service applicants can outsource their sensing tasks and recruit suitable workers to accomplish these tasks by using mobile devices (e.g., smartphones, wearables, etc.) [6]–[8]. Owing to the mobility of users as well as the diversity of massive embedded sensors, MCS can perform plenty of large-scale sensing tasks that individuals cannot deal with. Therefore, it has facilitated a corpus of practical applications, such as traffic data collection, air pollution monitoring, seismic amplitude sensing, etc [9]–[11]. Moreover, much effort has been devoted to investigating diverse MCS issues in recent years, including incentive mechanism designs [12]–[14], privacy-preserving approaches [15]–[17], task allocation schemes [18]–[20], and so on.

In this paper, we concentrate on the incentive mechanism design for MCS with concerns about the freshness of sensing data and workers' social benefits. Consider such an MCS scenario that the platform stimulates some workers via social networks to periodically collect the desired data (e.g., traffic data) from a group of Points of Interest (PoIs) so as to provide data services for requesters [9], [10]. Because data valuation and service quality largely depend on the timeliness of data, the platform will make efforts to collect fresh sensing data as much as possible. More precisely, the platform needs to guarantee that the Age of Information (AoI) values of all collected data are not larger than a certain threshold. Here, AoI, i.e., the elapsed time of data from being collected by the worker to being received and processed by the platform currently, is a widely-adopted application-independent metric to indicate data freshness [21]–[24]. On the other hand, the workers in the MCS system are social network users so that they can share their collected data with their social neighbors to obtain extra social benefits [25]–[28]. For example, when a worker's trajectory of collecting data covers its social neighbors' PoIs, it might piggyback to collect data for the neighbors, which can save their data collection time and costs as well as improve its own social reputation. Then, a critical issue is how to design the incentive mechanism to maximize the utilities (i.e., net profits) of the platform and workers simultaneously while taking the above two concerns into consideration.

There are three major challenges in the above-mentioned incentive mechanism design issue. First, the platform wishes to collect data with sufficient freshness, so the system needs to incentivize workers to frequently update their collected data on the platform (i.e., collect and upload the latest data copy to the

- *Y. Xu, M. Xiao (Corresponding author), and J. Zhou are with the School of Computer Science and Technology / Suzhou Institute for Advanced Research / the CAS Key Laboratory of Wireless-Optical Communications/ State Key Laboratory of Cognitive Intelligence, University of Science and Technology of China (USTC), Hefei, China. E-mail: {xuyin218@mail., xiaomj@, zzkevin@mail.}ustc.edu.cn.*
- *Y. Zhu is with the School of Data Science, University of Science and Technology of China (USTC), Hefei, China. E-mail: zhuyuzy@mail.ustc.edu.cn.*
- *J. Wu is with the Center for Networked Computing, Temple University, Philadelphia, PA 19122, USA. E-mail: jiewu@temple.edu.*
- *S. Zhang is with the State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, China. E-mail: sheng@nju.edu.cn.*

This paper is an extended version of the conference paper [1] published in IEEE INFOCOM 2023. This research was supported in part by the National Natural Science Foundation of China (NSFC) under Grants 62172386, 61872330, and 61572457, and in part by the Natural Science Foundation of Jiangsu Province in China under Grants BK20191194 and BK20131174.

platform). When workers increase their data update frequencies, the collected data will become fresher and the corresponding AoI values will be smaller, yielding higher data valuation and service quality. However, it may result in larger data collection costs for workers. Thus, there must be an optimal trade-off on the data update frequencies of all workers that needs to be balanced. Second, when workers update their data frequently, it might produce a congestion of data update due to the limited communication and processing capabilities on the platform side. Therefore, there exists a game among workers to compete for data update through queueing, which needs to be addressed in designing the payment strategy of the incentive mechanism. Third, since workers can bring extra social benefits to the whole system by sharing data, their social relationships need also to be considered. However, the social relationships are generally unknown or incomplete to the platform in many real-world MCS applications. Meanwhile, the platform normally needs some personal information shared by workers to determine the payment strategy, but such prior knowledge also may not be readily available in practice due to privacy reasons, which makes the design much more challenging.

So far, a considerable variety of incentive mechanisms have been proposed for MCS systems by utilizing diverse technologies, such as auction theory [29]–[31], Stackelberg game [32], [33], Bayesian game [34], Deep Reinforcement Learning (DRL) [35]–[38], etc. However, most of them have not discussed the freshness of collected data. Only a handful of studies have considered the pricing issue with AoI concerns [21]–[24]. For example, the authors in [23] developed a linear AoI-based payment mechanism. Nevertheless, none of these works investigate the incentive mechanism design with the concerns of AoI demands and workers' social benefits, which actually involves an incomplete information game with the AoI constraint. Additionally, although many researchers have devoted their efforts to addressing different AoI optimization problems [39]–[42], these solutions also cannot be leveraged to tackle the three challenges together.

Inspired by the above considerations, we first model the incentive mechanism design issue as an incomplete information two-stage Stackelberg game, in which the platform is treated as the leader and workers are seen as the followers. Meanwhile, we also append the AoI constraint for workers' data collection as well as the constraint on the total data update frequency incurred by the limited resources on the platform side to the Stackelberg game. Then, we consider the scenario that all participants share the public utility function parameters of the Stackelberg game. With the aid of the Karush-Kuhn-Tucker (KKT) conditions and a backward deduction approach, we derive the optimal strategies (i.e., the optimal remuneration paid by the platform and the optimal data update frequency for each worker), based on which we propose an AoI-guaranteed Incentive Mechanism (AIM). Next, we extend AIM to a general scenario that each participant has no prior knowledge of the utility function parameters of the game. By resorting to the powerful DRL technique and modeling the game as a finite Markov Decision Process (MDP), we further propose a DRL-based Incentive Mechanism (DIM) with AoI guarantees. Overall, our major contributions are summarized as follows:

- We introduce the incentive mechanism design problem for MCS systems with freshness concerns and turn it into a novel incomplete information two-stage Stackelberg game with constraints. Unlike existing studies, our problem takes into consideration the AoI values of data and workers' social benefits simultaneously.
- We utilize the AoI metric to measure the freshness of data and derive the closed-form expression for the AoI of the data that each worker uploads to the platform where workers' social influences with each other are also considered.
- We propose the mechanism, AIM, when all participants share the utility function parameters of the Stackelberg game. By deriving the optimal strategy for each participant, AIM can ensure that the platform and workers obtain their maximum utilities. Moreover, we theoretically prove that these optimal strategies constitute a unique Stackelberg equilibrium.
- We propose an extended mechanism, DIM, when each participant has no prior knowledge of the game. Based on the DRL technique, DIM enables the platform or each worker to quickly learn the optimal strategy directly from game experiences.
- We evaluate the performance of AIM and DIM with extensive simulations on real-world traces. The simulation results demonstrate the effectiveness of the proposed incentive mechanisms compared with some baseline methods.

The remainder of the paper is organized as follows. In Section 2, we introduce the system model and formulate our problem. The closed-form expression of the AoI of data is elaborated in Section 3. In Section 4, we propose AIM with theoretical analysis in great detail. Furthermore, we extend AIM to a general scenario and propose DIM in Section 5. The simulations and the evaluation results are illustrated in Section 6. We discuss the related works in Section 7 and conclude the paper in Section 8. Additionally, we summarize the major notations in Table 1 for ease of reference.

2 System Model & Problem

2.1 System Model

As illustrated in Fig. 1, we consider a typical MCS system, which is composed of a cloud platform and a group of workers. The platform has a long-term sensing task, e.g., collecting the latest traffic data from some PoIs. The workers, denoted by $\mathcal{N} \triangleq \{1, 2, \ldots, N\}$, are some social network users who are willing to share data with each other to obtain extra social benefits, e.g., saving data collection time and cost by piggyback, improving social reputation, and so on [25]–[28]. The workflow of the whole system can be concisely described as follows. At the beginning, the platform broadcasts the long-term sensing task with the corresponding requirements to all eligible workers. Then, the platform and workers will collaboratively determine an incentive mechanism, which includes the payment strategy for the platform and the data update strategy for each worker. Next, each worker needs to perform the sensing task and will continuously collect data from some specified PoIs. Note that the data of each worker is packed into packets of fixed-size. Meanwhile, the worker will upload the latest data to the platform with the data update frequency determined in advance. After that, the platform will repeatedly receive the data uploaded from each worker, conduct a data cleaning process, and utilize the cleaned data to update the corresponding last version (or directly store this cleaned data if it is the first version). Lastly, the platform will periodically pay the remuneration to each worker according to a pre-determined payment strategy and the worker's data update strategy until the whole sensing task is accomplished.

In the above MCS system, the platform will store the latest data copy uploaded by each worker in its cache. At the same time, in order to prevent conflicts, the platform will maintain a queue for cleaning the data from various workers. Without loss

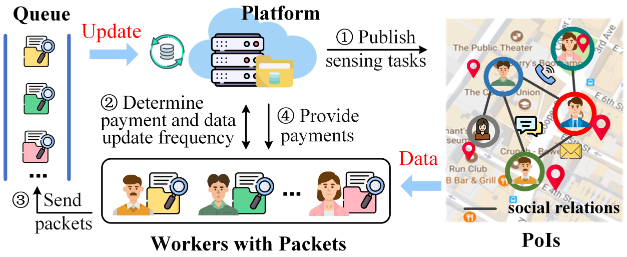


Fig. 1. System overview

of generality, the queue of data cleaning adopts the First-Come-First-Service (FCFS) strategy in this paper. For the purpose of avoiding congestion in the queue, the platform needs to ensure that the total data update frequency is not larger than a specified threshold. Besides, the platform hopes the collected data is as fresh as possible. Accordingly, the platform will record the AoI value of data uploaded by each worker and also try to keep the AoI values within a given threshold. For clarification, we define some important concepts and notations as follows.

Definition 1 (***Data Update Frequency and Total Frequency***)**.** The data update frequency of worker i refers to the frequency that the worker collects and uploads the data to the platform, denoted by p_i. We use $P \triangleq (p_1, p_2, \dots, p_N)$ and P_{-i} $(i \in \mathcal{N})$ to denote the data update frequencies of all workers and all workers except worker i, respectively. In addition, to avoid congestion in the queue, we assume that the total data update frequency of all workers is not larger than a constant $\hat{p}$, i.e., $\sum_{i=1}^{N} p_i \le \hat{p}$.

Definition 2 (***Unit-Remuneration***)**.** The remuneration that the platform pays to each worker i is proportional to its data update frequency. We call the remuneration per data update frequency paid to worker i as the unit-remuneration, denoted as R_i. Let $\mathcal{R} = \{R_1, R_2, ..., R_N\}$ be the unit-remunerations to all workers.

Definition 3 (***Age of Information, AoI***)**.** The AoI of data refers to the time elapsed since a worker collects this data. Specifically, the AoI of the data uploaded to the platform by worker i $(i \in \mathcal{N})$, called *worker i's data* or *data i* for short, is actually the difference between the current time t and the creation time $U_i(t)$ of this data, which can be defined as follows:

$$\delta_i(t) = t - U_i(t). \tag{1}$$

Note that each worker may need to collect data from multiple PoIs along a planned trajectory, so the collection time should be considered in $\delta_i(t)$. Moreover, if data i has been updated to the platform multiple times, $U_i(t)$ actually refers to the creation time of the latest version of this data up to time t.

Definition 4 (***Average AoI and AoI Threshold***)**.** Since the AoI of data might change over time, the average AoI will be put to use in practice. During a time interval of observation $(0, T)$, we define the average AoI of data i as follows:

$$\bar{\delta}_i = \frac{1}{T}\int_0^T \delta_i(t)dt. \tag{2}$$

Essentially, the average AoI $\bar{\delta}_i$ $(\forall i \in \mathcal{N})$ can be regarded as a function of p_i and P_{-i}, so we will occasionally employ the notation $\delta_i(p_i, P_{-i})$ to replace $\bar{\delta}_i$ for clarify. To guarantee the data freshness of each worker, we set a threshold ε to restrict the average AoI of each data, i.e., $\delta_i(p_i, P_{-i}) \le \varepsilon$.

Definition 5 (***Social Benefits*** [26]–[28], [32])**.** The workers are assumed to be social network users so that they can share data with each other via the social network and gain the social benefits from shared data. To be specific, a worker can obtain an additional income by leveraging the data shared by its social network neighbors (See Eq. (3)). Meanwhile, the worker might also ask neighbors to piggyback its desired data, which are exactly close to them so as to save the data collection time (See Theorem 1). Here, we employ an adjacency matrix $[\upsilon_{ij}]_{N\times N}$ to describe the social network, where υ_{ij} indicates the social network influence of worker j on worker i. From a statistical perspective, like in [25], [43], [44], we assume that the social relationships among workers remain relatively stable, so that the adjacency matrix of workers will not dynamically change during a certain time period. For simplicity, we exclude isolated workers and make the assumption that bilateral interactions are symmetric (i.e., $\upsilon_{ij} = \upsilon_{ji}$, $\upsilon_{ii} = 0$) due to the reciprocity of social relations [25].

TABLE 1
Description of major notations

Variable	Description
$\mathcal{N}, N$	the set of workers and the number of workers.
p_i	the data update frequency of worker i.
P, P_{-i}	the data update frequencies of all workers; P except p_i.
$R_i, \mathcal{R}$	the unit-remuneration to worker i and the payment vector.
p_i^*, R_i^*	the optimal data update frequency of worker i and the optimal unit-remuneration of the platform to worker i.
$\delta_i, \bar{\delta}_i$	the AoI value and average AoI value of worker i's data.
$\varepsilon, \hat{p}$	the AoI threshold and the total data update frequency.
Φ, Ω_i	the utility of the platform and the utility of worker i.
ψ_i, Ψ_i	the social network effects and the social benefits.
$\upsilon, F(f)$	the social network influence and the degree distribution.
μ, ρ	the serving rate of the platform and the offered load.
$p(f)$	the data update frequency of the worker with degree f.
$R(f)$	the unit-remuneration paid to the worker with degree f.
X_k, C_k	the inter-arrival time and the collection time.
W_k, H_k	the whole waiting time and the handling time.
$\overline{P_{-i}}, \underline{f}$	the average data update frequency of worker i's neighbors and the mean value of the degrees of the social network.
$s, \eta, a{\sim}d$	tunable parameters and the coefficients of functions.
$\Pi_{\{\cdot\}}, \mathcal{V}_{\{\cdot\}}$	the actor network and the critic network.
$\boldsymbol{\theta}, \boldsymbol{w}$	the parameters of the actor network and critic networks.
$\mathbf{s}^t, \mathbf{s}_i^t$	the states of the platform and worker i in the time slot t.
Υ^t, Υ_i^t	the rewards of the platform and worker i in the time slot t.
ξ, ϖ	the clipped Gaussian noise and the discount factor.

2.2 Problem Formalization

In the above MCS system, we need to determine the unit-remuneration for the platform and the data update frequency for each worker, in which two trade-offs need to be taken into consideration. Specifically, there is a trade-off on the data update frequencies of all workers. Meanwhile, the platform has also a trade-off between the gain from the collected data and the total payment to workers. To formulate these trade-offs, we give the definitions for the utilities of each worker and the platform.

