Gu JQ, Song C, Wu J *et al*. Unilateral control for social welfare of iterated game in mobile crowdsensing. JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY 40(1): 1-22 Jan. 2025. DOI: 10.1007/s11390-023-3041-0 CSTR: 32374.14.s11390-024-2896-z

Unilateral Control for Social Welfare of Iterated Game in Mobile Crowdsensing

JiQing Gu[1, 2] (顾记清), Chao Song[1, *] (宋　超), *Member, IEEE*
Jie Wu[3] (吴　杰), *Fellow, IEEE*, Li Lu[1] (鲁　力), *Member, IEEE*, and Ming Liu[1] (刘　明), *Member, IEEE*

[1] *School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China*

[2] *School of Computer Science, Chengdu University of Information Technology, Chengdu 610225, China*

[3] *Center for Networked Computing, Temple University, Philadelphia, PA 19122, USA*

E-mail: gujiqing@cuit.edu.cn; chaosong@uestc.edu.cn; jiewu@temple.edu; luli2009@uestc.edu.cn; csmliu@uestc.edu.cn

Received January 6, 2023; accepted November 24, 2023.

Abstract Mobile crowdsensing is a popular platform that takes advantage of the onboard sensors and resources on mobile nodes. The crowdsensing platform chooses to assign several sensing tasks each day, whose utility is based on the quality of harvested sensing data, the payment of transmitting data, and the recruitment of mobile nodes. The Internet Service Provider (ISP) selects a portion of access points (APs) to power on for uploading data, whose utility depends on three parts: the traffic income of transmitting sensing data, the energy cost of operating APs, and the energy cost of data transmissions by APs. The interaction between the crowdsensing platform and ISP is formulated as an iterated game, with social welfare defined as the sum of their expected utilities. In this paper, our objective is to unilaterally control social welfare without considering the opponent's strategy, with the aim of achieving a stable and maximized social welfare. To achieve this goal, we leverage the concept of a zero-determinant strategy in the game theory. We introduce a zero-determinant strategy for the vehicular crowdsensing platform (ZD-VCS) and analyze it in discrete and continuous models within the vehicular crowdsensing scenario. Furthermore, we analyze an extortion strategy between the platform and ISP. Experimental results demonstrate that the ZD-VCS strategy enables unilateral control of social welfare, leading to a high and stable value.

Keywords iterated game, social welfare control, vehicular crowdsensing, zero-determinant strategy

In Press

1 Introduction

With the rapid growth in various types of equipped sensors, including cameras, microphones, and GPS devices, mobile crowdsensing can now provide a wide range of services, such as urban monitoring[1], road and traffic condition monitoring[2], pollution level measurements, wildlife habitat monitoring[3], and cross-space public information sharing[4]. As one of the most promising Internet of Things (IoT) applications, this is referred to as vehicular crowdsensing (VCS)[5, 6]. It takes advantage of the mobility of vehicles to provide location-based services in a large-scale area, and plays a crucial role in creating a comfortable and convenient environment for the urban city.

In the crowdsensing application, the interactions between the crowdsensing platform and the Internet service provider (ISP) are treated as a game[7, 8]. We take vehicular crowdsensing in Fig.1 as a special case to study the game. The vehicular crowdsensing platform (simply called platform) is responsible for pub-

Regular Paper

This work was supported by the National Key Research and Developement Program of China under Grant Nos. 2021YFB3101302 and 2021YFB3101303, and the National Natural Science Foundation of China under Grant Nos. 62020106013 and 82241060.

*Corresponding Author

©Institute of Computing Technology, Chinese Academy of Sciences 2025

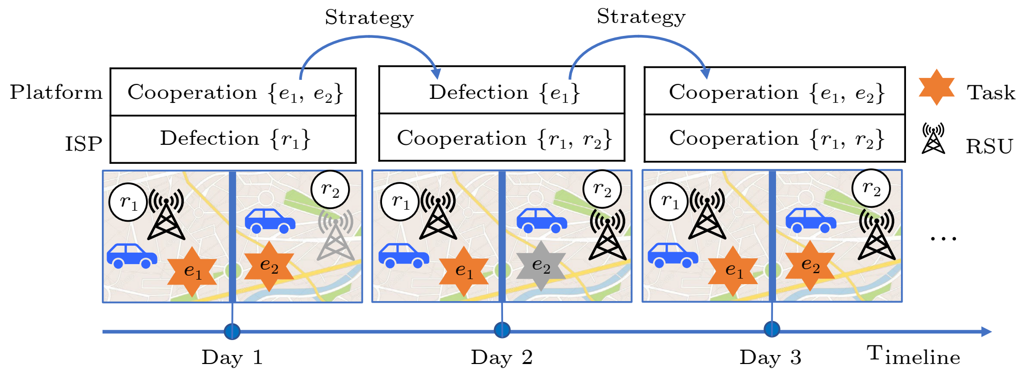


Fig.1. Iterated game between the platform and ISP in vehicular crowdsensing.

lishing the crowdsensing tasks. This income is determined by the quality of sensing data, which is calculated based on the quantity of sensing data and the diversity of tasks. The platform pays the traffic cost of transmitting sensing data via Road-Side Units (RSUs) to the ISP. To reduce energy costs, the switch-mode of cell activation is widely employed[9]. Thus, the ISP schedules to partially activate RSUs for uploading the sensing data generated by the moving vehicles[10, 11]. The income of ISP is generated from the traffic income obtained through transmitting the sensing data. When the platform publishes more tasks to cooperate, it hopes to receive a greater volume of sensing data. While ISP turns on fewer RSUs to defect, the quality of sensing data is reduced, and the platform's income is also reduced. When the ISP turns on more RSUs to cooperate, it hopes to earn more traffic income from data transmissions. However, if the platform publishes fewer tasks to defect, the income of ISP will be reduced. Fig.1 shows an iterated game in vehicular crowdsensing where the platform and ISP interact by playing a game repeatedly (infinitely many times). On each day, the crowdsensing platform publishes the selected tasks and the ISP powers on a portion of RSUs for uploading data, independently. According to their utilities and actions of the previous day, both of them will adjust their actions on the next day by their strategies.

In the iterated game, the social welfare is defined as the sum of expected utilities of the platform and ISP. Unilateral control the social welfare in the iterated games has been discussed in many previous works. These discussions include topics like controlling social welfare (total quality of data) in mobile crowdsensing[12], addressing the crowdsourcing dilemma through social welfare control[13–15], ensuring total Quality of Services (QoS) control for all participants in wireless communication[16], unilateral control over expected payoffs for both opponents and oneself in block withholding attacks[16], and administrator-led unilateral control over the total utilities of all players in wireless network resource management[17], among others. Crowdsensing games[18–20] are ubiquitous and worth studying since crowdsensing is a widely adopted method for data sensing in various daily applications[6, 21]. Social welfare is an important metric in crowdsensing games, therefore we study the social welfare control problem with different strategies to help operators in their decision-making process.

In this paper, we aim to investigate the unilateral control for social welfare of the iterated game in vehicular crowdsensing. The word "unilateral" represents a player can control the social welfare regardless of the opponent's strategy. We consider the crowdsensing platform belongs to the government[22], who concerns about the social welfare of the whole crowdsensing system, and thus the platform has the responsibility to control the social welfare regardless of ISP's actions①②.

We discuss such unilateral control for maximizing the social welfare of the iterated game with a stable value. We prove that the game between the platform and ISP exists an equilibrium in each iteration, but finding the equilibrium point would not achieve our goal of social welfare control.

Inspired by [15, 23], we propose a Zero-Determinant strategy for the Vehicular CrowdSensing platform (ZD-VCS) to control the social welfare of the crowdsensing system. The idea is that the strategy can derive a linear relationship between the expected

In Press

①https://archive.flossmanuals.net/bypassing-censorship/ch007_chapter-2-censorship.html, Nov. 2024.
②https://www.law.cornell.edu/wex/internet_service_provider_(isp), Nov. 2024.

payoff of the platform and that of the ISP, then we can set proper parameters for the to-be-adopted ZD-VCS to obtain the strategy, the details are presented in Section 5. More specifically, we analyze the problem of unilateral control for social welfare in the discrete model. We calculate the strategy ZD-VCS that the platform to-be-adopted, regardless of ISP's chosen strategies (such as TFT, Pavlov, random and evolved strategy). In the discrete case, ZD-VCS takes the form of a probability vector. In the continuous case, where the payoff function is continuous, ZD-VCS becomes a piece-wise function used to calculate the probability of cooperation. Each element of the strategy is a probability for cooperation in the current round under the actions of the previous round, and the probability is calculated by the payoff functions of the platform and ISP. The action of the platform or the ISP in each iteration is calculated by its strategy. Furthermore, we propose an extortion strategy to control the ratio of the expected utilities between the platform and the ISP. Our main contributions are summarized as follows.

1) We formulate the interactions between the platform and ISP as an iterated game and verify that the game exists an equilibrium in each iteration.

2) To help the platform establish unilaterally control, we propose a ZD-VCS strategy and analyze it in both discrete and continuous models. Furthermore, we study extortion strategy, which enforces an extortion relationship between the platform's and ISP's expected utility.

3) We implement the ZD-VCS strategy with real trace driven simulations. By setting proper parameters for the to-be-adopted ZD-VCS, experimental results show that the platform can control social welfare to achieve a high and stable value, and a ratio between the platform's and ISP's expected utilities.

This paper is organized as follows. In Section 2, we survey the related work. Section 3 presents the preliminary of the Zero-determinant strategy. Section 4 discusses the game between crowdsensing platform and ISP, and analyzes the game by Markov Decision Model. Section 5 discusses the ZD-VCS strategy in the discrete and continuous model. Section 6 evaluates the performance of the ZD-VCS strategy. The last section draws conclusions and presents our future work.

2 Related Work

We briefly review related work on vehicular crowdsensing, mobile crowdsensing game, and zero-determinant strategy.

Vehicular Crowdsensing. There are many applications under vehicular crowdsensing. Pu *et al.*[24] proposed Chimera, which is an energy-efficient and deadline-aware hybrid edge computing framework for vehicular crowdsensing applications. Morselli *et al.*[25] developed a framework for analyzing multidimensional stochastic sampling in vehicular crowdsensing, where samples are gathered from sensors on vehicles. This work is important for different kinds of applications based on environmental monitoring via IoT and vehicular communications. Campioni *et al.*[5] investigated the recruitment problems for vehicular crowdsensing and proposed several heuristics that outperform existing algorithms and obtain near optimal solutions.

Mobile Crowdsensing Game. Mobile crowdsensing (MCS) game include repeated (iterated) and static game. Some problems in mobile crowdsensing can be formulated as games, which inspires researchers an efficient way to solve them by using game theory. The work[20] introduced the repeated interactions between the MCS server and independent vehicles in a dynamic network as a dynamic vehicular crowdsensing game. A Q-learning-based MCS payment strategy and sensing strategy is proposed for the dynamic vehicular crowdsensing game. Di Stefano *et al.*[19] modeled and quantified the evolutionary dynamics of human sensing behaviors through the rounds of iterated social dilemmas, and they validated the methodology in a vehicular crowdsensing scenario. The work[18] presented an incentive mechanism for vehicular crowdsensing in the context of autonomous vehicles, so as to address the problem of sensing coverage of regions located out of the AVs planned trajectories. However, these studies focus on the interactions between task requestor and workers (users, vehicles), neglecting the role and action of ISP, and they applied the strategy obtained from reinforcement learning[20] or a greedy method[18] to maximize their utility, which may fall into local optimum.

Zero-Determinant Strategy. Zero-determinant strategy can achieve different goals with different parameter settings. With zero-determinant strategy, the player can control the total expected utilities of players as a stable value, unilaterally set the expected utility of an opponent or set a ratio between his and his opponent's expected payoff, regardless of the op-

In Press

ponent's strategy. Zero-determinant strategy also evolves to extortion and generosity strategies[26]. Press *et al.*[23] proposed a zero-determinant strategy in 2×2 Iterated Prisoner's Dilemma game, and then the zero-determinant strategy is extended into the general 2×2 iterated game by other researchers and has many applications[27, 28]. The work[29] investigated the power control problem in resource sharing among wireless users and network operators, and the network operator applied zero-determinant strategies to control social welfare. Previous work[12] formulated the interaction between a requestor and any workers in mobile crowdsensing as an iterated game, intending to improve data quality in mobile crowdsensing quality control. Hu *et al.*[13] proposed a zero-determinant strategy to address the malicious attack problem in crowdsourcing. A sequential zero-determinant strategies[15] is applied for quality control in crowdsourcing. Zero-determinant strategy is also extended for multi-player multi-action iterated games[30]. The work[31] introduced a more comprehensive class of autocratic strategies, by extending the concept of zero-determinant strategies to iterated games with more diverse and generalized action spaces. In order to address the challenge of dimensionality that arises when the complexity of games escalates, the work[32] presented a novel mathematical framework for analyzing strategic choices in repeated games with a varying number of actions or players, as well as arbitrary continuation probabilities. Our goal in this paper is to study unilaterally social welfare control in the iterated vehicular crowdsensing game.