Worker's Utility: The utility of worker i refers to the net profit of this worker, which can be defined as follows:

$$\begin{aligned}&\Omega_i(p_i, P_{-i}; s_i, a_i, b_i) = R(p_i) + \Psi_i(p_i, P_{-i}) - \Theta_i(p_i; s_i, a_i, b_i) \\ &= R_i p_i + \sum\nolimits_{j\in\mathcal{N}_i} \upsilon_{ij} p_i p_j - s_i(a_i p_i^2 + b_i p_i). \end{aligned} \tag{3}$$

In Eq. (3), the first term $R(p_i) = R_i p_i$ represents the remuneration that the platform pays to worker i. The second term $\Psi_i(p_i, P_{-i})$ refers to the social benefits. Like in [25], [27], [28], we adopt $\psi_i = \sum_{j\in\mathcal{N}_i} \upsilon_{ij} p_j$ and $\Psi_i(p_i, P_{-i}) = \sum_{j\in\mathcal{N}_i} \upsilon_{ij} p_i p_j$ to model the social network effects and social benefits of worker i respectively, where $\mathcal{N}_i$ denotes the set of all socially-connected neighbors of worker i. The third term $\Theta_i(p_i; s_i, a_i, b_i)$ is the cost function of worker i, which is assumed to be monotonically increasing, differentiable, and strictly convex. In this paper, we adopt a widely-used quadratic cost function like in [33], [43], [44], i.e., $\Theta_i(p_i;s_i,a_i,b_i)=s_i(a_ip_i^2+b_ip_i)$, where s_i, a_i, and b_i are positive parameters, and s_i denotes the equivalent monetary worth of the worker's data update frequency p_i.

Platform's Utility: The platform's utility is the revenue that it can gain from all collected data minus the total remuneration paid to all workers, which can be defined as follows:

$$\Phi(p_i,R_i;\eta,c,d)=\eta\sum\nolimits_{i=1}^{N}(cp_i-dp_i^2)-\sum\nolimits_{i=1}^{N}R_ip_i, \tag{4}$$

where η is a tunable parameter denoting the equivalent monetary worth. The first term denotes the revenue of the platform, which can be seen as a function of the data update frequencies of all workers. Like in [26], [43], [44], a linear-quadratic function is adopted to describe the income, i.e., $cp_i-dp_i^2$, where c and d are positive parameters characterizing the concavity extent of the function to capture the property of decreasing marginal returns. The second term is the sum of remunerations paid to all workers.

After defining the utility functions of workers and the platform, we model the AoI-guaranteed incentive mechanism design as a two-stage Stackelberg game, where the platform is the leader and the workers are the followers. For ease of exposition, we adopt $\mathbf{SG}(p_i,R_i;\oint)$ to denote the two-stage Stackelberg game, where $\oint=\{s_i,a_i,b_i|\forall i\in\mathcal{N}\}\cup\{\eta,c,d\}$ is the set of all participants' parameters. The objective of $\mathbf{SG}(p_i,R_i;\oint)$ is to achieve the Stackelberg Equilibrium (SE) with the constraints on total data update frequency and the AoI of data, which can be defined as:

Definition 6 (***Stackelberg Equilibrium with AoI Constraints***)**.** Let R_i^* and p_i^* stand for the optimal unit-remuneration paid by the platform to worker i and the optimal data update frequency of worker i, respectively. An optimal incentive strategy $\langle p_i^*,R_i^*\rangle$ constitutes an SE iff the following set of inequalities is satisfied:

Stage I (*Leader Game*):

$$\Phi(p_i^*,R_i^*;\eta,c,d)\geq\Phi(p_i,R_i;\eta,c,d); \tag{5}$$

Stage II (*Follower Game*):

$$\Omega_i(p_i^*,R_i^*;s_i,a_i,b_i)\geq\Omega_i(p_i,R_i;s_i,a_i,b_i); \tag{6}$$

$$\textit{Subject to}:\quad \delta_i(p_i,P_{-i})\leq\varepsilon,\quad \forall i\in\mathcal{N} \tag{7}$$

$$\sum\nolimits_{i=1}^{N}p_i\leq\hat{p}, \tag{8}$$

where Eq. (7) indicates that the AoI value of each worker's data is not larger than the given threshold, and the constraint of Eq. (8) signifies the constraint of total data update frequency.

The SE state shows that no one can improve its own utility by deviating from the optimal strategy. Based on the above game framework, we consider two scenarios, i.e., the parameters in $\oint$ are public or unknown, and then define two corresponding Stackelberg game problems, respectively. The detailed problem formulations are represented as follows:

Definition 7 (***Problem 1: Stackelberg game with Public Parameters, SPP***)**.** The SPP problem is to determine the optimal strategies (i.e., R_i^* and p_i^*) of the two-stage Stackelberg game $\mathbf{SG}(p_i,R_i;\oint)$, where the parameters in $\oint$ are public knowledge to the platform and each worker. Then, based on Def. 6, the problem is formulated as follows:

$$\langle p_i^*,R_i^*\rangle=\arg_{p_i,R_i}\mathbf{SG}(p_i,R_i;\oint)\ |\,\text{Eqs. (5)}\sim\text{(8) hold;} \tag{9}$$

$$\textit{Subject to}: s_i=s,a_i=a,b_i=b,\forall s_i,a_i,b_i\in\oint, \tag{10}$$

where Eq. (9) signifies the determination of optimal strategies while achieving the SE with AoI constraints, and Eq. (10) assumes that workers share the identical public parameters.

Definition 8 (***Problem 2: Stackelberg game with Unknown Parameters, SUP***)**.** The SUP problem is to find the optimal strategies (i.e., R_i^* and p_i^*) of the two-stage Stackelberg game $\mathbf{SG}(p_i,R_i;\cdot)$, where the platform and each worker have no prior knowledge of the game. Based on Def. 6, the problem is formulated as:

$$\langle p_i^*,R_i^*\rangle=\arg_{p_i,R_i}\mathbf{SG}(p_i,R_i;\cdot)\ |\,\text{Eqs. (5)}\sim\text{(8) hold.} \tag{11}$$

Different from Problem 1, Problem 2 removes the restriction in Eq. (10) (i.e., the parameter set $\oint$ is unknown). This means that each worker is allowed to possess a personalized utility function that is unknown to the platform and other workers.

Remark: In the aforementioned two problems, each worker i needs to determine its own optimal data update strategy p_i^* based on Eq. (3). Nonetheless, the second term $\Psi_i(p_i,P_{-i})$ involves other workers' strategies, which is unknown owing to the uncertainty of social network effects. As a result, the optimal strategy cannot be derived directly. Confronted with this uncertainty, we transform the follower game into a *Bayesian game with incomplete information*, which can be expressed as follows:

- The set of players $\mathcal{N}$ is a set of N workers;
- The action of player i is the data update frequency p_i;
- The type of player i is the social network effects ψ_i;
- The payoff of player i corresponding to its type ψ_i and its action p_i is the utility $\Omega_i(p_i,P_{-i};s_i,a_i,b_i)$;
- The strategy of player i is a function stating the action p_i for each type ψ_i at the unit-remuneration R_i, denoted by $\Gamma_i:\varphi_i\times\mathcal{R}\rightarrow\mathcal{P}_i$, where φ_i and $\mathcal{P}_i$ are the type space and action space of the worker i, respectively.

3 Characterizing AoI of Data

In this section, we derive the AoI of data as a premise for solving the two-stage Stackelberg game, where the influence of the social network on AoI is taken into consideration. First, we derive the average AoI for the case of a single worker as a building block. Then, we extend it to the case of multiple workers and calculate the closed-form expression of the AoI of each worker's data, which will be directly applied in the following sections.

3.1 Building Block: AoI of Data for A Single Worker

In this subsection, we deduce the closed-form expression of the average AoI of data in a queue system with a single source, in which the uploaded data can be seen as coming from a worker. As illustrated in Fig. 2, we first depict a sample curve of AoI along with time. We observe that the curve increases linearly with time when no update completes and is reset to a smaller value upon reception of a new data update. At the time t, the AoI of the data is $\delta(t)=t-t_1$. Here, up to time t, the latest version is received and updated by the platform at the time t_1', and thus the creation time of the latest version is the time t_1. Actually, AoI is the sum of the time the packet collected by the worker, the time it waited in the queue, and the time it spent in the data cleaning process.

According to the definition in Eq. (2), the average AoI can be seen as the area under the curve, which is calculated by the sum of the disjoint geometric part identified by $Q_k, k=\{1,2,...,I(T)\}$. Here, $I(T)$ indicates the number of data packets until the time T. We use $T_k=t_k'-t_k$ to represent the system time of the update, where t_k is the creation time and t_k' is the update time. Through accumulating all areas under the curve, we have

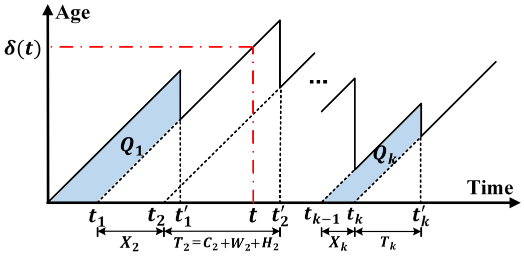


Fig. 2. Example of the AoI

$$\bar{\delta}^T = \frac{1}{T}\big(Q_1 + \sum\nolimits_{k=2}^{I(T)} Q_k + T^2_{I(T)}/2\big)$$
$$= \frac{Q_1 + T^2_{I(T)}/2}{T} + \frac{I(T)-1}{T}\frac{1}{I(T)-1}\sum\nolimits_{k=2}^{I(T)} Q_k. \quad (12)$$

From Fig. 2, we observe that the area Q_k can be calculated as the difference between the areas of two isosceles triangles. We define the interarrival time X_k to be the elapsed time between the generation of updates $k-1$ and k, i.e., $X_k = t_k - t_{k-1}$. Then, the area Q_k can be expressed as follows.

$$Q_k = \frac{1}{2}(T_k + X_k)^2 - \frac{1}{2}T_k^2 = X_k T_k + \frac{X_k^2}{2}. \quad (13)$$

By substituting Eq. (13) into Eq. (12), we further get

$$\bar{\delta}^T = \frac{Q_1 + T^2_{I(T)}/2}{T} + \frac{I(T)-1}{T}\frac{1}{I(T)-1}\sum\nolimits_{k=2}^{I(T)}\big[X_k T_k + \frac{X_k^2}{2}\big].$$

We notice that $Q_1 + T^2_{I(T)}/2$ represents a boundary effect that is finite with probability 1, and thus the first term in the above equality will vanish along with the growth of T. Let $p = \lim_{T\to\infty} I(T)/T$ indicate the data update frequency in the steady state. What's more, the remaining term will converge to the corresponding expected value when $I(T)$ approaches infinity. Hence, we obtain the average AoI of data:

$$\bar{\delta} = lim_{T\to\infty}\bar{\delta}^T = p(\mathbb{E}[XT] + \mathbb{E}[X^2/2]), \quad (14)$$

where $\mathbb{E}(\cdot)$ is the expectation operator. X and T are the random variables that correspond to the interarrival time and the system time of an update packet, respectively.

Now, we can derive the average AoI of a single worker's data. We take an example of the $M/M/1$ FCFS queue system with only one source, which includes three significant parameters, i.e., collection time β, data update frequency (a.k.a., arriving rate) p, and serving rate μ. In this system, the arriving time and serving time follow the Poisson distributions of $1/p$ and $1/\mu$, respectively. Moreover, the offered load is denoted by $\rho = p/\mu$. Inspired by [45], the average AoI of data can be easily derived using the queuing theory, i.e., $\bar{\delta} = [(\rho-1)(\rho^2 - \mu\rho\beta) + 1]/((1-\rho)\rho\mu)$.

3.2 AoI of Data for Multiple Workers

We extend the above closed-form expression of average AoI of data to the case of multiple workers, in which the AoI value of each worker may be affected by its neighbors. Consider the $M/M/1$ FCFS system with worker i's data update frequency p_i, collection time β_i, serving rate μ, and offered load $\rho_i = p_i/\mu$. The platform wishes to keep data fresh and encourage workers to frequently update data. Nevertheless, when worker i continuously sends its data with a high frequency, the AoI value for each other $N-1$ workers might have a rapid increase since it is a competitive N sources queue system with AoI constraints. Hence, the AoI values of some workers' data may exceed the specified threshold ε. To ensure that each worker's data can satisfy the constraints in Eqs. (7) and (8), we first need to derive the average AoI of each worker's data under the multi-source MCS system and the social network, as shown in the following theorem.

Theorem 1. *(**AoI for Multiple Workers**) N workers compete for the data update through an $M/M/1$ FCFS queue, in which each worker i's data update frequency, collection time, serving rate, and offered loads are p_i, β_i, μ, and ρ_i, respectively. Then, the average AoI $\bar{\delta}_i$ of worker i's data satisfies*

$$\bar{\delta}_i = \frac{\alpha\beta_i}{\sum_{j\in\mathcal{N}_i}\upsilon_{i,j}} + \frac{p_i/\mu^2}{1-\rho_{-i}}\big[\frac{\rho_i\rho_{-i}}{(1-\rho_{-i})^2} + \frac{\rho_i/(1-\rho)}{1-\rho_{-i}} + \frac{\rho_{-i}}{\rho_i}\big] + \frac{1}{\mu} + \frac{1}{p_i},$$

where $\rho = \sum\nolimits_{i=1}^{N}\rho_i$, $\rho_i = p_i/\mu$, *and* $\rho_{-i} = \sum\nolimits_{j\neq i}\rho_j$. (15)

Proof. The system time of an update T_k can be expressed by $C_k + W_k + H_k$, where C_k, W_k, and H_k represent the collection time, the whole waiting time in the system, and the handling time for data cleaning, respectively. Therefore, we have

$$\mathbb{E}(X_kT_k) = \mathbb{E}(X_kC_k) + \mathbb{E}(X_kW_k) + \mathbb{E}(X_kH_k)$$
$$= \mathbb{E}(X_k)\mathbb{E}(C_k) + \mathbb{E}(X_kW_k) + \mathbb{E}(X_k)\mathbb{E}(H_k). \quad (16)$$

The second equality holds since C_k and H_k are independent of X_k. If worker i collects data by itself from multiple PoIs, it will consume the collection time β_i. Consider that workers can share data via the social network. Worker i might obtain some data from its neighbors, and thus the expected collection time can be expressed as $\mathbb{E}(C_k) = \alpha\beta_i/\sum_{j\in\mathcal{N}_i}\upsilon_{i,j}$. Here, α is a constant coefficient. Meanwhile, we already know that there is $\mathbb{E}(H_k) = 1/\mu, \mathbb{E}(X_k) = 1/p_i$, and $\mathbb{E}(X_k^2) = 2/p_i^2$. Then, by substituting these equations and Eq. (16) into Eq. (14), the average AoI of worker i's data can be rewritten as follows.

$$\bar{\delta}_i = \alpha\beta_i/\sum\nolimits_{j\in\mathcal{N}_i}\upsilon_{i,j} + p_i\mathbb{E}(X_kW_k) + \frac{1}{\mu} + \frac{1}{p_i}. \quad (17)$$

To compute $\mathbb{E}(X_kW_k)$, we consider two cases: $A_k = \{X_k < T_{k-1}\}$ and $G_k = \{T_{k-1} < X_k\}$. Next, we rewrite $\mathbb{E}(X_kW_k)$ as $\mathbb{E}(X_kW_k) = \mathbb{E}(X_kW_k|G_k)P[G_k] + \mathbb{E}(X_kW_k|A_k)P[A_k]$. Inspired by the work [45], we can derive that $\mathbb{E}(X_kW_k|A_k) = \mathbb{E}_1 + \mathbb{E}_2$ with $\mathbb{E}_1 = \mathbb{E}[X_k(T_{k-1} - X_k)|A_k] = 1/(\mu^2(1-\rho)(1-\rho_{-i}))$ and $\mathbb{E}_2 = 2\rho_{-i}/(\mu^2(1-\rho_{-i})^2)$. That is, we have

$$\mathbb{E}(X_kW_k|A_k) = \mathbb{E}_1 + \mathbb{E}_2 = \frac{1}{\mu^2(1-\rho)(1-\rho_{-i})}$$
$$+ \frac{2\rho_{-i}}{\mu^2(1-\rho_{-i})^2} = \frac{\rho_{-i} - 2\rho\rho_{-i} + 1}{(1-\rho_{-i})^2(1-\rho)\mu^2}. \quad (18)$$

Similarly, we further derive $\mathbb{E}(X_kW_k|G_k)$, i.e.,

$$\mathbb{E}(X_kW_k|G_k) = \mathbb{E}[T_{k-1} + (X_k - T_{k-1})|G_k]\mathbb{E}[W_k|G_k]$$
$$= (\frac{1}{\mu - \mu\rho_{-i}} + \frac{1}{p_i})(\frac{\rho_{-i}}{\mu(1-\rho_{-i})}). \quad (19)$$

Based on the above equalities and the probability $P[A_i] = \rho_i/(1-\rho_{-i})$, $\mathbb{E}(X_kW_k)$ can be deduced as follows.

$$\mathbb{E}(X_kW_k) = \mathbb{E}(X_kW_k|A_k)P[A_k] + \mathbb{E}(X_kW_k|G_k)P[G_k]$$
$$= \frac{\rho_{-i} - 2\rho\rho_{-i} + 1}{(1-\rho_{-i})^2(1-\rho)\mu^2}\cdot\frac{\rho_i}{1-\rho_{-i}} +$$
$$(\frac{\rho_{-i}}{\mu^2(1-\rho_{-i})^2} + \frac{\rho_{-i}}{\mu p_i(1-\rho_{-i})})\cdot(1 - \frac{\rho_i}{1-\rho_{-i}})$$
$$= \frac{1}{\mu^2(1-\rho_{-i})}\big[\frac{\rho_i\rho_{-i}}{(1-\rho_{-i})^2} + \frac{\rho_i}{(1-\rho)(1-\rho_{-i})} + \frac{\rho_{-i}}{\rho_i}\big]. \quad (20)$$

Finally, we substitute Eq. (20) into Eq. (17) and get the average AoI of worker i's data, i.e., Eq. (15). □

4 AIM: Incentive Mechanism for SPP

In order to address the SPP problem, we propose the AoI-guaranteed Incentive Mechanism (called AIM) by leveraging the backward deduction approach and the KKT conditions. First, we introduce the distribution of workers' degrees and solve the follower game (i.e., Stage II) with the incomplete social benefit information, so that each worker i can determine its optimal data update frequency p_i^* under a given unit-remuneration R_i. Then, we turn to the leader game (i.e., Stage I) to derive the optimal unit-remuneration R_i^* paid by the platform. Finally, we present the detailed algorithm design of AIM and prove the existence of a unique Stackelberg equilibrium.

4.1 Solving the Bayesian Sub-Game

In Stage II, each worker's type (i.e., its social network effects) is uncertain to other workers, forming an incomplete information Bayesian sub-game. Due to the uncertainty of workers' types, we cannot obtain the expected utility for each type of worker, and the optimal strategy of the Bayesian sub-game also cannot be derived directly. Fortunately, many researchers have pointed out that workers' degrees in the social network can be used to indicate their social influences [26]–[28]. Therefore, we turn to exploit each worker's degree in the social network to derive its utility since the distribution of each worker's degree is public and known. Based on this idea, we can derive the expected utility function and then determine the closed-form expression of the optimal data update frequency for each worker.

First, we derive the expected utility of each worker according to Eq. (3). Specifically, we set the social network influence $\upsilon_{ij}=\upsilon$ for all i, j $(i \neq j)$ without loss of generality, where υ is a given social network effect coefficient. Note that υ can be treated as a variable instead of an exact value. Then, given the remuneration vector $\mathcal{R}$, the expected utility of worker i is

$$\begin{aligned}&\bar{\Omega}_i(p_i, P_{-i}, \mathcal{R}) = \mathbb{E}[\Omega_i(p_i, P_{-i}), \mathcal{R}]\\ &= R_i p_i + \upsilon p_i \mathbb{E}[\textstyle\sum_{j\in\mathcal{N}_i} p_j] - (ap_i^2 + bp_i)s. \qquad (21)\end{aligned}$$

Then, in order to determine the optimal data update frequency p_i^*, which can maximize $\bar{\Omega}_i(p_i^*, P_{-i}, \mathcal{R})$, we introduce the graph theory and harness the degree to describe the type of each worker. Specifically, we model the social network as an undirected graph whose structure can be described using different degrees. The degree of worker i is denoted as $f \in G$, where $G=\{1, \ldots, f^{max}\}$ and f^{max} is the maximum value of the degree. Let F be the probability distribution of the degree, denoted by $F : G \to [0, 1]$, where $\sum_{f\in G} F(f)=1$. To a certain extent, the distribution of the degree captures the social network effects from the network interaction patterns. Therefore, we can gain $E[\sum_{j\in\mathcal{N}_i} p_j]=f \times \overline{P_{-i}}$, where f represents the degree of worker i and $\overline{P_{-i}}$ is the average data update frequency of worker i's neighbors. Based on this, each worker's type can be transformed into the degree, and the utility of worker i in Eq. (21) can be rewritten as:

$$\bar{\Omega}_i(p_i, P_{-i}, \mathcal{R}) = R_i p_i + \upsilon p_i f \overline{P_{-i}} - (ap_i^2 + bp_i)s. \qquad (22)$$

Next, we need to figure out how to express $\overline{P_{-i}}$. Inspired by the study [46], we introduce the concept of "Configuration Model" in network science to model a randomly generated network. We concentrate on the Bayesian game with the symmetric type space, i.e., the workers with the same type f will choose the same data update frequency $p(f)$ and will be awarded the same remuneration $R(f)$ per data update frequency, which is a widely-adopted assumption in the social network works [28], [47], [48]. According to the property of the configuration model and the Bayes' rule, for a worker, the degree distribution of its randomly selected neighbor can be expressed as follows:

$$\overline{F}(f) = F(f)f/(\textstyle\sum_{f'\in G} F(f')f') = F(f)f/\underline{f}, \qquad (23)$$

where $\underline{f} = \sum_{f'\in G} F(f')f'$ indicates the mean value of the degrees of the whole social network. We can conclude that a randomly chosen social network neighbor of worker i has the degree distribution $\overline{F}(f)$. Therefore, each worker can treat $\overline{F}(f)$ as its neighbors' degree distribution, although they might not know the exact values of degrees. Based on Eq. (23), we further compute the average data update frequency of neighbors of a worker with degree f, denoted by $\overline{P_{-f}}$, i.e., $\overline{P_{-f}} = \sum_{f\in G} \overline{F}(f)p(f)$. Then, we substitute $\overline{P_{-f}}$ into Eq. (22), and the utility of the worker with degree f will be represented as follows:

$$\bar{\Omega}_f(p(f), P_{-f}) = R(f)p(f) + \upsilon p(f) f \overline{P_{-f}} - (ap^2(f) + bp(f))s. \quad (24)$$

Now, we can solve the Bayesian sub-game to determine the optimal strategy of each worker to maximize the above utility function and achieve the Bayesian Nash Equilibrium (BNE). Before presenting the follower's optimal strategy in Theorem 2, we first give the definition of BNE.

Definition 9 (***Bayesian Nash Equilibrium, BNE***)**.** A BNE is defined as a strategy profile that maximizes the expected payoff of each player for the given types and strategies performed by other players. A strategy vector $\mathit{\Gamma} = (\mathit{\Gamma}_1(\psi_1), \mathit{\Gamma}_2(\psi_2), \ldots, \mathit{\Gamma}_N(\psi_N))$ is a Bayesian nash equilibrium if and only if the following condition is satisfied for each player i:

$$\mathit{\Gamma}_i(\psi_i) \in argmax_{p_i\in\mathcal{P}_i} \Omega_i(p_i, \mathit{\Gamma}_{-i}, \psi_i, \psi_{-i}). \qquad (25)$$

Theorem 2. ***(Follower's Optimal Strategy).*** *Given any unit-remuneration $R(f)$, the closed-form expression of the action (i.e., data update frequency) of the follower game is*

$$p(f) = \frac{1}{2as}R(f) - \frac{b}{2a} + \frac{\upsilon f(\overline{R} - bs)}{2as(2as - \upsilon\overline{f})}, \qquad (26)$$

where $\overline{R} = \sum_{f\in G} \overline{F}(f)R(f)$ and $\overline{f} = \sum_{f\in G} \overline{F}(f)f$.

Proof. In order to acquire the closed-form solution of the unique BNE point of the follower game, we first apply the partial derivative of the expected utility in Eq. (24) and get

$$\frac{\partial\bar{\Omega}_f(p(f), P_{-f}, \mathcal{R})}{\partial p(f)} = R(f) + \upsilon f \overline{P_{-f}} - (2ap(f) + b)s. \qquad (27)$$

Then, we let $\partial\bar{\Omega}_f(p(f), P_{-f}, \mathcal{R})/\partial p(f) = 0$ and obtain

$$p(f) = \frac{1}{2as}R(f) - \frac{b}{2a} + \frac{\upsilon f}{2as}\overline{P_{-f}}. \qquad (28)$$

Since our social network is treated as a configuration model with dense types, we have the following approximation: $\overline{P_{-f}} = \mathbb{E}[p_j||j \in N_i] \approx \mathbb{E}[p(l)||l \in G]$. Thus, we have

$$p(f) = \frac{1}{2as}R(f) - \frac{b}{2a} + \frac{\upsilon f}{2as}\mathbb{E}[p(l)||l \in G]. \qquad (29)$$

By plugging Eq. (29) into $\overline{P_{-f}}=\sum_{f\in G} \overline{F}(f)p(f)$, we get

$$\begin{aligned}&\overline{P_{-f}}=\sum\nolimits_{f\in G} \overline{F}(f)\left[\frac{R(f)}{2as} - \frac{b}{2a} + \frac{\upsilon f}{2as}\mathbb{E}[p(l)||l \in G]\right],\\ &\Rightarrow \mathbb{E}[p(l)||l \in G] = \frac{1}{2as}\overline{R} - \frac{b}{2a} + \frac{\upsilon\overline{f}}{2as}\mathbb{E}[p(l)||l \in G],\\ &\Rightarrow \overline{P_{-f}} \approx E[p(l)||l \in G] = (\overline{R} - bs)/(2as - \upsilon\overline{f}), \qquad (30)\end{aligned}$$

where $\overline{R} = \sum_{f\in G} \overline{F}(f)R(f)$ and $\overline{f} = \sum_{f\in G} \overline{F}(f)f$. By substituting Eq. (30) into Eq. (29), we can get the closed-form expression of data update frequency and finish the proof. □

According to Theorem 2, when a worker wants to determine the value of $p(f)$ in Eq. (26), it only needs to know its own type f and the type distribution of neighbors instead of the exact type values of other workers. As a result, each follower's optimal strategy with incomplete information is solved. It is noteworthy that the follower's strategy $p(f)$ relies on the strategy $R(f)$ of the leader (i.e., the platform), so it is necessary to derive the optimal unit-remuneration for the platform in the next subsection.

4.2 Solving the Leader Game with Constraints

As the leader, the platform hopes to maximize its expected utility $\mathbb{E}[\Phi]$ by finding the optimal unit-remuneration $R^*(f)$ for each worker. After applying the configuration model, the expected utility of the platform can be expressed as follows.

$$\begin{aligned}\bar{\Phi} &= \mathbb{E}[\Phi] = \mathbb{E}[\eta \sum\nolimits_{i=1}^{N}(cp_i - dp_i^2) - \sum\nolimits_{i=1}^{N} R_i p_i] \\ &= N\sum\nolimits_{f\in G} F(f)\big[\big(\eta c - R(f)\big)p(f) - \eta d p^2(f)\big], \qquad (31)\end{aligned}$$

where $F(f)$ is known in advance. When taking the AoI constraint and the total data update frequency constraint into consideration simultaneously (i.e., Eqs. (7) and (8)), the optimization objective of the platform can be rewritten as follows:

$$\begin{aligned}&\textbf{max} \quad \bar{\Phi}(R(f)) \\ &\text{s.t.} \quad g(R(f)) = \delta_f(R(f)) - \varepsilon \le 0, \\ &\qquad g'(R(f)) = N\sum\nolimits_{f} F(f)p(f) - \hat{p} \le 0. \qquad (32)\end{aligned}$$

To find the optimal solution, we construct the Lagrangian function: $\mathcal{L}(R(f),\zeta) = \bar{\Phi}(R(f)) + \zeta_1 g(R(f)) + \zeta_2 g'(R(f))$, where ζ_1 and ζ_2 represent the Lagrangian multipliers. Since it is a convex optimization problem, the optimal solution must satisfy the Karush-Kuhn-Tucker (KKT) optimality conditions:

$$\begin{aligned}&\partial\mathcal{L}/\partial R(f)|_{R(f)=R^*(f)} = 0;\ \zeta_1 g(R(f)) = 0;\ \zeta_2 g'(R(f)) = 0; \\ &g(R(f)) \le 0;\ g'(R(f)) \le 0;\ \zeta_1 \le 0;\ \zeta_2 \le 0. \qquad (33)\end{aligned}$$

To meet the KKT conditions, we consider four cases:

(i) Case 1: $\zeta_1 = 0, \zeta_2 = 0$. When the optimal solution of maximizing $\bar{\Phi}(R(f))$ just falls within the feasible region (not including the boundary), the limitation of the feasible region does not work. Therefore, we can solve Eq. (32) by letting the first-order derivative of $\bar{\Phi}(R(f))$ be equal to zero directly. First, we compute the related first-order derivatives, i.e.,

$$\begin{aligned}&\frac{\partial \overline{R}}{\partial R(f)} = \overline{F}(f), \frac{\partial p(l)}{\partial R(f)} = \frac{\upsilon l \overline{F}(f)}{2as(2as - \upsilon\overline{f})} = \Delta\overline{F}(f)l \ \ (l \ne f), \\ &\frac{\partial p(f)}{\partial R(f)} = \frac{1}{2as} + \frac{\upsilon f \overline{F}(f)}{2as(2as - \upsilon\overline{f})} = \frac{1}{2as} + \Delta\overline{F}(f)f. \qquad (34)\end{aligned}$$

Here, we define $\Delta = \frac{\upsilon}{2as(2as - \upsilon\overline{f})}$ for ease of presentation. Based on Eq. (34), we derive the derivation $\partial\bar{\Phi}/\partial R(f)$ as follows:

$$\begin{aligned}\frac{\partial\bar{\Phi}}{\partial R(f)} =& NF(f)\big[-p(f) + \big(\eta c - R(f) - 2\eta d p(f)\big)(\frac{1}{2as} \\ &+\Delta\overline{F}(f)f)\big] + N\sum\nolimits_{l\ne f} F(l)\big[(\eta c - R(l) - 2\eta d p(l))\Delta\overline{F}(f)f\big].\end{aligned}$$

Because our social network is seen as a configuration model with large numbers of workers, we have the approximation:

$$\begin{aligned}&\sum\nolimits_{l\ne f} F(l)\big[(\eta c - R(l) - 2\eta d p(l))\Delta\overline{F}(f)f\big] \\ &= \Delta\overline{F}(f)\sum\nolimits_{l\in G} F(l)l\big(\eta c - R(l) - 2\eta d p(l)\big). \qquad (35)\end{aligned}$$

Based on the formulas defined in Section 4.1, i.e., $\overline{R} = \sum_{f\in G}\overline{F}(f)R(f)$, $\overline{f} = \sum_{f\in G}\overline{F}(f)f$, and $\underline{f} = \sum_{f\in G}F(f)f$, we can easily acquire $\sum_{f\in G}F(f)f^2 = \overline{f}\underline{f}$. For convenient presentation, we let $\Lambda = \sum_{f\in G}F(f)fR(f)$. Afterwards, we let $\partial\bar{\Phi}/\partial R(f) = 0$ and gain the following equation:

$$\begin{aligned}&\big[\frac{1}{2as} + (\Delta\overline{F}(f)f + \frac{1}{2as})(1 + \frac{\eta d}{as})\big]F(f)R(f) \\ &= F(f)\big[\frac{b}{2a} - \tilde{\Delta}f + (\Delta\overline{F}(f)f + \frac{1}{2as})\big(\eta c + \frac{b\eta d}{a} - 2f\eta d\tilde{\Delta}\big)\big] \\ &\quad +\Delta\overline{F}(f)\big[-(1 + \frac{\eta d}{as})\Lambda + \frac{\eta\underline{f}(ac + bd)}{a} - 2\eta d\tilde{\Delta}\underline{f}\overline{f}\,\big], \qquad (36)\end{aligned}$$

where $\tilde{\Delta} = \Delta(\overline{R} - bs)$. Based on Eq. (36), we can derive the expression of $R(f)$ easily. However, the expression of $R(f)$ still consists of unknown parameters $\overline{R} = \sum_{f\in G}\overline{F}(f)R(f)$ and $\Lambda = \sum_{f\in G}F(f)fR(f)$, which will be worked out below.