3 Preliminary: Zero-Determinate Strategy

Zero-determinant strategy[23] comprises a set of strategies in general memory-one iterated game. With different parameter settings, the zero-determinant strategy can generate different strategies with different goals such as controlling the total expected utilities of players as a stable value, unilaterally controlling the other player's payoff, or setting a ratio between its and the other player's payoff.

We utilize a Markov chain to illustrate the zero-determinant strategy. The iterated game starts with an initial action, the actions of players in each iteration are obtained from the strategy, and each player performs an action and obtains a utility (reward) in each round. The strategies of players decide a stochastic process. The state of each player are represented by their union action of the previous round. It is proven that long-memory player in the iterated game has no advantage over the short-memory player[23], since players in each iteration determine their actions based on the outcome of the previous round. This means the corresponding stochastic process can be represented by a Markov chain, where the state transitions are joint probabilities calculated from the probabilistic strategies of players.

In the discrete case, the zero-determinant strategy is a probability vector. In contrast, in the continuous case (payoff function is continuous), the zero-determinant strategy is a piece-wise function to calculate the probability of cooperation. In the two cases, each probability element is the probability of cooperation in the current round under the actions of the previous round, and the probability is calculated by the payoff functions in the game. A discrete case is taken as an example, and both players have two actions: {cooperation, defection} ($\{c, d\}$ for short). Their action pairs are $\{cc, cd, dc, dd\}$. Each player has a mixed strategy at each round, which denotes the probabilities for the next cooperation under the actions in the previous round. Accordingly, we define the mixed strategy of player X as $\boldsymbol{p} = (p_1, p_2, p_3, p_4)$ and that of his/her opponent Y as $\boldsymbol{q} = (q_1, q_2, q_3, q_4)$. p_i represents that X chooses cooperation in the current round conditioned on the i-th action pair of the previous round, and q_i has a similar meaning. We assume the corresponding payoff matrices of X and Y are $\boldsymbol{P} = (P_1, P_2, P_3, P_4)$ and $\boldsymbol{Q} = (Q_1, Q_2, Q_3, Q_4)$, respectively. P_i (or Q_i) is the utility of X (or Y) corresponding to the i-th actions pair. Let $\boldsymbol{p}$ be a zero-determinant strategy. If a player adopts zero-determinant strategy $\boldsymbol{p}$ with different settings, the player can unilaterally set the expected utility of an opponent, control the total expected utilities of players as a stable value, or control a ratio between the player's and his/her opponent's expected payoff, etc., regardless of the strategy of his/her opponent. That is, there is a linear relationship between the expected payoff of two players. When player X adopts zero-determinant strategy $\boldsymbol{p}$ in the iterated game, $\boldsymbol{p}$ satisfies the following equation:

$$p_i = \begin{cases} \phi(\alpha P_i + \beta Q_i + \gamma) + 1, & i = 1,\ 2, \\ \phi(\alpha P_i + \beta Q_i + \gamma), & i = 3,\ 4, \end{cases}$$

where α, β, γ, ϕ are parameters, $\phi \neq 0$, and p_i could be calculated by the parameters and payoff matrices. No matter which strategy Y adopts, the expected payoff of X and Y ($U^{\rm x}$ and $U^{\rm y}$) satisfies:

In Press

$$\alpha U^{x} + \beta U^{y} + \gamma = 0. \tag{1}$$

For example, when $\alpha = 0$, U^{y} is controlled by X. With different settings of parameters, equation (1) generates various strategies. Such strategies, which are referred to as zero-determinant strategies, are realized if the process of the game can be formulated as a one-step Markov process. The frequently-used notations are listed in Table 1.

Table 1. Description of Frequently-Used Notations

Notation	Description
$E = \{e_1, \ldots, e_{\rm M}\}$	Set of M candidate sensing tasks
$R = \{r_1, \ldots, r_{\rm N}\}$	Set of N candidate RSUs
$x_{\rm e}/x_{\rm r}$	Number of selected sensing tasks/RSUs
$\boldsymbol{W}$	Task-RSU weighted traffic matrix
$\vec{\boldsymbol{X}}_{\rm e}/\vec{\boldsymbol{X}}_{\rm r}$	Vector of selected sensing tasks/RSUs
$f_{\rm g}/f_{\rm t}$	Payoff function of platform/ISP
$u_{\rm r}/u_{\rm t}$, C	Energy cost/traffic cost, operating cost
$\boldsymbol{M}^{\rm e}$, $\boldsymbol{M}^{\rm r}$	Payoff matrix of platform and ISP
$m_i^{\rm e}$, $m_i^{\rm r}$	Elements in $M^{\rm e}$ and $M^{\rm r}$
$\boldsymbol{p}$, $\boldsymbol{q}$	Mixed strategy of platform and ISP
$\boldsymbol{v}_{\rm s}/\boldsymbol{f}$	Stationary vector/any vector
$U^{\rm e}/U^{\rm r}/U_{\rm all}$	Utility of platform/ISP/both
$\boldsymbol{H}$	Markov state transition matrix
α, β, γ	Parameters to determine the ZD-VCS strategy
χ	Extortion factor
$l^{\rm r}/l^{\rm e}$	Lowest number of RSU/sensing tasks
$h^{\rm r}/h^{\rm e}$	Highest of RSU/sensing tasks

4 System Model

In this section, we introduce the mobile crowdsensing model and game formulation between the platform and ISP.

4.1 Mobile Crowdsensing Model

Considering the general scenario of the mobile crowdsensing shown in Fig.2, there are three participants: crowdsensing platform, mobile nodes, and wireless access point (AP). The platform is responsible for publishing the crowdsensing tasks (such as monitoring the temperatures in some areas[3, 33]) and assigning the tasks to mobile nodes (e.g., vehicles, mobile users). We denote the set of M sensing tasks as $E = \{e_1, \ldots, e_{\rm M}\}$. The crowdsensing platform could belong to the government[22] or a service platform company (such as Uber[34]). The recruited mobile nodes are responsible for sensing the data from sensing tasks. The platform pays a constant operating cost (denoted as C) to the recruited mobile nodes every day. The mobile nodes sense data passively when they are in the sensing areas, instead of actively going to the sensing areas. The incentive mechanism for mobile nodes has been proposed in [14, 35, 36], but it is out of the scope of our paper.

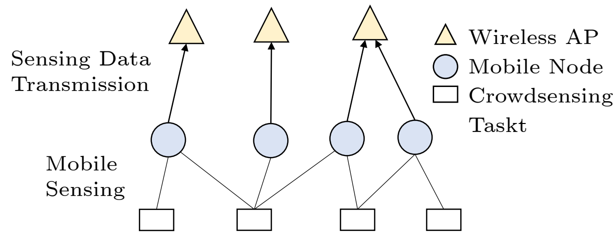
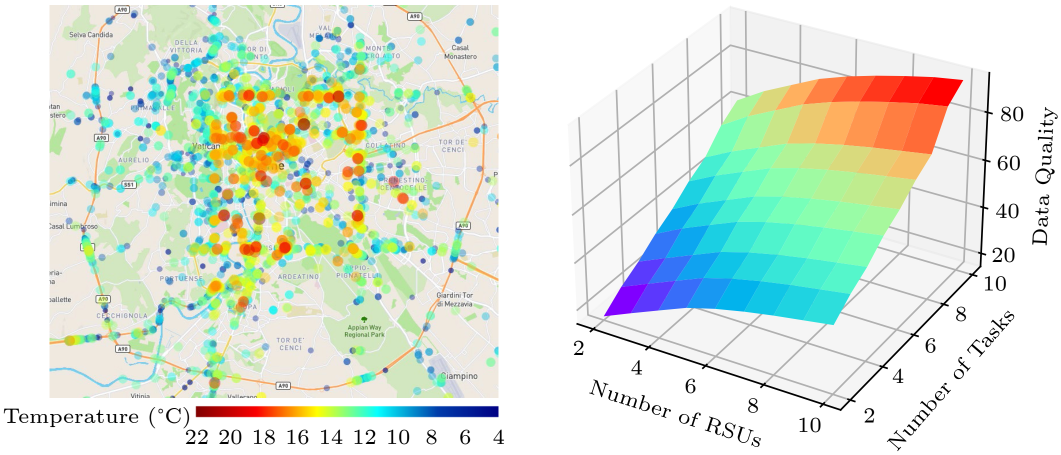


Fig.2. Mobile crowdsensing model.

The wireless APs, such as Wi-Fi, femtocell③, Smart Pole④, and other RSUs, are installed by the ISP and provide communication for the mobile nodes, which are denoted by $R = \{r_1, \ldots, r_{\rm N}\}$. These N APs are responsible for uploading the sensing data generated by the mobile nodes. Generally, compared with a cell-tower in cellular systems, the coverage of each AP is relatively small[37], such as the coverage of RSU is about $100 - 500$ meters. As a result, it is hard to provide seamless roaming for vehicles[38], therefore the sensing data's offloading can be delayed during its lifetime through an access point that acts as a gateway. If there are no available APs, some sensing data can be delayed instead of being immediately sent or received over the AP. This kind of technology has been extensively investigated[39, 40]. The distributions of sensing tasks, mobile modes, and the available APs have an influence on the quality of sensing data. Data quality can only be measured when the quantity of sensing data is large enough. The quantity of sensing data is closely related to sensing tasks, mobile modes, and the available APs. When the mobile nodes sense the data in the area of a sensing task, the sensing data are transmitted through an AP, and then the sensing data can be uploaded successfully. When the wireless APs and crowdsensing tasks belong to different operators, the interactions between the published tasks and the available APs form a game. We formu-

③http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white_paper_c11-520862.html, Nov. 2024.
④https://www.omniflow.io/smartpole, Nov. 2024

In Press



(a) (b)

Fig.3. Data analysis of the Roma temperature data in vehicular crowdsensing. (a) Sensing temperature. (b) Data quality.

late this game in Subsection 4.2.

Fig.3(a) is an example of a temperature monitoring scenario in Rome, Italy by the real data trace⑤. The points in Fig.3(a) are the temperature data sensed by the vehicles, which is also called sensing data. The color of a point close to red refers to the place with a higher temperature, and that close to blue refers to the place with a lower temperature. On each day, the platform publishes several tasks for sensing the temperatures of some areas in Fig.1, and the ISP powers on a portion of RSUs for uploading the sensing data to save energy costs. When a vehicle passes by an area that needs to be sensed, it will generate a sensing data packet with temperature records. When a vehicle leaves the current sensing area with sensing data and moves into the communication range of a working RSU, the sensing data packet is uploaded to a remote cloud server via the RSU.

The traffic distribution of vehicles is different each day, therefore different RSUs have different contributions to the data quality in crowdsensing. In order to measure the importance of each RSU, when giving the sensing tasks with priorities, we apply a matching algorithm to obtain the selected important RSUs. Thus, the number of RSUs can be used as the ISP's action in the game, which is formulated in the following Subsection 4.2. Fig.3(b) shows the trends of data quality in different numbers of sensing tasks and RSUs.

4.2 Game Formulation

We denote the action of assigning tasks by the platform as a vector $\vec{\boldsymbol{X}}_{\rm e}$ with dimension M, where the element is 0 or 1, indicating whether a candidate task is selected or not. Let $x_{\rm e}$ denote the number of selected tasks, i.e., $x_{\rm e} = ||\vec{\boldsymbol{X}}_{\rm e}||_1$. The action of operating RSUs by ISP is denoted by a vector $\vec{\boldsymbol{X}}_{\rm r}$ with dimension N, where the element is 0 or 1, indicating whether the RSU is powered on or not. Let $x_{\rm r}$ denote the number of selected RSUs, $x_{\rm r} = ||\vec{\boldsymbol{X}}_{\rm r}||_1$. We denote $\boldsymbol{W} \in \mathbb{R}^{M\times N}$ as a task-RSU weighted traffic matrix, which element $\boldsymbol{W}_{ij}$ is the number of sensing data uploaded by the RSU j for task i. $\boldsymbol{W}^{x_{\rm e},x_{\rm r}} = (\vec{\boldsymbol{X}}_{\rm e}^{\rm T} \cdot \vec{\boldsymbol{X}}_{\rm r}) \circ \boldsymbol{W}$, which refers to the traffic distribution of $x_{\rm e}$ sensing tasks under $x_{\rm r}$ RSUs. The symbol $\circ$ refers to the Hadamard product, and the element $\boldsymbol{W}_{i,j}^{x_{\rm e},x_{\rm r}}$ also represents the traffic count of the i-th sensing task under the j-th RSU.

We take an example to illustrate how to obtain the matrix $\boldsymbol{W}^{x_{\rm e},x_{\rm r}}$. In the case of $\vec{\boldsymbol{X}}_{\rm e} = (1,\ 0,\ 1,\ 1)$ and $\vec{\boldsymbol{X}}_{\rm r} = (1,\ 0,\ 1)$, that is, $M = 4$, $N = 3$, $x_{\rm e} = 3$, and $x_{\rm r} = 2$. Then,

$$\vec{\boldsymbol{X}}_{\rm e}^{\rm T} \cdot \vec{\boldsymbol{X}}_{\rm r} = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 1 \\ 1 & 0 & 1 \end{pmatrix}.$$

When $\boldsymbol{W}$ is set as:

$$\boldsymbol{W} = \begin{pmatrix} 15 & 20 & 31 \\ 40 & 20 & 26 \\ 20 & 20 & 18 \\ 18 & 30 & 16 \end{pmatrix},$$

$\boldsymbol{W}^{3,2}$ is calculated as:

⑤http://crawdad.org/roma/taxi/, Nov. 2024.

$$\boldsymbol{W}^{3,2} = ({\vec{\boldsymbol{X}}_{\mathrm{e}}}^{\mathrm{T}} \cdot \vec{\boldsymbol{X}}_{\mathrm{r}}) \circ \boldsymbol{W} = \begin{pmatrix} 15 & 0 & 31 \\ 0 & 0 & 0 \\ 20 & 0 & 18 \\ 18 & 0 & 16 \end{pmatrix},$$

$\boldsymbol{W}^{3,2}$ refers to the traffic distribution of three sensing tasks under two RSUs.