Since the degree distribution is known, we can directly calculate an algebraic expression of $\overline{F}(f)$. By substituting $R(f)$ and $\overline{F}(f)$ into the definitions of $\overline{R}$ and Λ, we can get two equations that together formulate a system of linear equations with two unknowns, i.e., $\overline{R}$ and Λ. In this way, we can solve the system of simultaneous equations to get the closed expressions of $\overline{R}$ and Λ, defined as $\overline{R^*}$ and Λ^*. Thereafter, we substitute the closed expressions of $\overline{R^*}$ and Λ^* into Eq. (36) and then obtain the closed expression of $R(f)$ as follows:

$$\begin{aligned}R^*(f) =& \frac{2a^2s^2}{(2as\Delta\overline{F}(f)f + 1)(as + \eta d) + as}\big[\frac{b}{2a} - \Delta f(\overline{R^*} - bs) \\ &+(\Delta\overline{F}(f)f + \frac{1}{2as})\big(\eta c + \frac{b\eta d}{a} - 2\Delta f\eta d(\overline{R^*} - bs)\big) \\ &+\Delta f\big(-(1 + \frac{\eta d}{as})\frac{\Lambda^*}{\underline{f}} + \frac{\eta(ac + bd)}{a} - 2\Delta\eta d(\overline{R^*} - bs)\overline{f}\big)\big]. \qquad (37)\end{aligned}$$

After the platform has determined the optimal unit-remuneration $R^*(f)$ based on Eq. (37), each worker i can also determine its own optimal data update frequency $p_i^*(f)$ by substituting $R^*(f)$ and $\overline{R^*}$ into Eq. (26). In short, the closed-form expression of the optimal data update frequency of worker i is

$$p_i^*(f) = \frac{1}{2as}R^*(f) - \frac{b}{2a} + \frac{\upsilon f(\overline{R^*} - bs)}{2as(2as - \upsilon\overline{f})}. \qquad (38)$$

By plugging $R^*(f)$ and $p_i^*(f)$ into the utility functions (i.e., $\bar{\Phi}$ and $\bar{\Omega}_i$), both the platform and each worker can reap their maximum expected utilities, denoted as Φ^* and Ω_i^*, i.e.,

$$\Phi^* = N\sum\nolimits_{f\in G} F(f)\big[\big(\eta c - R^*(f)\big)p^*(f) - \eta d p^{*2}(f)\big]. \qquad (39)$$

$$\Omega_i^* = R^*(f)p^*(f) + \upsilon p^*(f)f\overline{P_{-f}} - (ap^{*2}(f) + bp^*(f))s. \qquad (40)$$

(ii) Case 2: $\zeta_1 \ne 0, \zeta_2 = 0$. In this case, $R^*(f)$ and $p_i^*(f)$ derived by Case 1 are not eligible under the AoI constraint. Thus, we attain $R^*(f)$ by letting $g(R(f))$ be equal to zero, i.e.,

$$\frac{\alpha\beta}{f\upsilon} + \frac{p^2(f)}{\mu^2\check{\rho}}\left[\frac{\rho_{-i}(f)}{\mu(\check{\rho})^2} + \frac{1}{\check{\rho}\mu(1 - \rho(f))} + \frac{\rho_{-i}(f)\mu}{p^2(f)}\right] + \frac{1}{\mu} + \frac{1}{p(f)} = \varepsilon, \quad (41)$$

where $\check{\rho} = 1 - \rho_{-i}(f)$, $\rho(f) = \sum_f p(f)/\mu$, $\rho_{-i}(f) = (\sum_f p(f) - p(f))/\mu$, and β is the identical collection time. Combining with $\partial\bar{\Phi}/\partial R(f) + \partial g/\partial R(f) = 0$, we can derive $R^*(f)$ through solving Eq. (41) and further gain $p_i^*(f)$ as well.

(iii) Case 3: $\zeta_1 = 0, \zeta_2 \ne 0$. In this case, $R^*(f)$ and $p_i^*(f)$ derived by Case 1 cannot meet the total data update frequency constraint, so we need to calculate the derivation $\partial\bar{\Phi}/\partial R(f) + \zeta_2\partial g'/\partial R(f)$ and let it be equal to zero. Then, we attain

$$\begin{aligned}&\frac{\eta c F(f)}{2as} - \frac{F(f)R(f)}{2as} - (1 + \frac{\eta d}{as})F(f)p(f) + \Delta\overline{F}(f)(\eta c\underline{f} - \underline{f}\overline{R}) \\ &-2\eta d\Delta\overline{F}(f)\sum\nolimits_{l} lF(l)p(l) + \zeta_2\partial g'/\partial R(f) = 0. \qquad (42)\end{aligned}$$

Through solving the above equation, we gain $R^*(f)$:

$$R^*(f) = \frac{as(\eta c - bs + \zeta_2)}{2as + \eta d}(1 - \frac{f}{\underline{f}}) + \frac{2asf\hat{p}}{N\underline{f}} + bs - 2as\Delta f(\overline{R} - bs),$$

where ζ_2 can be derived by solving $g'(R(f)) = 0$.

(iv) Case 4: $\zeta_1 \ne 0, \zeta_2 \ne 0$. When the solutions in the above cases do not satisfy Eq. (32), we need to solve the equations:

$$\frac{\partial\bar{\Phi}}{\partial R(f)} + \frac{\partial g}{\partial R(f)} + \frac{\partial g'}{\partial R(f)} = 0;\ g(R(f)) = 0;\ g'(R(f)) = 0. \quad (43)$$

It is perplexing to obtain the closed-form $R^*(f)$ by solving Eq. (43), and we can adopt some mathematical approximating methods (e.g., the bisection, Newton's method, and so on) to acquire an approximation of $R^*(f)$.

4.3 The Detailed Algorithm Design

Based on the above idea, we propose the AoI-guaranteed Incentive Mechanism (AIM) for MCS systems, as illustrated in Algorithm

```
Algorithm 1: The AoI-guaranteed Incentive Mechanism
   Input  : degree distribution $F(f)$, worker $i$'s degree $f$, and
            some public parameters $a, b, c, d, \eta, s$;
   Output: $R^*(f)$, $p^*(f)$, $\Phi^*$, and $\Omega_i^*$;
1  Platform: Determine its tentative optimal strategy (i.e., the
     unit-remuneration $R^*(f)$) according to Eq. (37);
2  for each worker i, $i \in \mathcal{N}$ do
3      Determine its tentative optimal strategy (i.e., the data
         update frequency $p_i^*(f)$) based on $R^*(f)$ and Eq. (38);
4  if $\delta_i(p_i, P_{-i}) \le \varepsilon$ for $\forall i$ then
5      if $\sum_{i=1}^{N} p_i \le \hat{p}$ then
6          Platform: Obtain $\Phi^*$ according to Eq. (39);
7          Worker $i$: Obtain $\Omega_i^*$ according to Eq. (40);
8      else Solving Eq. (42) and $g'(R(f))=0 \Rightarrow R^*(f)$;
9              Platform: Update its strategy as $R^*(f)$;
10             Worker $i$: Update $p_i^*(f)$ based on $R^*(f)$;
11             Calculate $\Phi^*$ and $\Omega_i^*$ based on Eqs. (39) and (40);
12 else
13     if $\sum_{i=1}^{N} p_i \le \hat{p}$ then
14         Solving Eq. (41) and $\partial\mathcal{L}/\partial R(f)=0 \Rightarrow R^*(f)$ ;
15     else Solving Eq. (43)$\Rightarrow R^*(f)$;
16     Platform and Workers: Update $p_i^*(f), R^*(f), \Phi^*, \Omega_i^*$;
```

1. First, the leader (i.e., the platform) gives its strategy (i.e., the optimal unit-remuneration $R^*(f)$) according to Eq. (37) (Step 1). Then, each follower (i.e., each worker) determines its optimal strategy (i.e., the optimal data update frequency $p_i^*(f)$) based on the strategy of the platform (Steps 2-3). Note that apart from public parameters, each worker only knows its own degree and degree distribution while it does not know other workers' types and strategies. Next, the AoI of data can be calculated for multiple workers according to Theorem 1, and the platform needs to check whether the AoI of data is not larger than ε (Steps 4-16). As aforementioned, there exist four cases that need to be considered. In Steps 5-7, if there is $\sum_{i=1}^{N} p_i \le \hat{p}$, we can directly obtain the maximum utilities of the platform and each worker according to Eqs. (39) and (40). Otherwise, the strategy of the platform will be adjusted according to diverse cases, and the data update frequency of each worker will be updated accordingly (Steps 8-16). Finally, according to the determined payment strategy, the platform will pay the corresponding remuneration to the worker with the current degree f. Moreover, the computation complexity is $O(N)$.

4.4 The Equilibrium Analysis

We analyze the Bayesian sub-game equilibrium and the Stackelberg game equilibrium in this subsection.

Lemma 1. *The follower game exists at least one pure BNE.*

Proof. The work in [49] has pointed out that if the Bayesian sub-game satisfies the (Milgrom-Shannon) Single Crossing Property of Incremental Returns (SCP-IR), the Bayesian sub-game has at least one pure BNE. Based on Eq. (27), we have

$$\partial^2\bar{\Omega}_i(p_i, P_{-i}, \mathcal{R})/\partial p_i \partial\overline{P_{-i}} = \upsilon f > 0, \tag{44}$$

$$\partial^2\bar{\Omega}_i(p_i, P_{-i}, \mathcal{R})/\partial p_i^2 = -2as < 0. \tag{45}$$

Therefore, the follower game in Stage II meets the SCP-IR and there exists at least one pure BNE. □

Lemma 2. *[50] For the Bayesian sub-game, there exists at most one equilibrium if the following condition is satisfied:*

$$\left| \frac{\partial^2\bar{\Omega}_i(p_i, P_{-i}, \mathcal{R})}{\partial p_i \partial\overline{P_{-i}}} \Big/ \frac{\partial^2\bar{\Omega}_i(p_i, P_{-i}, \mathcal{R})}{\partial p_i^2} \right| < 1, \forall i \in \mathcal{N}. \tag{46}$$

According to Lemmas 1 and 2, when the condition $\upsilon f^{max} - 2as < 0$ is satisfied, the uniqueness of the BNE of the follower Bayesian sub-game can be guaranteed. Based on this, we prove the existence of the unique Stackelberg equilibrium.

Theorem 3. *The optimal incentive strategy $\langle p^*(f), R^*(f)\rangle$ determined by AIM constitutes the unique Stackelberg equilibrium while satisfying AoI constraints.*

Proof. In the whole two-stage Stackelberg game, each stage can derive its optimal closed-form solution: the payment strategy of the platform and the data update strategies of workers. As the role of the leader in Stage I, the platform can uniquely determine $R^*(f)$ according to Section 4.2. It is worth mentioning that the value of $R^*(f)$ is calculated just by the known distribution and some public parameters in $\oint$. That is, the value of $R^*(f)$ is only associated with the constant input without knowing workers' strategies and social structure information, and the platform cannot gain a larger utility if it adopts other strategies. When the optimal unit-remuneration is determined, workers can pick their optimal data update strategies based on Eq. (38), and these strategies constitute the unique Bayesian sub-game equilibrium. In a word, each stage has a unique equilibrium under the optimal incentive strategy $\langle p^*(f), R^*(f)\rangle$, and no one can improve its own utility by deviating from the optimal strategy during the process. At last, Eq. (7) guarantees that the AoI values of all workers' data are not larger than the given threshold. Thus, we can conclude that the two-stage Stackelberg game of AIM has the unique Stackelberg equilibrium while meeting AoI constraints. □

5 DIM: Incentive Mechanism for SUP

In this section, we design a DRL-based Incentive Mechanism (DIM) with AoI guarantees to address the SUP problem. In practice, the utility function parameters of the Stackelberg game might imply some sensitive information and would be unknown for privacy considerations. Therefore, we extend AIM to a more general scenario where the platform and workers have no prior knowledge of the Stackelberg game. In this case, by means of the strong DRL technique, the platform can conduct multiple interactions with workers to learn the optimal strategies directly from game experiences. In the following, we first introduce the network framework and the overall learning process. Then, we present the detailed procedures of the network training. Based on this, we propose DIM, which consists of the payment strategy searching algorithm for the platform and the data update strategy searching algorithm for each worker.

5.1 Learning Framework

Deep Reinforcement Learning (DRL) has gained recognition as an ideal technology for smoothly tackling a wide range of complex real-world decision-making problems. In this paper, we employ the Twin Delayed Deep Deterministic Policy Gradient (TD3) [51], a state-of-the-art model-free off-policy DRL solution, to derive an optimal strategy without relying on the prior information of the two-stage Stackelberg game. TD3 is an advanced version of the Deep Deterministic Policy Gradient (DDPG) algorithm, known for significantly improving both the performance and learning speed of DDPG in continuous action settings. This improvement stems from considering the interplay between function approximation

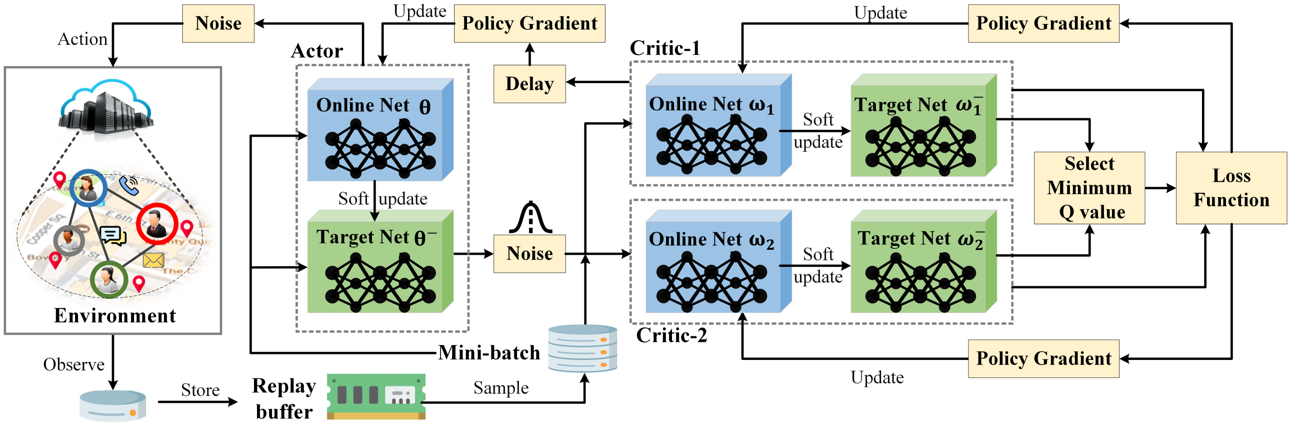


Fig. 3. Network framework for the proposed mechanism DIM

errors in both the policy and value updates, which can be easily incorporated into other actor-critic algorithms. Note that our goal is to design an AoI-guaranteed incentive mechanism based on an existing DRL network architecture. Therefore, we just focus on the searching process of the optimal strategies rather than the comparison of DRL algorithms.

As illustrated in Fig. 3, TD3 contains two types of networks, i.e., one actor network and two critic networks. Each actor (or critic) network is composed of two sub-networks: online network and target network. Accordingly, TD3 consists of six neural networks with proprietary parameters. Actor network parameters $\boldsymbol{\theta} = \{\theta, \theta^-\}$ mean the online parameter and target network parameter, respectively. Critic network parameters $\boldsymbol{w} = \{w_1, w_1^-, w_2, w_2^-\}$ denote the online parameter and target network parameter of critic network 1, as well as the online parameter and target network parameter of critic network 2. For notational simplicity, we use $\Pi_\theta, \mathcal{V}_{w_1}$, and $\mathcal{V}_{w_2}$ to denote the online actor network, the first online critic network, and the second online critic network, respectively. Each of them corresponds to a target network, denoted by $\Pi_{\theta^-}, \mathcal{V}_{w_1^-}$, and $\mathcal{V}_{w_2^-}$. During the training process, TD3 harnesses an experience replay buffer to store the historical interaction information. TD3 has an outstanding performance in handling continuous action spaces, improving training stability, balancing exploration and exploitation, and so on. Compared with DDPG, TD3 adds three techniques to solve the overestimation bias problem and exhibit more advantages, i.e.,

- Clipped double Q-learning: TD3 learns two critic networks instead of one. By selecting the minimum Q-value of the target networks, TD3 can achieve more accurate value estimations and the overestimation bias can be significantly reduced.
- Delayed policy updates: The update frequency of the actor network is less than the critic networks. This delay is conducive to mitigating policy oscillations, reducing the overfitting risks, and stabilizing the training process.
- Target policy smoothing regularization: TD3 injects clipped random noise into the action, which can smooth the value estimate by bootstrapping off of similar state-action value estimates. This promotes more diverse exploration and reduces the tendency to exploit a single action.

The learning process operates in a time-slotted manner, and the whole process is divided into T time slots. At each time slot t, the platform plays the role of the game leader, announcing the payment to each worker. Subsequently, workers act as game followers and determine their respective data update frequencies. More explicitly, based on the current state observed from the environment, the platform or each worker can map the state to an appropriate action and will execute the action. Upon observing the impact of the action on the environment, the platform or each worker will receive an immediate reward, and the current state is transited to the next state. After that, the experience (i.e., the tuple of the current state, the action, the reward, and the next state) is saved in the finite-size replay buffer, which will be used for the network update. The detailed training and update process is introduced in the next subsection. The objective of the training procedure is to identify the optimal payment strategy for the platform and the optimal data update strategy for each worker while adhering to AoI constraints, so that the expected discounted accumulated reward of each participant can be maximized within a finite time horizon.

5.2 Payment Strategy for the Platform

5.2.1 Markov Decision Process

In general, a higher unit-remuneration paid to workers decreases the platform's immediate utility while it may incentivize more workers to upload fresh data frequently. Apparently, the current payment strategy will affect the future data collection process and the future utility. Hence, the payment decision in the two-stage Stackelberg game can be formulated as a Markov Decision Process (MDP), and then the above TD3-based network framework can be applied. In the following, we need to specify the state space, the action space, and the reward function as below.

- State space: According to the Stackelberg game formalization, the state of the platform is built by the historical data update strategies of workers and the historical payment strategies. Assume that the platform records the information of historical τ time slots. The platform knows the data update frequency vectors of workers $\{P^{t-\tau}, \cdots, P^{t-1}\}$ and the unit-remuneration vectors $\{\mathcal{R}^{t-\tau}, \cdots, \mathcal{R}^{t-1}\}$ in the previous τ time slots. Thus, the observation space of the platform can be described as
$$\mathbf{s}^t = \{\mathcal{R}^{t-\tau}, P^{t-\tau}, \mathcal{R}^{t-\tau+1}, P^{t-\tau+1}, \cdots, \mathcal{R}^{t-1}, P^{t-1}\}. \quad (47)$$
- Action space: At the time slot t, the platform decides its action with respect to the current state. The action refers to the unit-remuneration vector $\mathcal{R}^t = \{R_1^t, R_2^t, \cdots, R_N^t\}$, where $R_i^t \in [R_{min}, R_{max}]$. Note that the platform determines the payment strategy in the current time slot t according to the actor network with random noise, i.e., the platform's action is defined as:
$$\mathcal{R}^t = \Pi_\theta(\mathbf{s}^t) + \xi, \quad (48)$$

where the noise ξ is a random vector, each element of which is independently sampled from the clipped normal distribution.

- Reward function: After executing an action $\mathcal{R}^t$ under a certain state $\mathbf{s}^t$, the platform will get an immediate reward $\Upsilon(\mathbf{s}^t, \mathcal{R}^t)$. According to the utility function, the reward function should be positively correlated with the net profit Φ. In Eqs. (5), (7), and (8), the objective of the platform is to maximize its own utility under the constraints on the AoI of data and the total data update frequency. Hence, when the constraints are violated, a penalty should be incorporated into the reward function. Based on these considerations, the reward function is designed as:

$$\Upsilon(\mathbf{s}^t, \mathcal{R}^t) = \varrho_1 \Phi(P^t, \mathcal{R}^t) - \varrho_2 \left[\frac{\sum_{i=1}^{N} p_i^t - \hat{p}}{\hat{p}}\right]^+ - \varrho_3 \sum_{i=1}^{N} \left[\frac{\delta_i^t - \varepsilon}{\varepsilon}\right]^+,$$

where $[x]^+ \triangleq \max(0, x)$. ϱ_1, ϱ_2, and ϱ_3 are three positive adjustable coefficients.

5.2.2 Network Training

In the TD3 architecture, the platform contains one actor network and two critic networks. The actor network aims at deriving a near-optimal unit-remuneration vector to maximize the long term accumulated reward. The state of the platform is taken as input, and the output is the action of the platform. The critic networks aim to derive an expectation of the accumulated reward from the current state as accurately as possible, in which using two target critic networks can efficiently prevent the overestimation problem of DDPG. Each critic network takes the state and action of the platform as the input and then outputs the Q-value.

In the time slot t, the platform first observes the current state $\mathbf{s}^t$. According to the state, the platform chooses a noisy action from a normal distribution with the expectation of $\Pi_\theta(\mathbf{s}^t)$, where the amount of added noise can be adjusted in terms of the exploration policy. After executing the action $\mathcal{R}^t$, the platform can acquire the corresponding reward $\Upsilon(\mathbf{s}^t, \mathcal{R}^t)$ from the environment, and then the state is transmitted to the next state $\mathbf{s}^{t+1}$. Afterwards, the platform obtains the tuple $\langle \mathbf{s}^t, \mathcal{R}^t, \Upsilon(\mathbf{s}^t, \mathcal{R}^t), \mathbf{s}^{t+1} \rangle$ and stores the experience tuple into the replay buffer for the following network update. It is worth mentioning that the experience selection can adopt a random strategy or prioritized replay strategy [52]. In this paper, we simply apply the random experience selection since it can break the correlation among the experiences. If the buffer is already full, the oldest experiences need to be thrown out of the memory to make room for the new one.

When congregating the sufficient experiments, the platform will take full advantage of this historical knowledge to train the networks of TD3. More specifically, the platform first samples a mini-batch of experiences with the size L from the replay buffer. Then, the target actor network predicts the noisy action with the new state, i.e., $\hat{\mathcal{R}}^{t+1} = \Pi_{\theta^-}(\mathbf{s}^{t+1}) + \xi'$, where ξ' is the Gaussian noise. Based on this, two target critic networks can also get the predictions. Consequently, the Temporal Difference (TD) target will be calculated according to the Bellman equations:

$$\hat{y}^t = \Upsilon(\mathbf{s}^t, \mathcal{R}^t) + \varpi \cdot \min_{i=1,2} \mathcal{V}_{w_i^-}(\mathbf{s}^{t+1}, \hat{\mathcal{R}}^{t+1}), \tag{49}$$

where ϖ is the discount factor determining the priority of the short-term reward. For two online critic networks (i.e., $\mathcal{V}_{w_1}$ and $\mathcal{V}_{w_2}$), their loss functions are defined as

$$\mathcal{L}(w_k) = \frac{1}{L} \sum\nolimits_{l=1}^{L} (\hat{y}^l - \mathcal{V}_{w_k}(\mathbf{s}^l, \mathcal{R}^l))^2, \forall k = 1, 2. \tag{50}$$

By minimizing the above loss functions, the online critic networks are updated based on the gradient descent method, i.e.,

$$w_1 \leftarrow w_1 - \kappa_1 \nabla_{w_1} \mathcal{L}(w_1),\ w_2 \leftarrow w_2 - \kappa_1 \nabla_{w_2} \mathcal{L}(w_2), \tag{51}$$

Algorithm 2: Payment Strategy Searching of DIM

```
Input: the learning rates $\kappa_1, \kappa_2$, the weighting factor $\kappa_3$, the
       discount factor $\varpi$, the size $L$, and the delay $\bar{\tau}$;
1  Initialization: the online actor network $\Pi_\theta$ and the online
     critic networks $\{\mathcal{V}_{w_1}, \mathcal{V}_{w_2}\}$; clean up the reply buffer;
2  Construct the target networks by copying the online networks:
     $\theta^- \xleftarrow{copy} \theta$, $w_1^- \xleftarrow{copy} w_1, w_2^- \xleftarrow{copy} w_2$;
3  for each time slot $t = 1, 2, \cdots, T$ do
4      Observe the state $\mathbf{s}^t$ from the environment;
5      Select the noisy action: $\mathcal{R}^t = \Pi_\theta(\mathbf{s}^t) + \xi$;
6      Acquire the reward $\Upsilon(\mathbf{s}^t, \mathcal{R}^t)$ and transit to $\mathbf{s}^{t+1}$;
7      Store $\langle \mathbf{s}^t, \mathcal{R}^t, \Upsilon(\mathbf{s}^t, \mathcal{R}^t), \mathbf{s}^{t+1} \rangle$ into the reply buffer;
8      Sample mini-batch of experiences from the reply buffer;
9      Predict: $\hat{\mathcal{R}}^{t+1} = \Pi_{\theta^-}(\mathbf{s}^{t+1}) + \xi'$;
10     Compute the TD-target $\hat{y}^t$ according to Eq. (49);
11     Update two online critic networks $\{\mathcal{V}_{w_1}, \mathcal{V}_{w_2}\}$:
         $w_1 \leftarrow w_1 - \kappa_1 \nabla_{w_1} \mathcal{L}(w_1), w_2 \leftarrow w_2 - \kappa_1 \nabla_{w_2} \mathcal{L}(w_2)$;
12     if $t \mod \bar{\tau} == 0$ then
13         Update $\Pi_\theta$: $\theta \leftarrow \theta + \kappa_2 \nabla_\theta \mathcal{G}(\theta)$;
14         Update $\Pi_{\theta^-}$: $\theta^- \leftarrow \kappa_3 \theta + (1 - \kappa_3) \theta^-$;
15         Update $\mathcal{V}_{w_1^-}$: $w_1^- \leftarrow \kappa_3 w_1 + (1 - \kappa_3) w_1^-$;
16         Update $\mathcal{V}_{w_2^-}$: $w_2^- \leftarrow \kappa_3 w_2 + (1 - \kappa_3) w_2^-$;
```

where κ_1 is a positive constant reflecting the learning rate. Inspired by the trick of delayed policy update, the update of the online actor network, the target actor network, and two target critic networks will be performed every $\bar{\tau}$ time slots. We introduce the learning objective of the online actor network, which is defined as:

$$\mathcal{G}(\theta) = \frac{1}{L} \sum\nolimits_{l=1}^{L} \mathcal{V}_{w_1}(\mathbf{s}^l, \Pi_\theta(\mathbf{s}^l)), \tag{52}$$

$$\Rightarrow \nabla_\theta \mathcal{G}(\theta) \approx L^{-1} \sum \nabla_{\mathcal{R}} \mathcal{V}_{w_1}(\mathbf{s}^t, \mathcal{R}^t) \mid_{\mathcal{R}^t = \Pi_\theta(\mathbf{s}^t)} \cdot \nabla_\theta \Pi_\theta(\mathbf{s}^t).$$

Since the platform aims to maximize $\mathcal{G}(\theta)$, the online actor network can be updated based on the gradient ascent method, i.e., $\theta \leftarrow \theta + \kappa_2 \nabla_\theta \mathcal{G}(\theta)$, where κ_2 is the learning rate of the network. Finally, we employ a soft-update strategy to update the target actor network and two target critic networks, i.e.,

$$\begin{cases} \theta^- \leftarrow \kappa_3 \theta + (1 - \kappa_3) \theta^-; \\ w_1^- \leftarrow \kappa_3 w_1 + (1 - \kappa_3) w_1^-; \\ w_2^- \leftarrow \kappa_3 w_2 + (1 - \kappa_3) w_2^-, \end{cases} \tag{53}$$

where $\kappa_3 \in (0, 1)$ is an adjustable weighting factor.