We investigate the actions with different numbers of tasks and RSUs in the iterated game, and simplify each action with x_{e} tasks or x_{r} RSUs to only one case. Thus, we use x_{e} or x_{r} to represent the unique action, respectively. That is, x_{e} and x_{r} denote the number of selected tasks and RSUs, respectively. Generally, the sensing tasks and RSUs of different areas have different priorities, and top x_{e} tasks and and x_{r} RSUs with higher priorities are selected. Bellow we illustrate how to calculate the priorities.

In this paper, we use vehicular crowdsensing as a case study. We use the Taxi-Roma dataset⑤ to explain how to determine the priority of crowdsensing tasks. The dataset includes the GPS coordinates of 320 taxi drivers that working in the center of Rome. These GPS coordinates are collected over 30 days. The dataset is preprocessed by filtering out some outliers. The traces cover an area with a range of 66 km × 59 km, and we divide the area into 10 × 10 grids. We assume that each RSU in the grid serves as a gateway and each grid is viewed as an area with a sensing task. The priority of each sensing task and RSU are calculated by the number of GPS coordinates in each grid. The more GPS coordinates in a grid, the higher priority of the each sensing task and RSU in this grid.

4.2.1 Payoff of Platform

In our vehicular crowdsensing scenario, the platform earns values from the quality of harvested sensing data and the diversity of tasks, paying for the data transmission and recruitment of vehicles. We assume the size of each sensing data packet is the same[41]. The data quality is defined as

$$Q_{\mathrm{d}}(x_{\mathrm{e}},\ x_{\mathrm{r}}) = \sum\nolimits_{i,\ j} \left(\log(W_{i,\ j}^{x_{\mathrm{e}},\ x_{\mathrm{r}}} + 1) - P_{ij} \log P_{ij} \right),$$

where $1 \leqslant i \leqslant M, 1 \leqslant j \leqslant N$, the term of $\sum_{i,j} \log(W_{i,j}^{x_{\mathrm{e}},x_{\mathrm{r}}} + 1)$ reflects the growth rate of platform's data quality decreases (diminishing return) as the increment of sensing data[42], and the log function also reflects that the redundant sensing data cannot contribute much to the data quality. We utilize entropy $-P_{ij} \log P_{ij}$ to model the diversity of sensing data[43], which reflects the task and transmission diversity. The term $P_{ij} = W_{i,j}^{x_{\mathrm{e}},x_{\mathrm{r}}} / \sum_{i,j} W_{i,j}^{x_{\mathrm{e}},x_{\mathrm{r}}}$.

The utility of the vehicular crowdsensing platform depends on the data quality of the sensing data, payment of transmitting them and recruiting vehicles. We formulate the payoff function of the platform as the value of sensing data minus the costs of data transmission, and the payoff function is represented as

$$f_{\mathrm{g}}(x_{\mathrm{e}}, x_{\mathrm{r}}) = u_{\mathrm{d}} \times Q_{\mathrm{d}}(x_{\mathrm{e}}, x_{\mathrm{r}}) - u_{\mathrm{t}} \times \sum\nolimits_{i,j} W_{i,j}^{x_{\mathrm{e}},x_{\mathrm{r}}} - C, \quad (2)$$

where u_{d} is the benefit that per data quality could bring, u_{t} is the transmission price per data traffic and the constant C refers to the cost of recruiting vehicles. Note that the platform hopes to publish more tasks, so as to cover more areas with more sensing data and obtain more utility.

4.2.2 Payoff of ISP

Fig.3(b) shows the trend of the data quality with the selected x_{e} sensing tasks and x_{r} RSUs by the real data trace⑤. We notice that the data quality is increased with more sensing tasks and RSUs, which determines the platform's income. To transmit more sensing data, the ISP is required to power on more RSUs, which causes more energy costs. The ISP earns the traffic income of transmitting the sensing data, and pays the energy costs of operations of RSUs and data transmissions by RSUs. Thus, we formulate the payoff function of ISP as the traffic income of sensing data minus the energy costs on RSUs as follows

$$f_{\mathrm{t}}(x_{\mathrm{e}}, x_{\mathrm{r}}) = (u_{\mathrm{t}} - u_{\mathrm{e}}) \times \sum\nolimits_{i,j} W_{i,j}^{x_{\mathrm{e}},x_{\mathrm{r}}} - x_{\mathrm{r}} u_{\mathrm{r}}, \quad (3)$$

where x_{r} represents the number of RSUs powered on, u_{r} refers to the per cost of operating an RSU, and u_{e} refers to the average energy cost[44] to transmit per sensing data.

From (2) and (3), we see that the platform's cost of uploading sensing data is $u_{\mathrm{t}} \times \sum_{i,j} W_{i,j}^{x_{\mathrm{e}},x_{\mathrm{r}}}$, which is also the income of the ISP. In (3), when the number of sensing tasks reaches a certain level, the utility of the traffic from the sensing data is greater than the energy costs of RSUs, and thus the ISP is willing to power on more RSUs.

4.2.3 Discussion

Fig.4 is an example of two actions in the game. Like prisoner's dilemma, cooperation denotes the

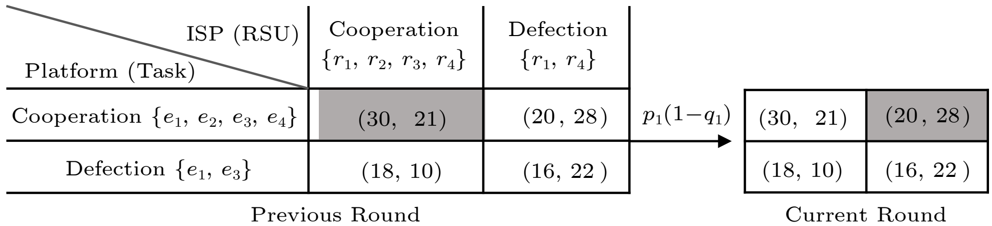


Fig.4. Probability transfer process of previous round to current round.

highest number of sensing tasks or RSUs, and defection refers to the lowest. $x_e, x_r \in \{\text{cooperation}, \text{defection}\}$. This example sets $x_e = x_r = 4$ when the platform and ISP cooperate and $x_e = x_r = 2$ when they defect. The value pair in Fig.4 is the utilities of the platform and ISP. In each iteration, they move on to the next round based on the actions generated by their probabilistic strategies.

4.3 Nash Equilibrium

We analyze the equilibrium of the game between the platform and ISP in each iteration. Proofs of all theorems are presented in the supplementary file⑥.

Theorem 1. *The game between the crowdsensing platform and ISP exists equilibrium.*

Theorem 2. *The equilibrium point is unique in the game between the crowdsensing platform and ISP.*

Next, we take Fig.4 as an example to illustrate the equilibrium point. We can obtain equilibrium points from a solver⑦. In Fig.4, from the platform's view, no matter what the ISP's action is, its best action is *cooperation* $(x_e = 4)$, and similarly, the ISP's best action is *defection* $(x_r = 2)$. Thus, this game has an equilibrium of (20, 28), where the social welfare is $20 + 28$, which is certainly less than the case when both cooperate. Generally, an ISP does not know how many sensing tasks the platform will publish, therefore a selfish ISP tends to power on fewer RSUs in each game round, resulting in lower social welfare. Thus, an equilibrium point is not the optimal solution to maximize social welfare.

4.4 MDP in the Iterated Game

The iterated game is viewed as a Markov decision process (MDP)[45]. For each player, an MDP is modeled as follows.

1) *Action Space*. The action space of the platform and ISP are denoted as $A_e = \{l^e, h^e\}, (x_e \in A_e)$ and $A_r = \{l^r, h^r\}, (x_r \in A_r)$ respectively. The platform chooses the action of assigning the highest or the lowest number of tasks, denoted as h^e and l^e respectively. The ISP determines the action on providing the highest or the lowest number of RSUs, and denoted as h^r and l^r respectively.

2) *State Space*. The state space is the action pairs in the previous round, and denoted as $S = \{(x_e, x_r)\} = \{(h^e h^r), (h^e l^r), (l^e h^r), (l^e l^r)\}$, $s_t \in S$.

3) *Reward*. R_e or R_r is the immediate reward received after transitioning from state s to state s' for platform and ISP, due to action a. We define the payoff matrices of the platform and ISP in each round as $\boldsymbol{M}^e$ and $\boldsymbol{M}^r$, respectively. $\boldsymbol{M}^e = (f_g(h^e, h^r), f_g(h^e, l^r), f_g(l^e, h^r), f_g(l^e, l^r))^T$, and $\boldsymbol{M}^r = (f_t(h^e, h^r), f_t(h^e, l^r), f_t(l^e, h^r), f_t(l^e, l^r))^T$.

4) *State Transition Probabilities*. $P_a(s, s') = \Pr(s_{t+1} = s' \mid s_t = s, a_t = a)$ is the probability that action a in state s at the round t will lead to state s' at the $t+1$ round.

A policy function π is a (potentially probabilistic) mapping from state space S to action space A. Once an MDP is combined with a policy, a Markov Chain forms. Since the action chosen in state s is completely determined by $\pi(s)$, the state transition probability $\Pr(s_{t+1} = s' \mid s_t = s, a_t = a)$ can reduce to $\Pr(s_{t+1} = s' \mid s_t = s)$ (a Markov transition matrix).

Briefly speaking, in the Markov Decision Process, a player will repeatedly observe the current state s_t of the environment and take action from all available actions in this state. Then, the state will transfer to s_{t+1}, and the agent will get a reward R_t from the environment for its action.

In the iterated game, due to RSU's actions are variable, the social welfare could change sometimes high and sometimes low, which is unstable. Thus, we aim to control social welfare at a stable possible maximal value, and we model the problem of social wel-

⑥https://github.com/gujiqing/supplementary-file/blob/main/JCST-Template-%20submit%20supplementary%20file.pdf, Nov. 2024.

⑦https://ww2.mathworks.cn/matlabcentral/fileexchange/27837-n-person-game, Nov. 2024 fare control as follows.

Problem 1 (**Social Welfare Control**). *The social welfare is required to be controlled at a stable and maximal possible value, which indicates the social welfare is showing little changes and maintains around a maximal possible value with increase of the game iteration.*

$$\begin{aligned} max \quad & U_{\mathrm{all}} = \alpha U^{\mathrm{e}} + \beta U^{\mathrm{r}} \\ s.t. \quad & U^{\mathrm{e}} + U^{\mathrm{r}} = -\gamma, \end{aligned} \tag{4}$$

where α and β refer to the weights of expected utilities U^{e} and U^{r}, respectively, and γ is a variable in a range.

The constraint in (4) refers to the social welfare is a constant. The objective function and the constraint determine that the goal is to maintain social welfare at a table and possible maximal value.

5 Social Welfare Control with ZD-VCS

In this section, we analyze the iterated game with the aim of social welfare control by zero-determinant strategy, and then we discuss the discrete and continuous strategies.

5.1 Game Analysis: Zero-Determinant Strategy

We aim to maximize the social welfare of the iterated game with a stable value, and the social welfare is the sum of the expected payoff of the platform and ISP. To achieve this goal, we take advantage of the zero-determinant strategy in game theory and propose a scheme named Zero-Determinant strategy for Vehicular CrowdSensing platform (ZD-VCS) to control social welfare. This strategy is a vector of conditional probabilities. Each element in the vector is a probability for cooperation in the current round under the actions of the previous round, and the probability is calculated by the payoff functions of the platform and ISP. Fig.4 shows the transfer process of the previous round to the current round, $p_1(1-q_1)$ is the probability from one state to another state. To simplify this game, we consider the discrete strategy, where the platform and ISP adopt either an extremely amicable or vicious action. While a continuous one refers to any integer action as long as it is in the corresponding continuous strategy space. It is essential to explore the integer actions because a larger action space may lead to higher social welfare.

5.2 ZD-VCS Strategy in the Discrete Model

In the discrete model, the player's action is private at each round; thus, there are four outcomes for each game iteration. We label the four outcomes of each iteration as 1, 2, 3 and 4, respectively, corresponding to the four states. We assume that the two players (platform and ISP) only have the state memory of the previous round. In the game, both players have mixed strategies at each round, denoting the cooperation probabilities under the four possible states of outcomes in the previous round. Accordingly, we define the mixed strategy of the platform as $\boldsymbol{p} = (p_1,\ p_2,\ p_3,\ p_4)$, which is the zero-determinant strategy of the platform. The mixed strategy of the ISP is $\boldsymbol{q} = (q_1,\ q_2,\ q_3,\ q_4)$. Here, $p_1,\ p_2,\ p_3,\ p_4$ and q_1, q_2, q_3, q_4 are the probabilities of choosing h^{e} or h^{r} in the current round when the outcome of the previous round is $x_{\mathrm{e}}x_{\mathrm{r}} = (h^{\mathrm{e}}h^{\mathrm{r}},\ h^{\mathrm{e}}l^{\mathrm{r}},\ l^{\mathrm{e}}h^{\mathrm{r}},\ l^{\mathrm{e}}l^{\mathrm{r}})$. Fig.4 shows the probability transfer process of the previous round's action to the current round's action. In the current round, we denote the possibilities of the four potential states of outcomes as $\boldsymbol{v} = (v_1,\ v_2,\ v_3,\ v_4)^{\mathrm{T}}$, where $\sum_{i=1}^{4} v_i = 1$. Thus, the expected payoffs of the platform and ISP are $U^{\mathrm{e}} = \boldsymbol{v}^{\mathrm{T}}\boldsymbol{M}^{\mathrm{e}}$ and $U^{\mathrm{r}} = \boldsymbol{v}^{\mathrm{T}}\boldsymbol{M}^{\mathrm{r}}$, respectively. Note that a character in bold in this paper refers to a vector or a matrix.

Based on the definitions of $\boldsymbol{p}$ and $\boldsymbol{q}$, we denote the Markov state transition matrix as $\boldsymbol{H}$, and the stationary vector as $\boldsymbol{v}_{\mathrm{s}}$. $\boldsymbol{v}_{\mathrm{s}}^{\mathrm{T}}\boldsymbol{H} = \boldsymbol{v}_{\mathrm{s}}^{\mathrm{T}}$, where

$$\boldsymbol{H} = \begin{pmatrix} p_1q_1 & p_1(1-q_1) & (1-p_1)q_1 & (1-p_1)(1-q_1) \\ p_2q_2 & p_2(1-q_2) & (1-p_2)q_2 & (1-p_2)(1-q_2) \\ p_3q_3 & p_3(1-q_3) & (1-p_3)q_3 & (1-p_3)(1-q_3) \\ p_4q_4 & p_4(1-q_4) & (1-p_4)q_4 & (1-p_4)(1-q_4) \end{pmatrix},$$

each element in $\boldsymbol{H}$ represents the transition probability from one state to another state. For example, the state $p_1(1-q_1)$ refers to the transition probability of the game from the state $h^{\mathrm{e}}h^{\mathrm{r}}$ to the state $h^{\mathrm{e}}l^{\mathrm{r}}$.

Inspired by [23], we suppose $\boldsymbol{H}' = \boldsymbol{H} - \boldsymbol{I}$, then $\boldsymbol{v}_{\mathrm{s}}^{\mathrm{T}}\boldsymbol{H}' = 0$. According to Cramer's rule, we can obtain $\mathrm{Adj}(\boldsymbol{H}')\boldsymbol{H}' = \det(\boldsymbol{H}')\boldsymbol{I} = 0$, where $\mathrm{Adj}(\boldsymbol{H}')$ denotes the adjugate matrix of $\boldsymbol{H}'$. Then, we obtain that every row of $\mathrm{Adj}(\boldsymbol{H}')$ is proportional to $\boldsymbol{v}_{\mathrm{s}}^{\mathrm{T}}$. Thus, the dot product of any vector $\boldsymbol{f}$ with the stationary vector $\boldsymbol{v}_{\mathrm{s}}$ is calculated as follows,

In Press

$$\boldsymbol{v}_{\mathrm{s}}^{\mathrm{T}} \cdot \boldsymbol{f} = D(\boldsymbol{p},\ \boldsymbol{q},\ \boldsymbol{f}) = \det\begin{pmatrix} p_1q_1-1 & p_1-1 & q_1-1 & f_1 \\ p_2q_2 & p_2-1 & q_2 & f_2 \\ p_3q_3 & p_3 & q_3-1 & f_3 \\ p_4q_4 & p_4 & q_4 & f_4 \end{pmatrix}. \tag{5}$$

It is clear that the second column of (5) consists of the elements of $\boldsymbol{p}$, and can be determined by the platform alone, which is denoted as $\tilde{\boldsymbol{p}} = (p_1-1,\ p_2-1,\ p_3,\ p_4)^{\mathrm{T}}$. Hence, when $\boldsymbol{f} = \alpha \boldsymbol{M}^{\mathrm{e}} + \beta \boldsymbol{M}^{\mathrm{r}} + \gamma \boldsymbol{1}$, where α and β are weight factors, we have $\boldsymbol{v}_{\mathrm{s}}^{\mathrm{T}} \cdot \boldsymbol{f} = \boldsymbol{v}_{\mathrm{s}}^{\mathrm{T}} \cdot (\alpha \boldsymbol{M}^{\mathrm{e}} + \beta \boldsymbol{M}^{\mathrm{r}} + \gamma \boldsymbol{1}) = \alpha U^{\mathrm{e}} + \beta U^{\mathrm{r}} + \gamma$, where γ is a scalar, U^{e} and U^{r} are the expected utility in the final stable state. Based on (5), we also have $\alpha U^{\mathrm{e}} + \beta U^{\mathrm{r}} + \gamma = D(\boldsymbol{p}, \boldsymbol{q}, \alpha \boldsymbol{M}^{\mathrm{e}} + \beta \boldsymbol{M}^{\mathrm{r}} + \gamma \boldsymbol{1})$. Namely, when $\tilde{\boldsymbol{p}} = \phi(\alpha \boldsymbol{M}^{\mathrm{e}} + \beta \boldsymbol{M}^{\mathrm{r}} + \gamma \boldsymbol{1})(\phi \neq 0)$, the corresponding matrix's second column is proportional to the fourth column. According to the properties of the matrix determinant, we have

$$\alpha U^{\mathrm{e}} + \beta U^{\mathrm{r}} + \gamma = 0. \tag{6}$$

Equation (6) indicates the expected payoffs have a liner relationship, which is brought by the $\tilde{\boldsymbol{p}} = \phi(\alpha \boldsymbol{M}^{\mathrm{e}} + \beta \boldsymbol{M}^{\mathrm{r}} + \gamma \boldsymbol{1})(\phi \neq 0)$. Note that $\tilde{\boldsymbol{p}}$ is determined by $\boldsymbol{p}$, and $\tilde{\boldsymbol{p}} = (p_1-1,\ p_2-1,\ p_3,\ p_4)^{\mathrm{T}}$. Therefore, the strategy $\boldsymbol{p}$ adopted by the platform is known as a ZD-VCS strategy. We define the weighted social welfare of this game as follows:

$$U_{\mathrm{all}} = \alpha U^{\mathrm{e}} + \beta U^{\mathrm{r}} = -\gamma. \tag{7}$$

The above analysis implies that when the platform adopts ZD-VCS strategy, the platform has unilateral control over the social welfare at a desired value ($U_{\mathrm{all}} = -\gamma$) no matter what strategy the ISP adopts. This provides the platform a powerful tool to maintain the stability of total utility. The maximal and stable social welfare that the platform maintains regardless of the ISP's strategy can be achieved by solving the following problem:

$$\begin{aligned} \max \quad & U_{\mathrm{all}} = \alpha U^{\mathrm{e}}(\boldsymbol{p},\boldsymbol{q}) + \beta U^{\mathrm{r}}(\boldsymbol{p},\boldsymbol{q}), \quad \forall \boldsymbol{q}, \\ \text{s.t.} \quad & \begin{cases} 0 \leqslant \boldsymbol{p} \leqslant \boldsymbol{1}, \\ \alpha U^{\mathrm{e}} + \beta U^{\mathrm{r}} + \gamma = 0. \end{cases} \end{aligned}$$

The terms $U^{\mathrm{e}}(\boldsymbol{p},\boldsymbol{q})$ and $U^{\mathrm{r}}(\boldsymbol{p},\mathrm{q})$ refer to the expected payoffs of the platform and ISP, which are determined by $\boldsymbol{p}$ and $\boldsymbol{q}$. Accordingly, it is equivalent to solving the following problem:

$$\begin{aligned} \min \quad & \gamma \\ \text{s.t.} \quad & \begin{cases} 0 \leqslant \boldsymbol{p} \leqslant \boldsymbol{1}, \\ \tilde{\boldsymbol{p}} = \phi(\alpha \boldsymbol{M}^{\mathrm{e}} + \beta \boldsymbol{M}^{\mathrm{r}} + \gamma \boldsymbol{1}), \\ \phi \neq 0. \end{cases} \end{aligned} \tag{8}$$

Next, we discuss how to calculate each element of $\boldsymbol{p}$. Specifically, we divide the discussion by $\phi > 0$ and $\phi < 0$, respectively. ϕ is a scaling coefficient that controls the convergence rate to the stable state.

When ϕ is positive, we put the constraint $\boldsymbol{p} \geqslant 0$ into the second constraint of (8), and get

$$\begin{aligned} &\gamma_{\min} = \max(\tau_i) \quad \forall i \in \{1,\ 2,\ 3,\ 4\}, \\ &\tau_i = \begin{cases} -\alpha m_i^{\mathrm{e}} - \beta m_i^{\mathrm{r}} - 1/\phi, & i = 1,\ 2, \\ -\alpha m_i^{\mathrm{e}} - \beta m_i^{\mathrm{r}}, & i = 3,\ 4. \end{cases} \end{aligned} \tag{9}$$

Correspondingly, we put the constraint $\boldsymbol{p} \leqslant 1$ into the second constraint of (8) and obtain

$$\begin{aligned} &\gamma_{\max} = \min(\tau_j) \quad \forall j \in \{5,\ 6,\ 7,\ 8\}, \\ &\tau_j = \tau_{i+4} = \begin{cases} -\alpha m_i^{\mathrm{e}} - \beta m_i^{\mathrm{r}}, & i = 1,\ 2, \\ -\alpha m_i^{\mathrm{e}} - \beta m_i^{\mathrm{r}} + 1/\phi & i = 3,\ 4, \end{cases} \end{aligned}$$

whereϕ is a positive value that normalizes $\boldsymbol{p}$ in the range $[0,\ 1]$. Note that γ is feasible only when it satisfies $\gamma_{\min} \leqslant \gamma_{\max}$.

Similarly, when ϕ is negative and normalizes $\boldsymbol{p}$ in the range $[0,\ 1]$. Considering the constraint $\boldsymbol{p} \geqslant 0$, we have

$$\gamma_{\min} = \max(\tau_j),\ \forall j \in \{5,\ 6,\ 7,\ 8\}. \tag{10}$$

While when considering $\boldsymbol{p} \leqslant 1$, we have $\gamma_{\max} = \min(\tau_i)$, $\forall i \in \{1,\ 2,\ 3,\ 4\}$. Therefore, γ is feasible when $\gamma_{\min} \leqslant \gamma_{\max}$, that is $\max(\tau_j) \leqslant \min(\tau_i)$, $\forall i \in \{1,\ 2,\ 3,\ 4\}$, $\forall j \in \{5,\ 6,\ 7,\ 8\}$.

According to (9) and (10), when γ reaches $\gamma_{\min}$, each element of $\boldsymbol{p}$ is represented as follows:

$$p_i = \begin{cases} \phi(\alpha m_i^{\mathrm{e}} + \beta m_i^{\mathrm{r}} + \gamma_{\min}) + 1, & i = 1,\ 2, \\ \phi(\alpha m_i^{\mathrm{e}} + \beta m_i^{\mathrm{r}} + \gamma_{\min}), & i = 3,\ 4. \end{cases} \tag{11}$$

Thus, the ZD-VCS strategy $\boldsymbol{p}$ of platform meets $\tilde{\boldsymbol{p}} = \phi(\alpha \boldsymbol{M}^{\mathrm{e}} + \beta \boldsymbol{M}^{\mathrm{r}} + \gamma \boldsymbol{1})(\phi \neq 0)$.

We take the payoff matrices in Fig.4 as an example, here $\boldsymbol{M}^{\mathrm{e}} = (m_1^{\mathrm{e}},\ m_2^{\mathrm{e}},\ m_3^{\mathrm{e}},\ m_4^{\mathrm{e}}) = (30,\ 20,\ 18,\ 16)$ and $\boldsymbol{M}^{\mathrm{r}} = (m_1^{\mathrm{r}},\ m_2^{\mathrm{r}},\ m_3^{\mathrm{r}},\ m_4^{\mathrm{r}}) = (21,\ 28,\ 10,\ 22)$. If we set $\phi = -0.05$ and $\alpha = \beta = 1$, then $\gamma_{\min} = (-51,\ -48,\ -28-20,\ -38-20) = -48$, and $\gamma_{\max} = (-51+20,\ -48+20,\ -28,\ -38) = -28$, therefore $\gamma_{\min} < \gamma_{\max}$ is feasible when $\phi = -0.05$. Note that ϕ is set to guarantee each element in $\boldsymbol{p}$ is in the range of $[0,\ 1]$. According to (11), $p_1 = -0.05(51-48)+1 = 0.85$, $p_2 = -0.05(48-48)+1 = 0$, $p_3 = -0.05(28-48) = 1$, and $p_4 = -0.05(38-48) = 0.5$. Thus, the ZD-VCS strategy is represented as $\boldsymbol{p} = (0.85,\ 0,\ 1,\ 0.5)$. When $\boldsymbol{p}$ is adopted by the platform in the iterated game, the social welfare can achieve to a stable and maximal stable value $-\gamma_{\min} = 48$.