5.2.3 Algorithm Description

In Algorithm 2, we present the pseudocode of finding the optimal payment strategy for the platform. The TD3-based algorithm is mainly comprised of the strategy determination process (Steps 3-10) and the policy update process (Steps 11-16). More precisely, we first randomly initialize the parameters of the online actor network and two online critic networks (Step 1) and then prepare the reply buffer with a fixed size. At the beginning, the parameters of all target networks are equal to the online networks. For each iteration of the Stackelberg game at the time slot t, we observe the current state $\mathbf{s}^t$ and select an action $\mathcal{R}^t$ with the exploration noise ξ (Steps 4-5). It should be noted that the added noise is clipped to remain the noisy action close to the original action. Next, the platform will broadcast the unit-remuneration vector $\mathcal{R}^t$ to all workers and receive the collected data. According to the current data update frequencies of workers, the platform obtains an immediate reward $\Upsilon(\mathbf{s}^t, \mathcal{R}^t)$ and transits to the next

Algorithm 3: Data Update Strategy Searching of DIM

```
   Input: the learning rates $\kappa_{i,1}, \kappa_{i,2}$, the weighting factor $\kappa_{i,3}$,
          and the discount factor $\varpi_i$, for $\forall i \in \mathcal{N}$;
          the size $L$, the delay $\bar{\tau}$, and the total time slot $T$;
1  Initialization: the online actor networks $\{\Pi_{i,\theta}, \forall i \in \mathcal{N}\}$ and
     the online critic networks $\{\mathcal{V}_{i,w_1}, \mathcal{V}_{i,w_2}, \forall i \in \mathcal{N}\}$ ;
2  Each worker cleans up the reply buffer and constructs the target
     networks: $\theta^- \xleftarrow{copy} \theta$, $w_1^- \xleftarrow{copy} w_1$, $w_2^- \xleftarrow{copy} w_2$;
3  for each time slot $t = 1, 2, \cdots, T$ do
4      Update the environment;
5      for each worker $i \in \mathcal{N}$ in parallel do
6          Observe the state $\mathbf{s}_i^t$ from the environment;
7          Select the noisy action: $p_i^t = \Pi_{i,\theta}(\mathbf{s}_i^t) + \xi_i$;
8          Perform action $p_i^t$ and acquire the reward $\Upsilon_i(\mathbf{s}_i^t, p_i^t)$;
9          Store $\langle \mathbf{s}_i^t, p_i^t, \Upsilon_i(\mathbf{s}_i^t, p_i^t), \mathbf{s}_i^{t+1} \rangle$ into the reply buffer;
10         Sample a mini-batch of tuples from the reply buffer;
11         Compute the TD-target and the corresponding loss;
12         Update two online critic networks $\{\mathcal{V}_{i,w_1}, \mathcal{V}_{i,w_2}\}$;
13         if $t \mod \bar{\tau} == 0$ then
14             Update $\Pi_{i,\theta}, \Pi_{i,\theta^-}, \mathcal{V}_{i,w_1^-}, \mathcal{V}_{i,w_2^-}$;
```

state $\mathbf{s}^{t+1}$ (Step 6). Meanwhile, the platform stores the tuple $\langle \mathbf{s}^t, \mathcal{R}^t, \Upsilon(\mathbf{s}^t, \mathcal{R}^t), \mathbf{s}^{t+1} \rangle$ into the reply buffer, which will be applied for training networks (Steps 7-8). Aided by the target actor network, we predict the action at the time slot $t+1$ and compute the TD-target $\hat{y}^t$ according to Eq. (49). In steps 9-11, the platform updates two online critic networks $\{\mathcal{V}_{w_1}, \mathcal{V}_{w_2}\}$ by minimizing the corresponding loss functions $\{\mathcal{L}(w_1), \mathcal{L}(w_2)\}$ based on Eqs. (50) and (51). When there is $t \mod \bar{\tau} = 0$, the platform will update the online actor network and three target networks (Steps 12-16). Specifically, in order to maximize $\mathcal{G}(\theta)$, Π_θ will be updated by the deterministic policy gradient (Step 13). Besides, according to Eqs. (52) and (53), the parameters of three target networks (i.e., $\{\Pi_{\theta^-}, \mathcal{V}_{w_1^-}, \mathcal{V}_{w_2^-}\}$) will also be updated (Steps 14-16).

5.3 Date Update Strategy for Workers

Since each worker has no prior knowledge of the Stackelberg game, workers also cannot immediately determine their own optimal strategies. In particular, the decision-making procedure of each worker needs to take into account the strategies of other workers and the influence of the social network. Therefore, we adopt a multi-agent TD3 approach to learn the optimal data update strategies to maximize workers' utilities, in which each worker becomes an agent to make a decision (i.e., the data update frequency). For each worker i, the online networks are denoted by $\{\Pi_{i,\theta}, \mathcal{V}_{i,w_1}$, and $\mathcal{V}_{i,w_2}\}$, and the corresponding target networks are represented by $\{\Pi_{i,\theta^-}, \mathcal{V}_{i,w_1^-}$, and $\mathcal{V}_{i,w_2^-}\}$. First, we introduce the state space, the action space, and the reward function.

- State space: Each worker i interacts with the environment to attain the auxiliary information, including the strategies of other workers and the unit-remunerations of the platform in the previous τ time slots, as well as the payment strategy in the current time slot t. Hence, the state space of worker i is
$$\mathbf{s}_i^t = \{P_{-i}^{t-\tau}, \mathcal{R}^{t-\tau}, P_{-i}^{t-\tau+1}, \mathcal{R}^{t-\tau+1}, \cdots, P_{-i}^{t-1}, \mathcal{R}^{t-1}, \mathcal{R}^t\}.$$
- Action space: At the time slot t, worker i decides its action under the current state, where the action refers to the data update frequency $p_i^t \in [p_{min}, p_{max}]$. Similarly to the platform, worker i chooses a noisy action with the Gaussian noise ξ_i, i.e.,
$$p_i^t = \Pi_{i,\theta}(\mathbf{s}_i^t) + \xi_i. \tag{54}$$
- Reward function: After all workers execute their strategies, worker i can observe the actions of others. Since each worker hopes to improve its utility, the immediate reward can be directly calculated by $\Upsilon_i(\mathbf{s}_i^t, p_i^t) = \Omega_i$ according to Eq. (3).

TABLE 2
Simulation settings

Parameter name	Values
number of workers N	[10, 300] (100 in default)
quadratic parameter d for platform	[5, 10] (5 in default)
conversion parameter η for platform	[6, 11] (10 in default)
quadratic parameter a for worker	[5, 15] (5 in default)
conversion parameter s for worker	[6, 16] (6 in default)
total data update frequency $\hat{p}$	[70, 100] (100 in default)
total time slot T and iteration number	[0, 400]; 400

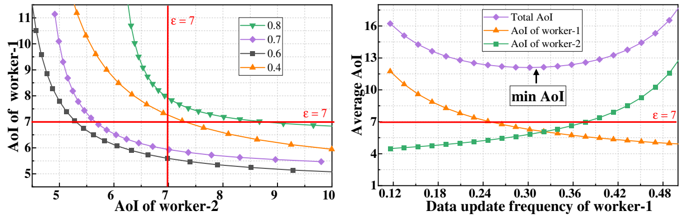
Algorithm 3 summarizes the process of seeking the optimal data update strategies for workers. Since the workers are unwilling to share their state inputs, we design a decentralized TD3-based approach for each worker. Because the DRL network of each worker is similar to that of the platform, we only present the key steps, and some update expressions can be obtained by referring to Section 5.2.2. More precisely, all workers first set the related parameters and initialize the original network parameters. Meanwhile, the target networks can be easily created by copying the online networks (Steps 1-2). Then, each worker observes the environment to get the current state $\mathbf{s}_i^t$ (Steps 4-6). Under the state, the worker determines its data update frequency p_i^t based on the online actor network and the random Gaussian noise (Step 7). After all workers perform their own current actions, each worker can procure the reward $\Upsilon_i(\mathbf{s}_i^t, p_i^t)$ and the state $\mathbf{s}_i^t$ will be changed to $\mathbf{s}_i^{t+1}$ (Step 8). Next, each worker stores the transition $\langle \mathbf{s}_i^t, p_i^t, \Upsilon_i(\mathbf{s}_i^t, p_i^t), \mathbf{s}_i^{t+1} \rangle$ into its reply buffer. When the worker needs to update the network parameters, it will sample some transitions from the replay buffer (Steps 9-10). In steps 11-12, worker i computes the TD-targets and then updates two online critic networks by minimizing the corresponding loss functions. When the current time slot is a multiple of the preset constant $\bar{\tau}$, the parameters of four networks $\{\Pi_{i,\theta}, \Pi_{i,\theta^-}, \mathcal{V}_{i,w_1^-}, \mathcal{V}_{i,w_2^-}\}$ will be updated simultaneously (Steps 13-14).

6 Performance Evaluations

In this section, we evaluate the performance of two proposed mechanisms (i.e., AIM and DIM) with extensive simulations of real-world data traces. We first describe the simulation settings and introduce the compared algorithms, and then the experimental results are presented in great detail.

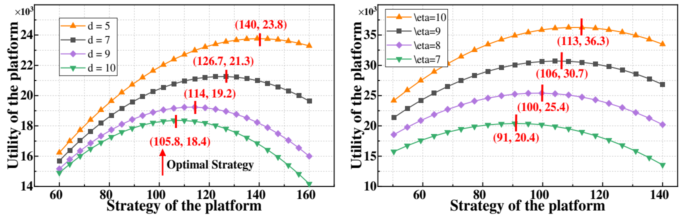
6.1 Evaluation Methodology

Simulation Settings: We perform our simulations on the real-world data of Chicago Taxi Trips [53]. Each trace records the taxi ID, timestamp, trip seconds, trip miles, pickup/dropoff areas, etc. We select a data set of 27055 taxi records. In our simulations, we select some taxi drivers as MCS workers and treat the taxi-hailing requests as sensing tasks. First, we choose 15 PoIs and find 300 taxis from the trace. Then, we choose N taxis as workers, where N ranges from $[10, 300]$. We also simulate the social network based on a real data trace from SNAP (Gowalla) [54], which is



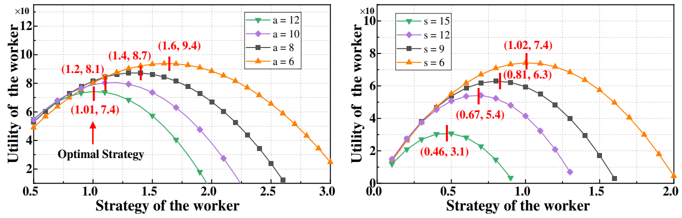
(a) worker-1's AoI vs. worker-2's AoI (b) total AoI and each worker's AoI

Fig. 4. AoI of two competitive workers



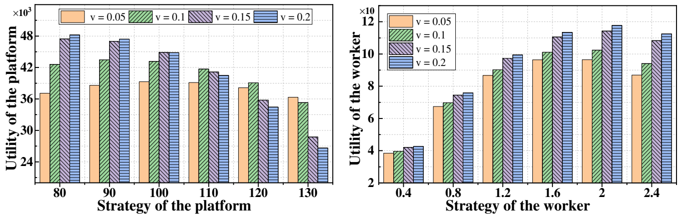
(a) quadratic parameter d (b) conversion parameter η

Fig. 5. Strategy of the platform vs. Utility



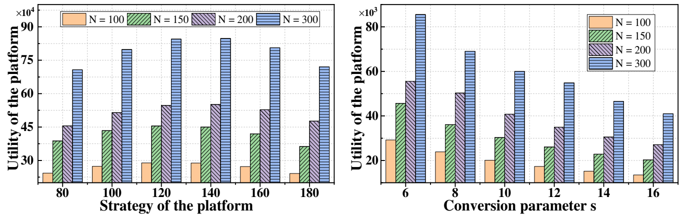
(a) quadratic parameter a (b) conversion parameter s

Fig. 6. Strategy of a worker vs. Utility



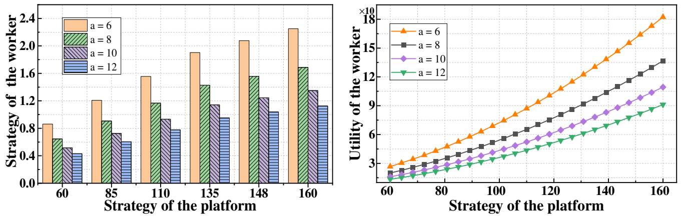
(a) utility of the platform (b) utility of the worker

Fig. 7. Influence of social network effects



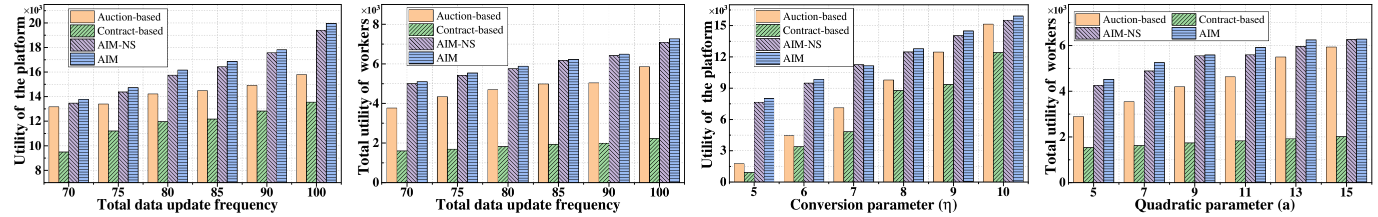
(a) PU under different N (b) PU under different N and s

Fig. 8. Influence of the number of workers



(a) strategy of the worker (b) utility of the worker

Fig. 9. Influence of the strategy of the platform



(a) PU vs. $\hat{p}$ (b) total WU vs. $\hat{p}$ (c) PU vs. η (d) total SU vs. a

Fig. 10. PU and total WU under different incentive mechanisms and varied parameters

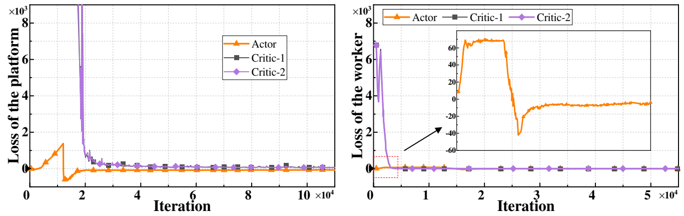
a location-based social friendship network built by mobile phone users. Next, we randomly pick N nodes from the network, and the social network effect coefficient v is produced from $[0.01, 0.2]$. The conversion parameters s and η change from $[6, 16]$ and $[6, 11]$, respectively. Meanwhile, the quadratic parameters a and d are set in the range $[5, 15]$ and $[5, 10]$, respectively. Additionally, the default values are $a = 5, b = 1, c = 40, d = 5, s = 6$, and $\eta = 10$. During the training process, we set the batch size and the learning rate as 256 and 3×10^{-4}, respectively. The standard deviation of Gaussian noise, the discount factor ϖ, the delay $\bar{\tau}$, and the size of the replay buffer are respectively initialized as 0.1, 0.99, 2, and 4×10^3. The maximum number of time slots is 400, and the iteration number of each time slot is 400. Moreover, the parameters used in our simulations are shown in Table 2.

Compared Algorithms: Since AIM combines the Stackelberg game and AoI to keep data fresh and solve the incentive problem with incomplete information, we compare AIM with some existing state-of-the-art studies with incentive mechanism designs [12], [29]. However, the models and problems in these works are different from ours, so we cannot compare them directly. Thus, we tailor the basic idea in these algorithms and carefully design three incentive mechanisms for comparison: Auction-based algorithm [29], Contract-based algorithm [12], and AIM-NS. Here, the auction-based scheme utilizes the technique of game theory, the contract-based algorithm is based on contract theory, and the AIM-NS mechanism means that we do not consider social network effects. Moreover, we compare DIM with the random approach [55], where the platform randomly selects its own unit-remuneration with a given payment range in each time slot.

6.2 Evaluation Results

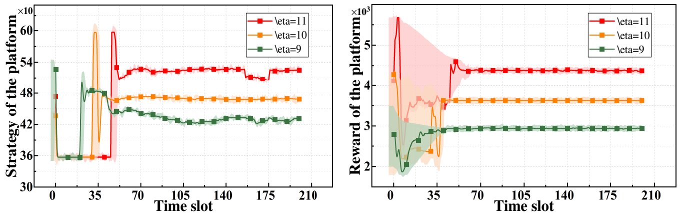
For the evaluation, we use the following main metrics: AoI, strategy, and utility. To be more precise, we use PU, WU, PS, and WS to denote the platform's utility, a worker's utility, the platform's strategy, and the worker's strategy, respectively.