5.3 ZD-VCS Strategy in the Continuous Model

In this subsection, we further analyze the social welfare control problem in the continuous case. Then, we discuss how to obtain the ZD-VCS strategy in the continuous case when we aim to maintain the maximal stable total expected payoff.

To further analyze the social welfare control problem, we consider the ZD-VCS strategy in the continuous case. Based on the number of sensing tasks x_{e} and the candidate available RSUs x_{r}, we fit the continuous payoff functions $F_{\mathrm{g}}(x_{\mathrm{e}},\ x_{\mathrm{r}})$ and $F_{\mathrm{t}}(x_{\mathrm{e}},\ x_{\mathrm{r}})$ for the platform and ISP, respectively, where the fixed x_{e} and x_{r} correspond to the fixed actions. We assume that both the platform and ISP choose their actions according to the outcome of the previous round. Similarly, we define the mixed strategy of the platform $p(x'_{\mathrm{e}},\ x'_{\mathrm{r}},\ x_{\mathrm{e}})$ as the conditional probability to choose the action x_{e} at the current round when the state at the previous round is $x'_{\mathrm{e}}x'_{\mathrm{r}}$, where $x'_{\mathrm{e}},\ x_{\mathrm{e}} \in [l^{\mathrm{e}},\ h^{\mathrm{e}}]$ and $x'_{\mathrm{r}},\ x_{\mathrm{r}} \in [l^{\mathrm{r}},\ h^{\mathrm{r}}]$. Since x_{e} can be any value in the continuous domain in mathematical analysis, we have $\int_{l^{\mathrm{e}}}^{h^{\mathrm{e}}} p(x'_{\mathrm{e}},\ x'_{\mathrm{r}},\ x_{\mathrm{e}})\mathrm{d}x_{\mathrm{e}} = 1$. In addition, the ISP's mixed strategy $q(x'_{\mathrm{e}},\ x'_{\mathrm{r}},\ x_{\mathrm{r}})$ also refers to the conditional probability that the ISP adopts action x_{r} when the state at the previous round is $x'_{\mathrm{e}}x'_{\mathrm{r}}$. The mixed strategy $q(x'_{\mathrm{e}},\ x'_{\mathrm{r}},\ x_{\mathrm{r}})$ satisfies $\int_{l^{\mathrm{r}}}^{h^{\mathrm{r}}} q(x'_{\mathrm{e}},\ x'_{\mathrm{r}},\ x_{\mathrm{r}})\mathrm{d}x_{\mathrm{r}} = 1$.

Next, we denote the joint probability for the platform and ISP to choose x_{e} and x_{r} in each round by $v(x_{\mathrm{e}},\ x_{\mathrm{r}})$. Considering the payoff functions $F_{\mathrm{g}}(x_{\mathrm{e}},\ x_{\mathrm{r}})$ and $F_{\mathrm{t}}(x_{\mathrm{e}},\ x_{\mathrm{r}})$, we obtain the expected utility of the platform and the ISP at the current round as: $U^{\mathrm{e}} = \int_{l^{\mathrm{r}}}^{h^{\mathrm{r}}}\int_{l^{\mathrm{e}}}^{h^{\mathrm{e}}} v(x_{\mathrm{e}},\ x_{\mathrm{r}})F_{\mathrm{g}}(x_{\mathrm{e}},\ x_{\mathrm{r}})\mathrm{d}x_{\mathrm{e}}\mathrm{d}x_{\mathrm{r}}$, and $U^{\mathrm{r}} = \int_{l^{\mathrm{r}}}^{h^{\mathrm{r}}}\int_{l^{\mathrm{e}}}^{h^{\mathrm{e}}} v(x_{\mathrm{e}},\ x_{\mathrm{r}})F_{\mathrm{t}}(x_{\mathrm{e}},\ x_{\mathrm{r}})\mathrm{d}x_{\mathrm{e}}\mathrm{d}x_{\mathrm{r}}$ respectively. Furthermore, similar to the state transition matrix $\boldsymbol{H}$ in the discrete model, we denote a transition function $H(x'_{\mathrm{e}},\ x'_{\mathrm{r}},\ x_{\mathrm{e}},\ x_{\mathrm{r}})$, indicating the state transition probability from the state $x'_{\mathrm{e}}x'_{\mathrm{r}}$ of the previous round to the state $x_{\mathrm{e}}x_{\mathrm{r}}$ of thecurrentround,whichisexpressedas $H(x'_{\mathrm{e}},\ x'_{\mathrm{r}},\ x_{\mathrm{e}},\ x_{\mathrm{r}}) = p(x'_{\mathrm{e}},\ x'_{\mathrm{r}},\ x_{\mathrm{e}})q(x'_{\mathrm{e}},\ x'_{\mathrm{r}},\ x_{\mathrm{r}})$. Then, the relationship of the state probabilities at two sequential rounds is denoted as $v(x'_{\mathrm{e}},\ x'_{\mathrm{r}})H(x'_{\mathrm{e}},\ x'_{\mathrm{r}},\ x_{\mathrm{e}},\ x_{\mathrm{r}}) = v(x_{\mathrm{e}},\ x_{\mathrm{r}})$.

We denote the stationary state as $v^{\mathrm{s}}(x_{\mathrm{e}},\ x_{\mathrm{r}})$. The iterated game reaches a stable state when $v(x'_{\mathrm{e}},\ x'_{\mathrm{r}}) = v(x_{\mathrm{e}},\ x_{\mathrm{r}}) = v^{\mathrm{s}}(x_{\mathrm{e}},\ x_{\mathrm{r}})$. Thus, we have the following theorem.

Theorem 3. *When the platform's strategy* $p(x'_{\mathrm{e}},\ x'_{\mathrm{r}},\ x_{\mathrm{e}})$ *satisfies* $\tilde{p}(x'_{\mathrm{e}},\ x'_{\mathrm{r}},\ h^{\mathrm{e}}) = \phi(\alpha F_{\mathrm{g}}(x_{\mathrm{e}},\ x_{\mathrm{r}}) + \beta F_{\mathrm{t}}(x_{\mathrm{e}},\ x_{\mathrm{r}}) + \gamma)(\phi \neq 0)$, *the platform's expected utility* U^{e} *and the ISP's expected utility* U^{r} *satisfy the following relationship:* $\alpha U^{\mathrm{e}} + \beta U^{\mathrm{r}} + \gamma = 0$, *where the function* $\tilde{p}(x'_{\mathrm{e}},\ x'_{\mathrm{r}},\ h^{\mathrm{e}})$ *is defined as*

$$\tilde{p}(x'_{\mathrm{e}},\ x'_{\mathrm{r}},\ h^{\mathrm{e}}) = \begin{cases} p(x'_{\mathrm{e}},\ x'_{\mathrm{r}},\ h^{\mathrm{e}}), & \text{if } x'_{\mathrm{e}} < h^{\mathrm{e}}, \\ p(x'_{\mathrm{e}},\ x'_{\mathrm{r}},\ h^{\mathrm{e}}) - 1, & \text{if } x'_{\mathrm{e}} = h^{\mathrm{e}}. \end{cases}$$

Next, we discuss how to obtain the function $p(x'_{\mathrm{e}},\ x'_{\mathrm{r}},\ h^{\mathrm{e}})$, which is the continuous ZD-VCS.

The weighted social welfare is defined as $U_{\mathrm{all}} = \alpha U^{\mathrm{e}} + \beta U^{\mathrm{r}} = -\gamma$. According to Theorem 3, the platform's strategy $p(x'_{\mathrm{e}},\ x'_{\mathrm{r}},\ h^{\mathrm{e}})$ is the only factor affecting U_{all}, which is viewed as the ZD-VCS strategy in the continuous case. Specifically, the platform can solve the following optimization problem to achieve unilateral control of social welfare:

$$\begin{aligned} &\max \quad U_{\mathrm{all}} = \alpha U^{\mathrm{e}}(\boldsymbol{p},\boldsymbol{q}) + \beta U^{\mathrm{r}}(\boldsymbol{p},\boldsymbol{q}), \quad \forall \boldsymbol{q}, \\ &\text{s.t.} \begin{cases} \boldsymbol{0} \leqslant \boldsymbol{p} \leqslant \boldsymbol{1}, \\ \alpha U^{\mathrm{e}} + \beta U^{\mathrm{r}} + \gamma = 0. \end{cases} \end{aligned} \tag{12}$$

To simply illustrate the following discussion, we denote $T(x_{\mathrm{e}},\ x_{\mathrm{r}}) = \alpha F_{\mathrm{g}}(x_{\mathrm{e}},\ x_{\mathrm{r}}) + \beta F_{\mathrm{t}}(x_{\mathrm{e}},\ x_{\mathrm{r}})$. Then (12) is converted into

$$\begin{aligned} &\min \quad \gamma \\ &\text{s.t.} \begin{cases} 0 \leqslant p(x'_{\mathrm{e}},\ x'_{\mathrm{r}},\ h^{\mathrm{e}}) \leqslant 1, \\ \tilde{p}(x'_{\mathrm{e}},\ x'_{\mathrm{r}},\ h^{\mathrm{e}}) = \phi(T(x_{\mathrm{e}},\ x_{\mathrm{r}}) + \gamma), \\ \phi \neq 0. \end{cases} \end{aligned}$$

In order to obtain the function $p(x'_{\mathrm{e}},\ x'_{\mathrm{r}},\ h^{\mathrm{e}})$, we divide the discussion by $\phi > 0$ and $\phi < 0$, respectively.

When ϕ is positive and the constraint $p \geqslant 0$, we obtain

$$\begin{aligned} &\gamma_{\min} = \max(\tau(x'_{\mathrm{e}},\ x'_{\mathrm{r}})) \quad \forall x'_{\mathrm{e}} \in [l^{\mathrm{e}},\ h^{\mathrm{e}}], \forall x'_{\mathrm{r}} \in [l^{\mathrm{r}},\ h^{\mathrm{r}}], \\ &\tau(x'_{\mathrm{e}},\ x'_{\mathrm{r}}) = \begin{cases} -T(x'_{\mathrm{e}},\ x'_{\mathrm{r}}), & \text{if } x'_{\mathrm{e}} < h^{\mathrm{e}}, \\ -T(x'_{\mathrm{e}},\ x'_{\mathrm{r}}) - 1/\phi, & \text{if } x'_{\mathrm{e}} = h^{\mathrm{e}}. \end{cases} \end{aligned}$$

While considering the constraint condition $p \leqslant 1$, we obtain

$$\begin{aligned} &\gamma_{\max} = \min(\tau(x''_{\mathrm{e}},\ x''_{\mathrm{r}})), \quad \forall x''_{\mathrm{e}} \in [l^{\mathrm{e}},\ h^{\mathrm{e}}], \forall x''_{\mathrm{r}} \in [l^{\mathrm{r}},\ h^{\mathrm{r}}], \\ &\tau(x''_{\mathrm{e}},\ x''_{\mathrm{r}}) = \begin{cases} 1/\phi - T(x''_{\mathrm{e}},\ x''_{\mathrm{r}}), & \text{if } x''_{\mathrm{e}} < h^{\mathrm{e}}, \\ -T(x''_{\mathrm{e}},\ x''_{\mathrm{r}}), & \text{if } x''_{\mathrm{e}} = h^{\mathrm{e}}. \end{cases} \end{aligned}$$

γ is feasible when $\gamma_{\min} < \gamma_{\max}$. That is $\max(\tau(x'_{\mathrm{e}},\ x'_{\mathrm{r}})) < \min(\tau(x''_{\mathrm{e}},\ x''_{\mathrm{r}})), \forall x'_{\mathrm{e}}, x''_{\mathrm{e}} \in [l^{\mathrm{e}},\ h^{\mathrm{e}}], \forall x'_{\mathrm{r}}, x''_{\mathrm{r}} \in [l^{\mathrm{r}}, h^{\mathrm{r}}]$. Since ϕ is a positive value that normalizes $\boldsymbol{p}$ in the range $[0,\ 1]$, we get the minimum value of γ:

In Press

$$\gamma_{\min}=\max(-T(x'_{\rm e},\ x'_{\rm r})),\ \forall x'_{\rm e}\in[l^{\rm e},\ h^{\rm e}),\ \forall x'_{\rm r}\in[l^{\rm r},\ h^{\rm r}]. \tag{13}$$

When ϕ is negative, considering the constraint $p\geqslant 0$, we obtain $\gamma_{\max}=\min(\tau(x'_{\rm e},\ x'_{\rm r})),\ \forall x'_{\rm e}\in[l^{\rm e},\ h^{\rm e}],\ \forall x'_{\rm r}\in[l^{\rm r},\ h^{\rm r}]$. When considering $p\leqslant 1$, we have $\gamma_{\min}=\max(\tau(x''_{\rm e},\ x''_{\rm r}),\forall x''_{\rm e}\in[l^{\rm e},\ h^{\rm e}],\forall x''_{\rm r}\in[l^{\rm r},\ h^{\rm r}]$. Then, γ is feasible only when $\gamma_{\min}<\gamma_{\max}$, i.e., $\max(\tau(x''_{\rm e},\ x''_{\rm r}))<\min(\tau(x'_{\rm e},\ x'_{\rm r}))$, that is $\max(-T(x''_{\rm e},\ x''_{\rm r}))<\min(-T(x'_{\rm e},\ x'_{\rm r}))$, $\forall x'_{\rm e},x''_{\rm e}\in[l^{\rm e},\ h^{\rm e}],\forall x'_{\rm r},x''_{\rm r}\in[l^{\rm r},\ h^{\rm r}]$. Thus, we have the following result:

$$\begin{aligned}\gamma_{\min} &= \max(\tau(x''_{\rm e},\ x''_{\rm r})) = \max(-T(x''_{\rm e},\ x''_{\rm r})),\\ &\forall x''_{\rm e} = h^{\rm e},\forall x''_{\rm r}\in[l^{\rm r},\ h^{\rm r}].\end{aligned} \tag{14}$$

Therefore, according to (13) and (14), the platform's continuous ZD-VCS strategy is computed as follows:

$$p(x'_{\rm e},\ x'_{\rm r},\ h^{\rm e})=\begin{cases}\phi(T(x'_{\rm e},\ x'_{\rm r})+\gamma_{\min}), & \text{if } x'_{\rm e}<h^{\rm e},\\ \phi(T(x'_{\rm e},\ x'_{\rm r})+\gamma_{\min})+1, & \text{if } x'_{\rm e}=h^{\rm e}.\end{cases} \tag{15}$$

We illustrate how to obtain the wise-function by a simple example. We set $\phi<0$, $\alpha=\beta=1$, $x_{\rm e}\in[1,\ 10]$, $(h^{\rm e}=10)$ and $x_{\rm r}\in[1,\ 10]$, then $F_{\rm g}(x_{\rm e},\ x_{\rm r})=6x_{\rm r}-x_{\rm e}$, and $F_{\rm t}(x_{\rm e},\ x_{\rm r})=3x_{\rm e}-6x_{\rm r}$. According to the previous discussion, we obtain $T(x_{\rm e},\ x_{\rm r})=F_{\rm g}(x_{\rm e},\ x_{\rm r})+F_{\rm t}(x_{\rm e},\ x_{\rm r})=2x_{\rm e}$. In the case of $\phi<0$, $\gamma_{\min}=\max(\tau(x''_{\rm e},\ x''_{\rm r}))=\max(-T(x''_{\rm e},\ x''_{\rm r}))=\max(-2x''_{\rm e})$, where $x''_{\rm e}=h^{\rm e}=10$, therefore $\gamma_{\min}=\max(-2\times 10)=-20$. And $\gamma_{\max}=\min(\tau(x'_{\rm e},\ x'_{\rm r}))=\max(-2x'_{\rm e})$, where $x_{\rm e}\in[1,\ 10]$, therefore $\gamma_{\max}=-2\times 1=-2$. Thus, $\gamma_{\min}<\gamma_{\max}$. According to (15), the ZD-VCS strategy $p(x'_{\rm e},\ x'_{\rm r},\ h^{\rm e})$ in the continuous case is represented as follows:

$$p(x'_{\rm e},\ x'_{\rm r},\ h^{\rm e})=\begin{cases}\phi(2x_{\rm e}-20), & \text{if } x'_{\rm e}\in[1,10),\\ \phi(2x_{\rm e}-20)+1=1, & \text{if } x'_{\rm e}=10.\end{cases}$$

Note that we need to choose proper $\phi<0$ to satisfy $0\leqslant p(x'_{\rm e},\ x'_{\rm r},\ h^{\rm e})\leqslant 1$. From the equation $0\leqslant\phi(2x'_{\rm e}-20)\leqslant 1$,wecanobtain$(2x'_{\rm e}-20)^{-1}\leqslant\phi<0,\ 1(\leqslant x'_{\rm e}<10)$. Therefore, the parameter ϕ is chosen as $(2x'_{\rm e}-20)^{-1}\leqslant\phi<0$ to guarantee $0\leqslant p(x'_{\rm e},\ x'_{\rm r},\ h^{\rm e})\leqslant 1$. In the iterated game, when the platform adopts the pre-calculated $p(x'_{\rm e},\ x'_{\rm r},\ h^{\rm e})$ in the continuous case, the social welfare can be unilaterally controlled at a maximal and stable value $-\gamma_{\min}=20$ in the long term.

In sum, in the continuous case, the ZD-VCS strategy adopted by the platform is a piece-wise function, as shown in (15). No matter what the strategy of ISP is, the social welfare can be unilaterally maintained at the value $-\gamma_{\min}$.

5.4 Extension: Extortion Strategy

Based on the equation (6), a special kind of zero-determinant strategy, extortion strategy, is derived. Below we discuss how the extortion strategy is derived, and how it is used in the iterated vehicular crowdsensing game.

When the platform is in a dominant role and wants to control its and its opponent's payoffs at a predefined ratio in some scenarios, for the purpose of unilateral control, we can help the platform to extortionately control the payoff of the opponent at a low value. Thus, we propose an extortion strategy in a discrete model for the platform, which is a kind of zero-determinant strategy with special parameter settings, and the details are as follows.

In the case that the platform attempts to enforce an extortionate share of payoffs larger than the mutual noncooperation value. Based on (6), when $\tilde{\boldsymbol{p}}=\phi((\boldsymbol{M}^{\rm e}-m_4^{\rm e}\boldsymbol{1})-\chi(\boldsymbol{M}^{\rm r}-m_4^{\rm r}\boldsymbol{1}))$, $\alpha=\phi$, and $\beta=-\phi\chi$ $(\phi\neq 0)$, that is $\chi=-\dfrac{\beta}{\alpha}$, $\gamma=\phi(\chi m_4^{\rm r}-m_4^{\rm e})$. Then the linear relationship between the expected payoff of the platform and that of the ISP is obtained, and $(U^{\rm e}-m_4^{\rm e})=\chi(U^{\rm r}-m_4^{\rm r})$, where $\chi\geqslant 1$ is called the extortion ratio and ϕ is employed to guarantee that each element in $\boldsymbol{p}$ is at the range of $[0,\ 1]$. The specific extortion strategy of the platform can be obtained by solving the following equations:

$$\begin{cases}p_1=1+\phi((m_1^{\rm e}-m_4^{\rm e})-\chi(m_1^{\rm r}-m_4^{\rm r})),\\ p_2=1+\phi((m_2^{\rm e}-m_4^{\rm e})-\chi(m_2^{\rm r}-m_4^{\rm r})),\\ p_3=\phi((m_3^{\rm e}-m_4^{\rm e})-\chi(m_3^{\rm r}-m_4^{\rm r})),\\ p_4=0.\end{cases}$$

Obviously, the feasible solution exists and is determined by any χ and a sufficiently small ϕ. Specifically, to ensure that each element of $\boldsymbol{p}$ belongs to $[0,\ 1]$, ϕ should satisfy: $\phi\in(0,\ \tilde{\phi}_1]$ when $\phi>0$, and $\phi\in[\tilde{\phi}_2,\ 0)$ when $\phi<0$, where $\tilde{\phi}_1=\min\{\pi_1,\ \pi_2,\ \pi_3\}$, $\tilde{\phi}_2=\max\{\pi_1,\ \pi_2,\ \pi_3\}$. For convenience, we set

$$\pi_1=\frac{-1}{(m_1^{\rm e}-m_4^{\rm e})-\chi(m_1^{\rm r}-m_4^{\rm r})}$$

$$\pi_2=\frac{-1}{(m_2^{\rm e}-m_4^{\rm e})-\chi(m_2^{\rm r}-m_4^{\rm r})}$$

$$\pi_3=\frac{1}{(m_3^{\rm e}-m_4^{\rm e})-\chi(m_3^{\rm r}-m_4^{\rm r})}$$

When ISP always takes the action of defection, that is $\boldsymbol{q}=(0,\ 0,\ 0,\ 0)$, the minimum utility of ISP is $m_4^{\rm r}$. While when ISP unconditionally cooperates, that is $\boldsymbol{q}=(1,\ 1,\ 1,\ 1)$. According to (5), the maximum utility of ISP can be calculated as

$$\min\ U^{\mathrm{r}}|_{\boldsymbol{q}=(1,\ 1,\ 1,\ 1)} = \frac{D(\boldsymbol{p},\ \boldsymbol{q},\ \boldsymbol{M}^{\mathrm{r}})}{D(\boldsymbol{p},\ \boldsymbol{q},\ \boldsymbol{1})}.$$

Furthermore, when $\chi = 1$, that means there is no extortion of the platform to ISP, and thus the maximum utility of the platform can be obtained by (5), and be calculated as follows:

$$\max\ U^{\mathrm{r}}|_{\boldsymbol{q}=(1,\ 1,\ 1,\ 1),\ \chi=1} = \frac{D(\boldsymbol{p},\ \boldsymbol{q},\ \boldsymbol{M}^{\mathrm{r}})}{D(\boldsymbol{p},\ \boldsymbol{q},\ \boldsymbol{1})}.$$

6 Performance Evaluation

In this section, we first introduce the experimental setup. Then, we present the experimental results.

6.1 Experimental Setup

We perform experiments on the Taxi-Roma dataset⑤. The dataset includes the GPS coordinates of 320 taxi drivers that work in the center of Rome. These GPS coordinates are collected over 30 days. The dataset is preprocessed by filtering out some outliers. We implement the proposed strategy with real trace-driven simulations. The communication in the physical layer is assumed to be stable and reliable[24]. The traces cover an area with a range of 66 km × 59 km, and we divide the area into 10×10 grids. We assume each RSU in the grid serves as a gateway and each grid is viewed as an area with a sensing task. We calculate the priority of the sensing tasks and RSUs by the number of GPS coordinates in the grid, and both top x_{e} sensing tasks and top x_{r} RSUs are selected by their priorities. From the analysis of traces, we obtain the average task-RSU weighted traffic matrix. Each element in the matrix is the number of sensing data uploaded to an RSU for a task. In the discrete model, we set the lowest and the highest numbers of sensing tasks and RSUs are 2 and 8, respectively. That is $l^{\mathrm{e}} = l^{\mathrm{r}} = 2$, and $h^{\mathrm{e}} = h^{\mathrm{r}} = 8$. The parameters are set as $u_{\mathrm{d}} = 30$, $u_{\mathrm{t}} = 1$, $u_e = 0.01$, $u_r = 50$, and $C = 500$. We discuss how to implement the ZD-VCS strategy in Subsection 6.5 when task-RSU weighted traffic matrix varies on each day. We implement the ZD-VCS strategy based schemes in Python 3.8. All experiments were conducted on a computer with Intel Core i7-6700 CPU and 8G RAM.

6.2 Comparison Methods

We describe the following baselines for comparison.

1) *ALLC*[12, 14, 29]. It is an all cooperation strategy $\boldsymbol{p} = (1,\ 1,\ 1,\ 1)$, which means whatever the opponent player has done in the previous round, and it always chooses cooperation.

2) *ALLD*[12, 29]. It is an all defection strategy $\boldsymbol{p} = (0,\ 0,\ 0,\ 0)$, which means whatever the opponent player has done in the previous round, and it always chooses defection.

3) *Random*[12, 14, 29]. It is an all-random strategy, that is $\boldsymbol{p} = (0.5,\ 0.5,\ 0.5,\ 0.5)$, which means whatever the opponent player has done in the previous round, it randomly chooses cooperation.

4) *TFT* (*tit-for-tat*)[46, 47]. A TFT player cooperates in the first round and then does whatever the opponent player has done in the previous round.

5) *Evolved*[15, 23]. It is an evolved strategy that an evolutionary player starts his/her strategy $\boldsymbol{q} = (0,\ 0,\ 0,\ 0)$, and his/her opponent adopts ZD-VCS strategy in the iterated games. $\boldsymbol{q}$ is updated in each iteration and will be stable at the end of the iterations. The process of calculating evolved strategy is shown in Algorithm 2.

6) *Pavlov* (*win-stay-lose-shift strategy*)[47, 48]. If a Pavlov player receives a higher payoff, it will repeat the same action in the next round, which is "win-stay". If a Pavlov player receives a lower payoff, it will switch to the opposite action, which is "lose-shift".

6.3 Performance Evaluation on Social Welfare Control

In this subsection, we evaluate the performance of the proposed ZD-VCS on social welfare control in the discrete and continuous model.

6.3.1 Experiments in the Discrete Model

Algorithms 1 and 2 together are the processes of calculating the utilities of the platform and ISP in the experiment. Considering the general definition of social welfare in (7), we set $\alpha = \beta = 1$ in social welfare control. In order to evaluate the effectiveness of our proposed scheme, we compare the ZD-VCS strategy with five other classical strategies that might be adopted by the platform. We display the results when the platform adopts the proposed ZD-VCS strategy, all cooperation ($\boldsymbol{p} = (1,\ 1,\ 1,\ 1)$, denoted as ALLC), all defection ($\boldsymbol{p} = (0,\ 0,\ 0,\ 0)$, denoted as ALLD), and random ($\boldsymbol{p} = (0.5,\ 0.5,\ 0.5,\ 0.5)$, denoted as Random) strategies while the ISP adopts three strategies, i.e., ALLC, ALLD, and Random. Furthermore, we also adopt an evolved strategy with ISP. We view ISP as an evolutionary player that adopts an evolved strategy. As shown in Fig.5, when the platform adopts the proposed ZD-VCS strategy, whatever the ISP adopts, the social welfare can maintain stable and achieve its possible maximal value. However, when the platform takes the other strategies, the social welfare is determined by the strategies of both the platform and the ISP, which indicates that the platform does not dominate the control over the social welfare. In Fig.5(b), when the platform adopts ALLC strategy, and the ISP adopts ALLC or Random strategy, the social welfare keeps stable in the long term. But, the platform adopted ALLC cannot control social welfare when ISP adopts different strategies.