1) *Evaluation of AoI:* We measure the AoI values for the $M/M/1$ FCFS queue system with $\mu = 1$. For simplicity, we set that there are only two workers competing for the data update with a fixed total load $\rho_1 + \rho_2 = \hat{\rho}$ and $\beta_1 = \beta_2$. From Fig. 4(a), we can see that the average AoI value of worker-1 decreases with



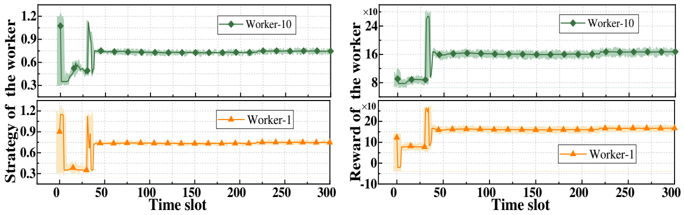
(a) loss of the platform (b) loss of the worker

Fig. 11. Training loss of the platform and the worker



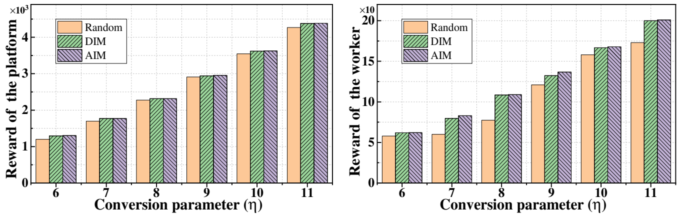
(a) strategy of the platform (b) reward of the platform

Fig. 12. Strategy/Reward convergence of the platform



(a) strategy of the worker (b) reward of the worker

Fig. 13. Strategy/Reward convergence of workers



(a) reward of the platform (b) reward of the worker

Fig. 14. Reward comparison under different parameters

the increase of worker-2's AoI value. If we set the threshold as $\varepsilon = 7$, workers can meet the AoI constraint only when $\hat{\rho} = 0.6$ or 0.7. As illustrated in Fig. 4(b), the sum of AoI decreases first and increases later, and the total AoI value can reach a minimum value. The general result for such systems is that the multi-user AoI optimization problem depends on both the total load $\hat{\rho}$ and the allocation of data update frequency among workers.

2) *Evaluation of AIM:* We first verify the existence of the Stackelberg equilibrium for the platform. As depicted in Fig. 5, we evaluate PU under different parameters (i.e., the quadratic parameter d and conversion parameter η) when the platform changes its strategy in the range [60, 160]. We can see that PU will always find a maximum point, which demonstrates that the platform can find its optimal payment strategy to maximize its utility. Besides, using a small d or a larger η will result in the growth of the optimal PS and the optimal PU. This is consistent with the theoretical results of the Stackelberg equilibrium. Similarly, we display the evaluation results of the Bayesian game of workers. As shown in Fig. 6, we randomly select a worker and then change the strategy of a worker from 0.1 to 3. Under different parameters (i.e., a and s), we evaluate the worker's utility and find the maximum point. Therefore, each worker can also seek out its optimal data update frequency and acquire the maximum utility. In addition, the worker's utility will increase when applying a smaller a or s since the cost of the worker becomes smaller. The SE existence of each participant can ensure the stability of the whole system.

Then, we evaluate the impact of the social network effects in terms of the utilities of the platform and workers, as depicted in Fig. 7. When we enlarge the social network effect coefficient υ from 0.05 to 0.2, both the worker and the platform can possess a higher utility. On the one hand, the platform and all workers can find their optimal strategies under different coefficients. On the other hand, the optimal strategies of workers become higher along with the increase of the coefficient υ. This is because workers can obtain more social benefits from their neighbors and are willing to collect data with a high frequency.

Next, we investigate the effect of the number of workers by changing the strategy of the platform from 80 to 180 and adjusting the conversion parameter s in the range $[6, 16]$. According to Fig. 8(a), we observe that the platform can also determine its optimal strategy under diverse number of workers N, which is consistent with Fig. 5. Clearly, more workers who are willing to engage in the MCS system can improve the utility of the platform, which reflects the significance of incentive mechanism designs. Moreover, Fig. 8(b) shows that applying a lower conversion parameter can also make the platform earn more profits. As presented in Fig. 9, we change the strategy of the platform in the range [60, 160] and then observe the influence of PS on any worker's strategy and utility. When the platform invests more money to incentivize workers, each worker will upload data as frequently as possible so as to acquire more remunerations. In addition, a smaller a will result in a high WU, which is also reflected in Fig. 6(a). The reason is that the worker will expend less effort to update data.

Finally, Fig. 10 investigates the utilities of the platform and all workers under different incentive mechanisms, where these mechanisms are based on the contract theory, auction theory, and game theory. In Figs. 10(a) and 10(b), we evaluate the impact of total data update frequency $\hat{\rho}$ from 70 to 100 with N=100. Along with the growth of $\hat{\rho}$, the utility of the platform and the total utility of workers will increase since more workers have chances to participate in the MCS system. More importantly, the PU and the total WU of compared algorithms are lower than those of the mechanism AIM. Especially, when $\hat{\rho} = 100$, the achieved PU of AIM is about 47.3% and 26.4% higher than those of the contract-based and auction-based algorithms on average, respectively. This is because the contract-based algorithm and the auction-based algorithm only guarantee the non-negativity of workers' utilities and cannot achieve utility maximization. Meanwhile, the total worker utility of the contract-based algorithm exhibits slower growth. This can be attributed to the limited number of designed contracts, which prevents the optimization of all workers' utilities compared to AIM. As shown in Figs. 10(c) and 10(d), we present the superiority of AIM by altering the conversion parameter η and the quadratic parameter a. In addition to the obtained results from Figs. 10(a) and 10(b), it is worth noting that AIM performs better than AIM-NS, indicating that social network effects can bring additional benefits to the MCS system.

3) *Evaluation of DIM:* We describe the convergence results for our extended mechanism (i.e., DIM). The actor network and critic network in DIM are both two-layer fully connected neural

networks, and the Relu function is utilized to activate these networks. For the sake of the computing power, we select ten workers to learn their optimal strategies in parallel, and the platform pays the same unit-remuneration to all workers. Fig. 11 validates the satisfactory convergence of DIM by recording the training loss of each iteration. From Fig. 11(a), we find that both the actor network and two critic networks of the platform tend to a stable state as the training time increases. Since the changing trend of training loss for each worker is similar, we only plot the curves of one worker for presentation, as depicted in Fig. 11(b). In the initial iterations, the training loss of the worker will drop off at an astonishing speed. Along with the increasing iteration times, the training loss will also remain stable and cannot go down.

Then, we study the strategy convergence and utility convergence of the platform, as reflected in Figs. 12(a) and 12(b), respectively. When the conversion parameter η is 10, we set the action space of the platform as [350, 600]. From the evaluation results, it is worth noting that the fluctuating range of curves is higher when the time slot is less than 70, especially for the reward of the platform. This is because the platform needs to explore more payment strategies in the initial phase. After learning enough knowledge from the environment and multiple interactions, the platform is inclined to pick the payment strategy that can maximize its own utility. Hence, the platform's strategy and reward will be close to the optimal value. In addition, we change the conversion parameter η from 9 to 11 and evaluate the convergence performance. A higher η needs more time slots to determine the optimal strategy and can increase the reward of the platform, which is in accordance with the results in Fig. 5(b).

Next, we analyze the strategy convergence and utility convergence of workers. As shown in Fig. 13, we take worker-1 and worker-10 as an example, where their social relationships and utility functions are the same. For the convenience of exploration, the action space of each worker is specified as [0.3, 1.2]. According to Figs. 13(a) and 13(b), we see that the strategies and rewards of worker-1 and worker-10 will converge to the nearly optimal solution. Since the strategies of followers (i.e., workers) depend on the strategy of the leader (i.e., the platform), the rewards of two workers will have dramatic changes in the initial time slots. Meanwhile, although the learning processes of the two workers are dissimilar, they will ultimately find out the near-identical strategy to acquire their own maximum utilities.

Finally, we measure the rewards of the platform and worker-1 with respect to the conversion parameter η under different algorithms, as illustrated in Fig. 14. Here, AIM can be regarded as the theoretically optimal solution according to Theorem 3. From Figs. 14(a) and 14(b), when we change the conversion parameter η from 6 to 11, DIM is obviously more stable than the heuristic method. In addition, the platform's reward and the worker's reward are higher than those of the random approach. Especially, when the conversion parameter is 11, the reward of worker-1 is about 15.6% higher than that of the random approach on average. More significantly, there is only a slight gap between DIM and AIM, which demonstrates the great learning performance of DIM. That is, even though the platform and workers do not have any prior information, they can still customize their near-optimal strategies and attain the nearly maximum utilities by trial and error.

7 Related Works

In our study, we mainly review related works from two categories: incentive mechanism and age of information in MCS systems.

Incentive Mechanism: Many remarkable incentive mechanisms have been designed for various MCS systems [12], [29]–[33], [56], [57]. Diverse tools have been leveraged in previous studies, such as game theory [32], [33], [57], auction mechanism [29]–[31], [58], contract theory [12], DRL [35]–[37], [59], and so on. For example, Jin *et al.* [30] developed a reverse auction-based incentive mechanism, which can pick out reliable workers and compensate workers' costs (including sensing cost and privacy leakage risk). Sun *et al.* [12] proposed a personalized privacy-preserving incentive mechanism based on the contract theory. Liu *et al.* [57] designed a three-stage game-theoretical model to motivate the miners' participation in blockchain systems. Nevertheless, these methods generally require some personal information of the participants, which is impractical in some real applications. Fortunately, DRL has been serving as an efficient solution to address a variety of complex games under the uncertain and stochastic environment. For example, the work in [33] formulated a multi-leader multi-follower Stackelberg game as the Markov decision process and proposed a dynamic incentive mechanism based on the proximal policy optimization method. Li *et al.* [35] presented a DRL-based solution to the Stackelberg game and applied a multi-agent DDPG algorithm to train networks. However, many of them did not take the social network and freshness concerns into account, so they cannot be applied to our system.

A handful of studies integrate the social network effects into incentive mechanism designs [38], [43], [44], [60]–[63]. For instance, Cheung *et al.* [43] exploited the users' social relationships and effort levels to design an incentive mechanism, from which a closed-form solution is derived and the system can achieve a win-win situation. The authors in [44] took the strategic connections of service providers and the social influence of users into consideration and proposed the game-based incentive mechanisms with complete information. Xu *et al.* [63] considered the task diffusion in mobile social networks and proposed a budget-feasible incentive mechanism to increase the number of participants. The authors in [38] designed a secure edge caching scheme for mobile users and the content provider in mobile social networks and employed the Q-learning to find the optimal payment policy in dynamic network scenarios. Nevertheless, most of these researches ignore the importance of data freshness, especially for time-sensitive MCS applications.

Age of Information: AoI is a widely-used application layer metric for information freshness performance, which is suitable to evaluate the timeliness of data collection. There have been plenty of works that focus on addressing various AoI optimization problems [39]–[42], [64]. For example, Dai *et al.* [39] introduced a model-based DRL framework including a Monte Carlo tree search structure, which can maximize the geographical coverage of unmanned aerial vehicles and collected data while minimizing all mobile users' AoI values. Yang *et al.* [64] focused on the scheme design of the AoI optimization using AI-empowered diagnostic bots and the mixed game so that the AoI value is lower during the biosensing data transmission. Only a few researchers have studied the AoI demand with the pricing issue [21]–[24]. For instance, the work in [24] devised a long-term decomposition mechanism in a status acquisition system, where the freshness of the data can still be guaranteed under time-varying information asymmetry. The authors in [21] leveraged the AoI metric to model the waiting time cost of the requester and designed the dynamic task pricing strategy to achieve profit maximization for MCS systems. Nevertheless, none of the existing works take the

AoI constraint and workers' social benefits into account together, which involves a complex incomplete information game due to the uncertainty of social network effects. Moreover, our work also takes into consideration the Stackelberg game with unknown parameters, i.e., the prior information of the game is absent, which will be more in accordance with practical applications.

8 Conclusion

In this article, we study the issue of the incentive mechanism design with freshness concerns and social benefits for MCS systems. We first model the problem as a two-stage Stackelberg game, embedded with an incomplete information Bayesian subgame. Next, we propose two AoI-guaranteed incentive mechanisms: AIM and DIM. AIM derives the optimal strategies for the platform and workers that are proven to form a unique Stackelberg equilibrium. It is applicable to the scenario that the parameters of the Stackelberg game are public and can maximize the utilities of the platform and workers simultaneously. Furthermore, DIM is an extended mechanism with the aid of the powerful DRL technique. It is suitable to the general scenario that each participant has no prior knowledge of the Stackelberg game and can quickly learn the optimal strategy for each participant without requiring any prior information. Moreover, the system can ensure that the AoI values of all data are not larger than a given threshold. At last, extensive experiments on real-world traces are supplemented to validate the great performance of our proposed mechanisms.

References

[1] M. Xiao, Y. Xu, J. Zhou, J. Wu, S. Zhang, , and J. Zheng, "Aoi-aware incentive mechanism for mobile crowdsensing using stackelberg game," in *IEEE INFOCOM*, 2023, pp. 1–10.

[2] X. Zhang, Z. Yang, W. Sun, Y. Liu, S. Tang, K. Xing, and X. Mao, "Incentives for mobile crowd sensing: A survey," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 1, pp. 54–67, 2016.

[3] 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.

[4] B. Guo, H. Chen, Y. Liu, C. Chen, Q. Han, and Z. Yu, "From crowdsourcing to crowdmining: using implicit human intelligence for better understanding of crowdsourced data," *World Wide Web*, vol. 23, no. 2, pp. 1101–1125, 2020.

[5] W. Liu, Y. Yang, E. Wang, H. Wang, Z. Wang, and J. Wu, "Dynamic online user recruitment with (non-) submodular utility in mobile crowdsensing," *IEEE/ACM Transactions on Networking*, vol. 29, no. 5, pp. 2156–2169, 2021.

[6] J. Sun, H. Jin, Z. Yang, L. Su, and X. Wang, "Optimizing long-term efficiency and fairness in ride-hailing via joint order dispatching and driver repositioning," in *ACM SIGKDD*, 2022, pp. 3950–3960.

[7] B. Guo, J. Liu, S. Liu, J. Wang, M. Li, C. Wang, and Z. Yu, "Crowdim: Crowd-inspired intelligent manufacturing space design," *IEEE Internet of Things*, vol. 9, no. 19, pp. 19 387–19 397, 2022.

[8] Q. Ma, J. Huang, T. Basar, J. Liu, and X. Chen, "Reputation and pricing dynamics in online markets," *IEEE/ACM Transactions on Networking*, vol. 29, no. 4, pp. 1745–1759, 2021.

[9] C. H. Liu, Z. Dai, H. Yang, and J. Tang, "Multi-task-oriented vehicular crowdsensing: A deep learning approach," in *IEEE INFOCOM*, 2020, pp. 1123–1132.

[10] 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, 2018.

[11] G. Gao, M. Xiao, J. Wu, H. Huang, S. Wang, and G. Chen, "Auction-based VM allocation for deadline-sensitive tasks in distributed edge cloud," *IEEE Transactions on Services Computing*, vol. 14, no. 6, pp. 1702–1716, 2021.

[12] P. Sun, Z. Wang, L. Wu, Y. Feng, X. Pang, H. Qi, and Z. Wang, "Towards personalized privacy-preserving incentive for truth discovery in mobile crowdsensing systems," *IEEE Transactions on Mobile Computing*, vol. 21, no. 1, pp. 352–365, 2022.

[13] Y. Duan and J. Wu, "Spatial-temporal inventory rebalancing for bike sharing systems with worker recruitment," *IEEE Transactions on Mobile Computing*, vol. 21, no. 3, pp. 1081–1095, 2022.

[14] J. Xu, Z. Rao, L. Xu, D. Yang, and T. Li, "Incentive mechanism for multiple cooperative tasks with compatible users in mobile crowd sensing via online communities," *IEEE Transactions on Mobile Computing*, vol. 19, no. 7, pp. 1618–1633, 2020.