Algorithm 1. ZD-VCS Strategy in Discrete Model

Input: $\boldsymbol{p}$, $\boldsymbol{q}$, $x_{\rm r}$, previous state $x'_{\rm e}$, $x'_{\rm r}$, $\boldsymbol{M}^{\rm e}$, number of iterations n

Output: $U^{\rm e}$, $x'_{\rm e}$

for $i = 1; i \leqslant n; i++$ **do**
  if $\text{random}() \leqslant \boldsymbol{p}(h^{\rm e}|x'_{\rm e}x'_{\rm r}) : x_{\rm e} = h^{\rm e}$;
  else: $x_{\rm e} = l^{\rm e}$;
  $x'_{\rm e} = x_{\rm e}$;
  $U^{\rm e}_i = \boldsymbol{M}^{\rm e}(x_{\rm e},\ x_{\rm r})$;
  return $(\sum_{i=1}^{n} U^{\rm e}_i)/n,\ x'_{\rm e}$;

Fig.6 shows the comparison results between the ZD-VCS strategy and two other classical strategies, i.e., TFT[46] and Pavlov[48]. The experiment starts from a random state of the game. It is obvious that the platform adopts the ZD-VCS strategy and ISP takes either TFT or Pavlov, and the social welfare is approximately the same and is stable. However, when the platform changes its strategy to any other strategies, the social welfare is not dominated by the platform and is affected by the strategies of both players.

Algorithm 2. Evolved Strategy by ISP

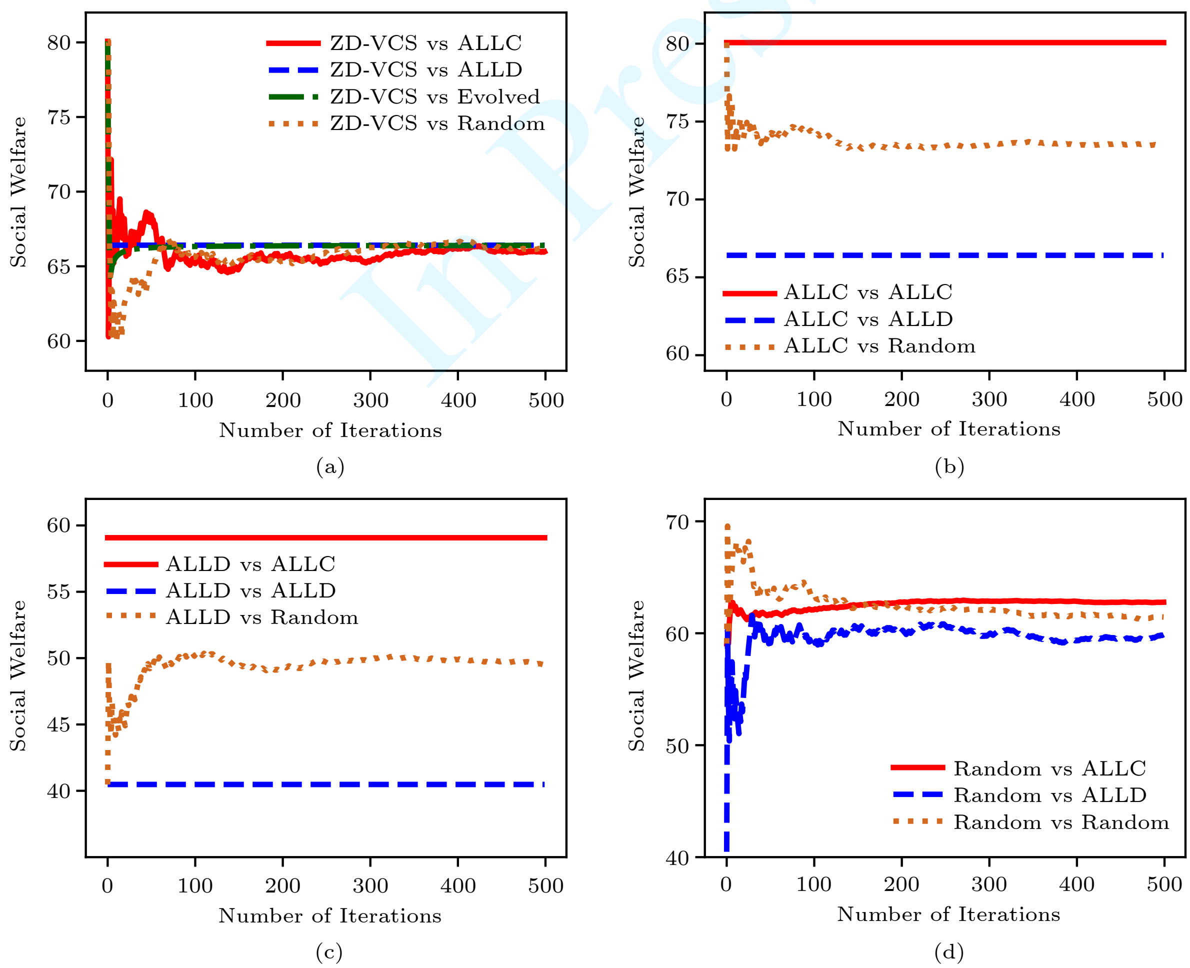
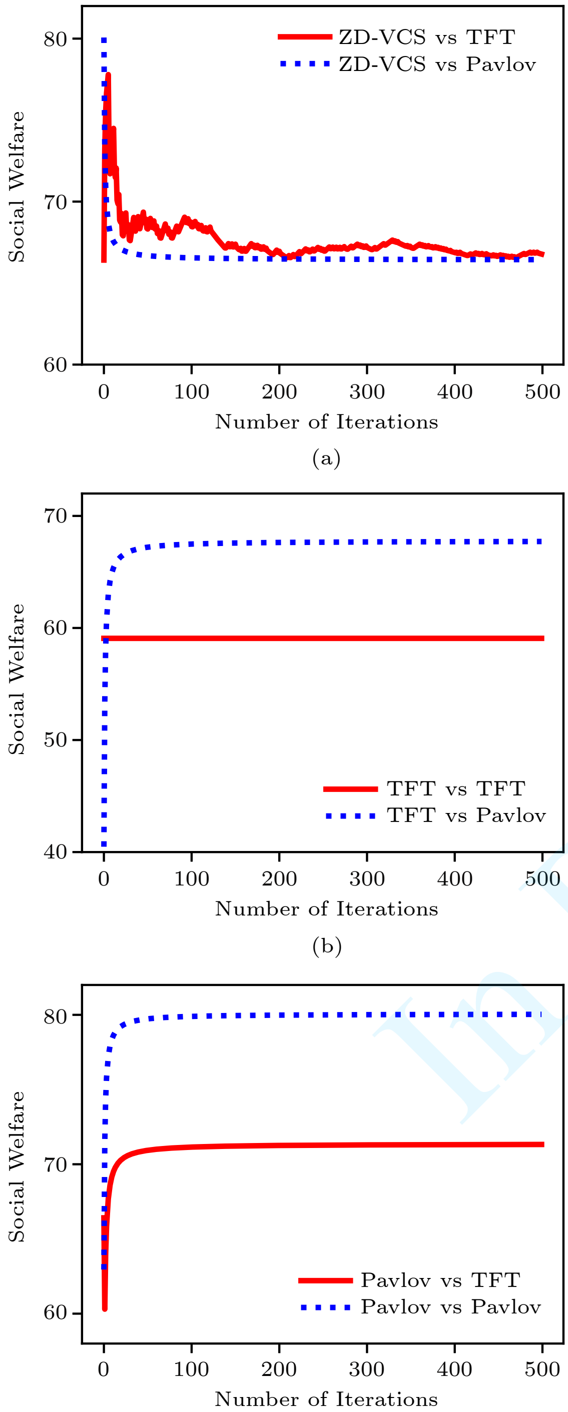


Fig.5. Social welfare with different strategy pairs (Platform vs ISP) in the discrete model. (a) ZD-VCS vs ALLC/ALLD /Evolved/ Random. (b) ALLC vs ALLC/ALLD/Random. (c) ALLD vs ALLC/ALLD/Random. (d) Random vs ALLC/ALLD/ Random.

In Press



(c)

Fig.6. Social welfare with different strategy pairs (Platform vs ISP) in the discrete model. (a) ZD-VCS vs TFT/Pavlov. (b) TFT vs TFT/Pavlov. (c) Pavlov vs TFT/Pavlov.

Input: $\boldsymbol{p}$, $\boldsymbol{q}$, $x_{\rm e}$, previous state $x'_{\rm e}$, $x'_{\rm r}$, $\boldsymbol{M}^t$, learning rate θ, number of iterations n

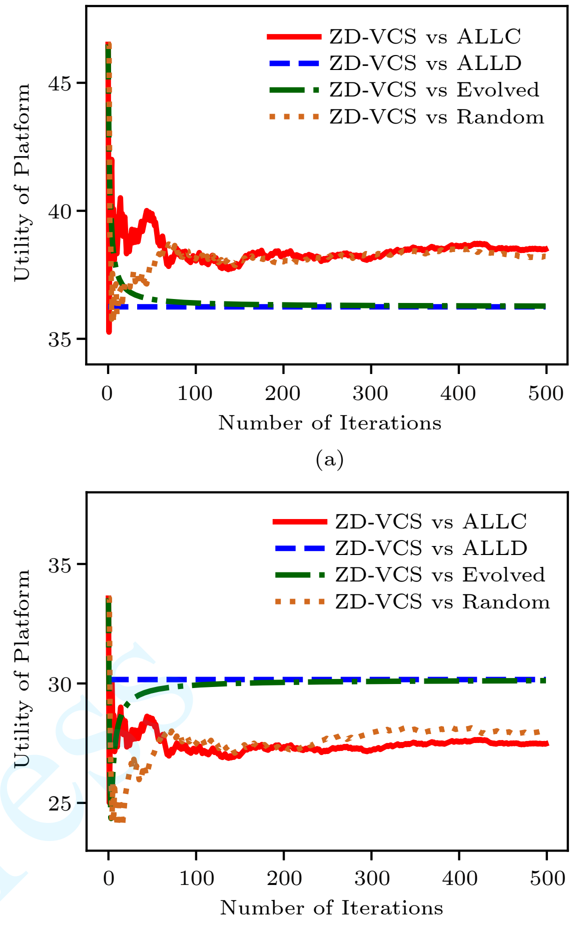
Output: $U^{\rm e}$, $\boldsymbol{q}$, $x'_{\rm r}$

for $i = 1; i \leqslant n; i++$ **do**

  if $\text{random}() \leqslant \boldsymbol{q}(h^{\rm r}|x'_{\rm e}x'_{\rm r}) : x_{\rm r} = h^{\rm r}$;

  else: $x_{\rm r} = l^{\rm r}$;

  $x'_{\rm r} = x_{\rm r}$;



(b)

Fig.7. Utility of Platform and ISP when platform adopts the ZD-VCS strategy, and ISP adopts ALLC/ALLD/Evolved/Random strategy in the discrete model. (a) Utility of platform. (b) Utility of ISP.