[15] Y. Li, H. Xiao, Z. Qin, C. Miao, L. Su, J. Gao, K. Ren, and B. Ding, "Towards differentially private truth discovery for crowd sensing systems," in *IEEE ICDCS*, 2020, pp. 1156–1166.

[16] M. Huai, D. Wang, C. Miao, J. Xu, and A. Zhang, "Privacy-aware synthesizing for crowdsourced data," in *IJCAI*, 2019, pp. 2542–2548.

[17] H. Wang, E. Wang, Y. Yang, J. Wu, and F. Dressler, "Privacy-preserving online task assignment in spatial crowdsourcing: A graph-based approach," in *IEEE INFOCOM*, 2022, pp. 570–579.

[18] Y. Qian, Y. Ma, J. Chen, D. Wu, D. Tian, and K. Hwang, "Optimal location privacy preserving and service quality guaranteed task allocation in vehicle-based crowdsensing networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 7, pp. 4367–4375, 2021.

[19] X. Zhang, Z. Yang, Y. Liu, and S. Tang, "On reliable task assignment for spatial crowdsourcing," *IEEE Transactions on Emerging Topics in Computing*, vol. 7, no. 1, pp. 174–186, 2019.

[20] E. Wang, M. Zhang, Y. Xu, H. Xiong, and Y. Yang, "Spatiotemporal fracture data inference in sparse urban crowdsensing," in *IEEE INFOCOM*, 2022, pp. 1499–1508.

[21] H. Gao, H. Xu, C. Zhou, H. Zhai, C. Liu, M. Li, and Z. Han, "Dynamic task pricing in mobile crowd sensing: an age of information-based queueing game scheme," *IEEE Internet of Things Journal*, vol. 9, no. 21, pp. 21 278–21 291, 2022.

[22] L. Wu, Y. Xiong, K.-Z. Liu, and J. She, "A real-time pricing mechanism considering data freshness based on non-cooperative game in crowdsensing," *Information Sciences*, vol. 608, pp. 392–409, 2022.

[23] B. Li and J. Liu, "Achieving information freshness with selfish and rational users in mobile crowd-learning," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 5, pp. 1266–1276, 2021.

[24] Z. Wang, L. Gao, and J. Huang, "Taming time-varying information asymmetry in fresh status acquisition," in *IEEE INFOCOM*, 2021, pp. 1–10.

[25] X. Gong, L. Duan, X. Chen, and J. Zhang, "When social network effect meets congestion effect in wireless networks: Data usage equilibrium and optimal pricing," *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 2, pp. 449–462, 2017.

[26] O. Candogan, K. Bimpikis, and A. E. Ozdaglar, "Optimal pricing in networks with externalities," *Operations Research*, vol. 60, no. 4, pp. 883–905, 2012.

[27] I. P. Fainmesser and A. Galeotti, "Pricing network effects," *The Review of Economic Studies*, vol. 83, no. 1, pp. 165–198, 2016.

[28] M. Belhaj and F. Deroïan, "The value of network information: Assortative mixing makes the difference," *Games and Economic Behavior*, vol. 126, pp. 428–442, 2021.

[29] X. Chen, L. Zhang, Y. Pang, B. Lin, and Y. Fang, "Timeliness-aware incentive mechanism for vehicular crowdsourcing in smart cities," *IEEE Transactions on Mobile Computing*, vol. 21, no. 9, pp. 3373–3387, 2022.

[30] H. Jin, L. Su, H. Xiao, and K. Nahrstedt, "Incentive mechanism for privacy-aware data aggregation in mobile crowd sensing systems," *IEEE/ACM Transactions on Networking*, vol. 26, no. 5, pp. 2019–2032, 2018.

[31] J. Xu, G. Chen, Y. Zhou, Z. Rao, D. Yang, and C. Xie, "Incentive mechanisms for large-scale crowdsourcing task diffusion based on social influence," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 4, pp. 3731–3745, 2021.

[32] Y. Xu, M. Xiao, J. Wu, S. Zhang, and G. Gao, "Incentive mechanism for spatial crowdsourcing with unknown social-aware workers: A three-stage stackelberg game approach," *IEEE Transactions on Mobile Computing*, p. Forthcoming, 2022.

[33] Y. Zhan, C. H. Liu, Y. Zhao, J. Zhang, and J. Tang, "Free market of multi-leader multi-follower mobile crowdsensing: An incentive mechanism design by deep reinforcement learning," *IEEE Transactions on Mobile Computing*, vol. 19, no. 10, pp. 2316–2329, 2020.

[34] M. Xiao, W. Jin, C. Li, and M. Li, "Eliciting joint truthful answers and profiles from strategic workers in mobile crowdsourcing systems," *IEEE Transactions on Mobile Computing*, p. Forthcoming, 2022.

[35] B. Li, K. Xie, X. Huang, Y. Wu, and S. Xie, "Deep reinforcement learning based incentive mechanism design for platoon autonomous driving with social effect," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 7, pp. 7719–7729, 2022.

[36] Y. Liu, H. Wang, M. Peng, J. Guan, and Y. Wang, “An incentive mechanism for privacy-preserving crowdsensing via deep reinforcement learning,” *IEEE Internet of Things Journal*, vol. 8, no. 10, pp. 8616–8631, 2021.

[37] Z. Hu, Y. Liang, J. Zhang, Z. Li, and Y. Liu, “Inference aided reinforcement learning for incentive mechanism design in crowdsourcing,” in *NeurIPS*, 2018, pp. 5512–5522.

[38] Q. Xu, Z. Su, and R. Lu, “Game theory and reinforcement learning based secure edge caching in mobile social networks,” *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 3415–3429, 2020.

[39] 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, pp. 1029–1038.

[40] S. K. Kaul, R. D. Yates, and M. Gruteser, “Real-time status: How often should one update?” in *IEEE INFOCOM*, 2012, pp. 2731–2735.

[41] Z. Qin, H. Wang, Z. Wei, Y. Qu, F. Xiong, H. Dai, and T. Wu, “Task selection and scheduling in uav-enabled MEC for reconnaissance with time-varying priorities,” *IEEE Internet of Things Journal*, vol. 8, no. 24, pp. 17 290–17 307, 2021.

[42] Z. Ning, P. Dong, X. Wang, X. Hu, L. Guo, B. Hu, Y. Guo, T. Qiu, and R. Y. Kwok, “Mobile edge computing enabled 5g health monitoring for internet of medical things: A decentralized game theoretic approach,” *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 2, pp. 463–478, 2021.

[43] M. H. Cheung, F. Hou, and J. Huang, “Make a difference: Diversity-driven social mobile crowdsensing,” in *IEEE INFOCOM*, 2017, pp. 1–9.

[44] J. Nie, J. Luo, Z. Xiong, D. Niyato, P. Wang, and H. V. Poor, “A multi-leader multi-follower game-based analysis for incentive mechanisms in socially-aware mobile crowdsensing,” *IEEE Transactions on Wireless Communications*, vol. 20, no. 3, pp. 1457–1471, 2021.

[45] R. D. Yates and S. K. Kaul, “The age of information: Real-time status updating by multiple sources,” *IEEE Transactions on Information Theory*, vol. 65, no. 3, pp. 1807–1827, 2019.

[46] M. Newman, *Networks*. Oxford university press, 2018.

[47] I. P. Fainmesser and A. Galeotti, “Pricing network effects: Competition,” *American Economic Journal: Microeconomics*, vol. 12, no. 3, pp. 1–32, 2020.

[48] J. Nie, J. Luo, Z. Xiong, D. Niyato, and P. Wang, “A stackelberg game approach toward socially-aware incentive mechanisms for mobile crowdsensing,” *IEEE Transactions on Wireless Communications*, vol. 18, no. 1, pp. 724–738, 2019.

[49] S. Athey, “Single crossing properties and the existence of pure strategy equilibria in games of incomplete information,” *Econometrica*, vol. 69, no. 4, pp. 861–889, 2001.

[50] E. L. Glaeser and J. A. Scheinkman, “Non-market interactions,” *Advances in Economics and Econometrics: Theory and Applications*, pp. 339–370, 2003.

[51] S. Fujimoto, H. van Hoof, and D. Meger, “Addressing function approximation error in actor-critic methods,” in *ICML*, vol. 80, 2018, pp. 1582–1591.

[52] Y. Oh, J. Shin, E. Yang, and S. J. Hwang, “Model-augmented prioritized experience replay,” in *ICLR*, 2022.

[53] “Chicago taxi trips,” [Online] Downloaded from https://www.kaggle.com/chicago/chicago-taxi-trips-bq, 2018.

[54] J. Leskovec and A. Krevl, “SNAP Datasets: Gowalla: Stanford large network dataset collection,” Downloaded from http://snap.stanford.edu/data/loc-Gowalla.html, 2014.

[55] Y. Zhao and C. H. Liu, “Social-aware incentive mechanism for vehicular crowdsensing by deep reinforcement learning,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 4, pp. 2314–2325, 2021.

[56] Z. Wang, J. Li, J. Hu, J. Ren, Q. Wang, Z. Li, and Y. Li, “Towards privacy-driven truthful incentives for mobile crowdsensing under untrusted platform,” *IEEE Transactions on Mobile Computing*, vol. 22, no. 2, pp. 1198–1212, 2023.

[57] Y. Liu, Z. Fang, M. H. Cheung, W. Cai, and J. Huang, “An incentive mechanism for sustainable blockchain storage,” *IEEE/ACM Transactions on Networking*, vol. 30, no. 5, pp. 2131–2144, 2022.

[58] M. Xiao, B. An, J. Wang, G. Gao, S. Zhang, and J. Wu, “Cmab-based reverse auction for unknown worker recruitment in mobile crowdsensing,” *IEEE Transactions on Mobile Computing*, vol. 21, no. 10, pp. 3502–3518, 2022.

[59] Y. Song and H. Jin, “Minimizing entropy for crowdsourcing with combinatorial multi-armed bandit,” in *IEEE INFOCOM*, 2021, pp. 1–10.

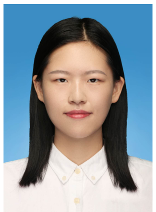
[60] J. Xu, Z. Luo, C. Guan, D. Yang, L. Liu, and Y. Zhang, “Hiring a team from social network: Incentive mechanism design for two-tiered social mobile crowdsourcing,” *IEEE Transactions on Mobile Computing*, p. Forthcoming, 2022.

[61] Z. Wang, Y. Huang, X. Wang, J. Ren, Q. Wang, and L. Wu, “Socialrecruiter: Dynamic incentive mechanism for mobile crowdsourcing worker recruitment with social networks,” *IEEE Transactions on Mobile Computing*, vol. 20, no. 5, pp. 2055–2066, 2021.

[62] R. Wang, F. Zeng, L. Yao, and J. Wu, “Game-theoretic algorithm designs and analysis for interactions among contributors in mobile crowdsourcing with word of mouth,” *IEEE Internet Things Journal*, vol. 7, no. 9, pp. 8271–8286, 2020.

[63] J. Xu, Y. Zhou, G. Chen, Y. Ding, D. Yang, and L. Liu, “Topic-aware incentive mechanism for task diffusion in mobile crowdsourcing through social network,” *ACM Transactions on Internet Technology (TOIT)*, vol. 22, no. 1, pp. 1–23, 2021.

[64] Y. Yang, W. Wang, Z. Yin, R. Xu, X. Zhou, N. Kumar, M. Alazab, and T. R. Gadekallu, “Mixed game-based aoi optimization for combating COVID-19 with AI bots,” *IEEE Journal on Selected Areas in Communications*, vol. 40, no. 11, pp. 3122–3138, 2022.



Yin Xu received her B.S. degree from the School of Computer Science and Technology at the Anhui University (AHU), Hefei, China, in 2019. She is currently a PhD student in the School of Computer Science and Technology at the University of Science and Technology of China (USTC), Hefei, China. Her research interests include mobile crowdsensing, federated learning, privacy preservation, game theory, edge computing, and incentive mechanism design.

Mingjun Xiao is a professor in the School of Computer Science and Technology at the University of Science and Technology of China (USTC). He received his Ph.D. from USTC in 2004. His research interests include mobile crowdsensing, edge computing, federated learning, auction theory, data security and privacy. He has published more over 100 papers in referred journals and conferences, including TMC, TC, TPDS, TON, TKDE, TSC, INFOCOM, ICDE, ICNP, etc. He served as the TPC member of INFOCOM'23, INFOCOM'22, IJCAI'22, INFOCOM'21, IJCAI'21, INFOCOM'20, INFOCOM'19, ICDCS'19, DASFAA'19, INFOCOM'18, etc. He is on the reviewer board of several top journals such as TMC, TON, TPDS, TSC, TVT, TCC, etc.

Yu Zhu received the B.S. degree in nuclear engineering and technology from Xi'an JiaoTong University. He is currently working toward the master's degree in computer science at the University of Science and Technology of China. His research focuses on reinforcement learning and multi-agent system.

Jie Wu is the Director of the Center for Networked Computing and Laura H. Carnell professor at Temple University. He also serves as the Director of International Affairs at College of Science and Technology. He served as Chair of Department of Computer and Information Sciences from the summer of 2009 to the summer of 2016 and Associate Vice Provost for International Affairs from the fall of 2015 to the summer of 2017. Prior to joining Temple University, he was a program director at the National Science Foundation and was a distinguished professor at Florida Atlantic University. His current research interests include mobile computing and wireless networks, routing protocols, network trust and security, distributed algorithms, applied machine learning, and cloud computing. Dr. Wu regularly publishes in scholarly journals, conference proceedings, and books. He serves on several editorial boards, including IEEE Transactions on Mobile Computing, IEEE Transactions on Service Computing, Journal of Parallel and Distributed Computing, and Journal of Computer Science and Technology. Dr. Wu is/was general chair/co-chair for IEEE IPDPS'08, IEEE DCOSS'09, IEEE ICDCS'13, ACM MobiHoc'14, ICPP'16, IEEE CNS'16, WiOpt'21, and ICDCN'22 as well as program chair/cochair for IEEE MASS'04, IEEE INFOCOM'11, CCF CNCC'13, and ICCCN'20. He was an IEEE Computer Society Distinguished Visitor, ACM Distinguished Speaker, and chair for the IEEE Technical Committee on Distributed Processing (TCDP). Dr. Wu is a Fellow of the AAAS and a Fellow of the IEEE. He is the recipient of the 2011 China Computer Federation (CCF) Overseas Outstanding Achievement Award.

Sheng Zhang is an associate professor in the Department of Computer Science and Technology, Nanjing University. He is also a member of the State Key Lab. for Novel Software Technology. He received the BS and PhD degrees from Nanjing University in 2008 and 2014, respectively. His research interests include distributed computing systems, edge intelligence and edge computing. To date, he has published more than 80 papers, including those appeared in JSAC, TMC, TON, TPDS, TC, MobiHoc, ICDCS, INFOCOM, SECON, IWQoS, and ICPP. He received the Best Paper Award of IEEE ICCCN 2020, the Best Paper Runner-Up Award of IEEE MASS 2012, and the Outstanding Paper Runner-Up Award of IEEE ICPADS 2021. He is the recipient of the 2020 ACM Nanjing Rising Star Award and the 2015 ACM China Doctoral Dissertation Nomination Award. He is a member of IEEE, ACM, and a senior member of CCF.

Jinrui Zhou received his B.S. degree at the Department of Statistics and Finance, the University of Science and Technology of China (USTC), Hefei, China, in 2021. He is currently a master student in the School of Computer Science and Technology at USTC. His research interests include crowdsensing, edge computing, federated learning, sequential decision-making, online learning, and applied statistics.