  $U^{\rm r} = D(\boldsymbol{p}, \boldsymbol{q}, \boldsymbol{M}^{\rm r})/D(\boldsymbol{p}, \boldsymbol{q}, \boldsymbol{1})$,;

  $U^{\rm r}_i = \boldsymbol{M}^{\rm r}(x_{\rm e},\ x_{\rm r})$;

  for each element $q_i \in \boldsymbol{q}$ **do**

$$q_i = q_i + \theta \frac{\partial U^{\rm r}(\boldsymbol{q})}{\partial q_i}$$

  return $(\sum_{i=1}^{n} U^{\rm r}_i)/n$, $\boldsymbol{q}$, $x'_{\rm r}$;

Fig.7 and Fig.8 show the respective utilities of the platform and ISP. The results are shown in Fig.7 when the platform adopts the ZD-VCS strategy, and the ISP takes Evolved, ALLC, ALLD, and Random strategies. Fig.7(a) and Fig.7(b) show the average utility for the platform and ISP of all current iterations, respectively. From the results, we can see that the utilities of the platform and ISP are becoming stable as the number of iterations increases. The corresponding total utility (i.e., social welfare) is shown in Fig.5(a). Similarly, Fig.8 shows the results when the platform adopts ZD-VCS strategy and ISP takes TFT, and Pavlov strategies. Fig.8(a) and Fig.8(b)

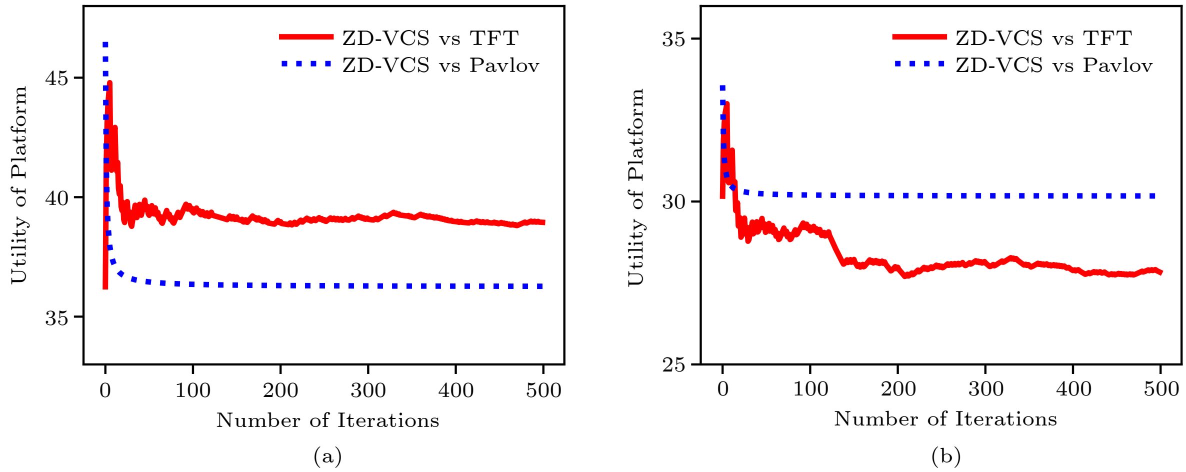


Fig.8. Utility of Platform and ISP when platform adopts the ZD-VCS strategy, and ISP adopts TFT/Pavlov strategy in the discrete model. (a) Utility of platform. (b) Utility of ISP.

show the mean utility for the platform and ISP of all current iterations, respectively. From the results, we can find that the utilities of the platform and ISP gradually become stable as the number of iterations increases. Its corresponding total utility is shown in Fig.6(a). From Fig.7 and Fig.8, we can find the utility of the platform in a stable state is larger than that of the ISP in all strategy pairs.

6.3.2 *Experiments in the Continuous Model*

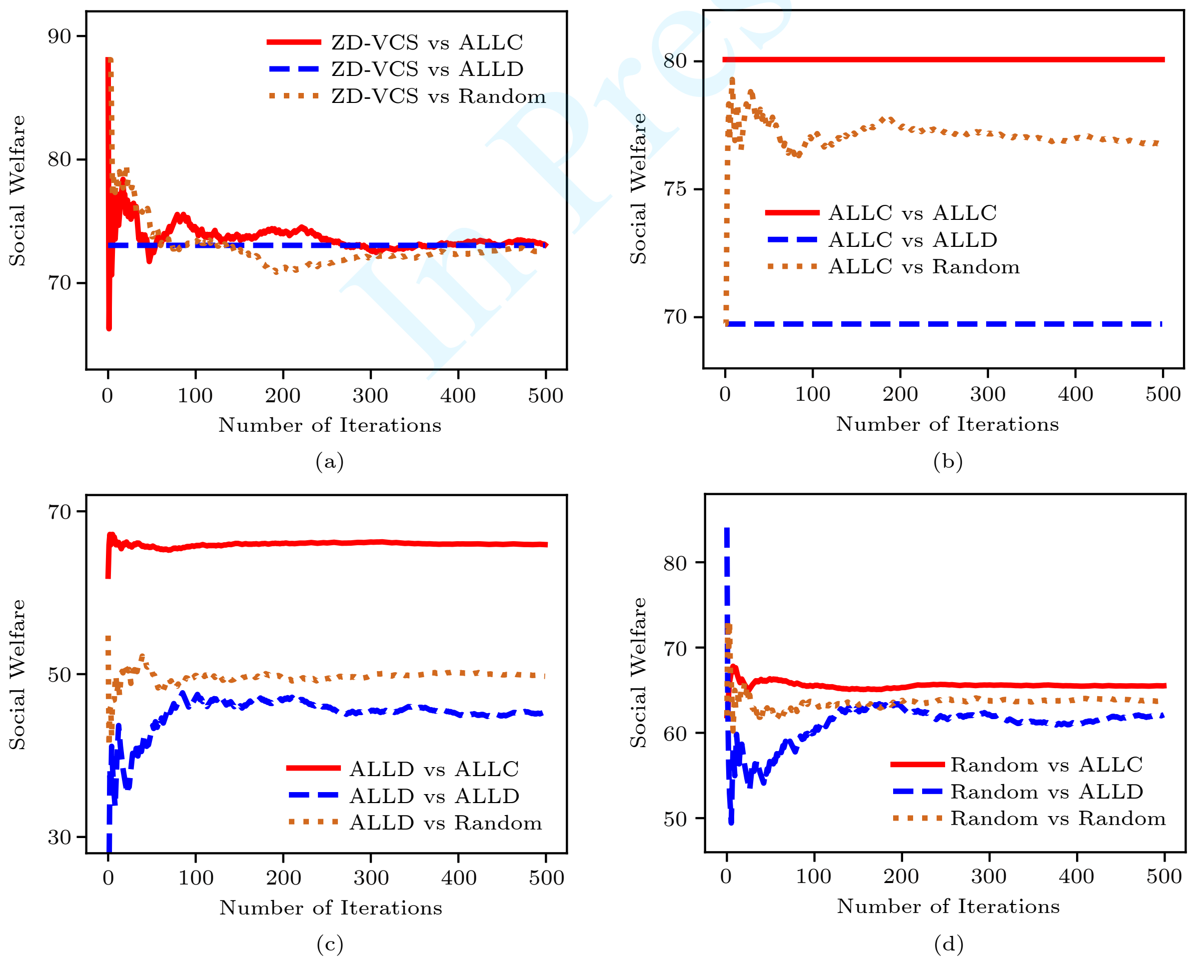


Fig.9. Social welfare with different strategy pairs (platform vs ISP) in the continuous model. (a) ZD-VCS vs ALLC/ALLD/Random. (b) ALLC vs ALLC/ALLD/Random. (c) ALLD vs ALLC/ALLD/Random. (d) Random vs ALLC/ALLD/Random.

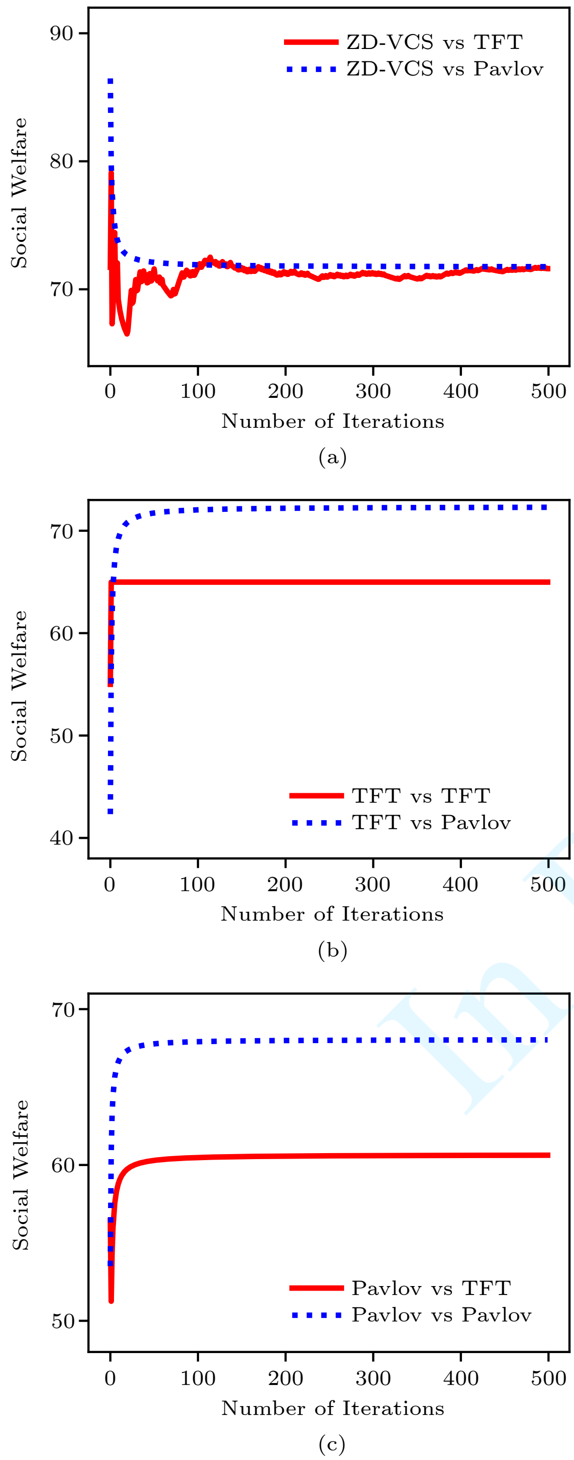


Fig.10. Social welfare with different strategy pairs (Platform vs ISP) in the continuous model. (a) ZD-VCS vs TFT/Pavlov. (b) TFT vs TFT/Pavlov. (c) Pavlov vs TFT/ Pavlov.

In the continuous model, we adopt continuous strategies for the platform and ISP. We assume that both the number of sensing tasks and available RSUs are selected from [2, 8]. We first compare the social welfare when the platform adopt the ZD-VCS strategy and the normal strategies, such as the ALLC strategy $p(x'_{\rm e},\ x'_{\rm r},\ h^{\rm e}) = 1$, the ALLD strategy $p(x'_{\rm e},\ x'_{\rm r},\ h^{\rm e}) = 0$, and the Random strategy $p(x'_{\rm e},\ x'_{\rm r},\ h^{\rm e}) = 1/(h^{\rm e} - l^{\rm e})$. As shown in Fig.9(a), when the platform adopts ZD-VCS strategy, the social welfare becomes stable regardless of the ISP's strategy (the ALLC strategy $q(x'_{\rm e},\ x'_{\rm r},\ h^{\rm r}) = 1$, the ALLD strategy $q(x'_{\rm e},\ x'_{\rm r},\ h^{\rm r}) = 0$, or the Random strategy $q(x'_{\rm e},\ x'_{\rm r},\ h^{\rm r}) = 1/(h^{\rm r} - l^{\rm r})$). However, in Fig.9(b), Fig.9(c), and Fig.9(d), when the platform adopts ALLC, ALLD, and Random strategies, it cannot control the social welfare.

Fig.10(a) shows that social welfare stays stable when the platform adopts ZD-VCS strategy and ISP adopts TFT and Pavlov strategies. When the platform adopts TFT and Pavlov strategies, it cannot control social welfare in Fig.10(b) and Fig.10(c). In Fig.9(a) and Fig.10(a), the stable social welfare is slightly higher than that in the discrete model. Fig.11(a) and Fig.11(b) show the utility of the platform. When it adopts ZD-VCS strategy and ISP adopts ALLC, ALLD, and Random strategy, respectively, and the platform's utility is higher than that of the ISP on the whole. Fig.12(a) and Fig.12(b) show the utility of the platform adopts ZD-VCS strategy and ISP adopts TFT and Pavlov strategies, respectively. Similarly, the platform's utility is higher than that of the ISP.

In Press

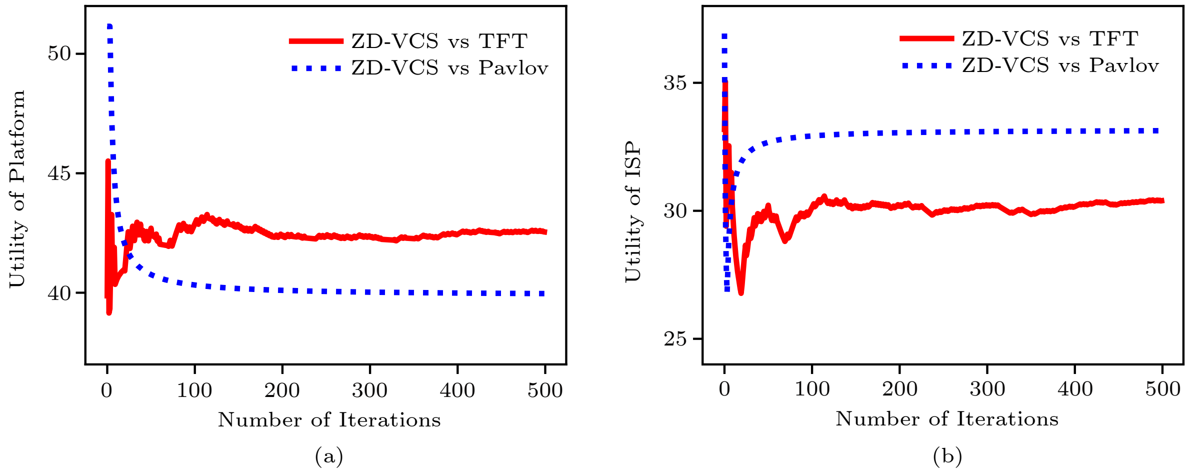


Fig.12. Utility of Platform and ISP when platform adopts the ZD-VCS strategy, and ISP adopts TFT/Pavlov strategy in the continuous model. (a) Utility of platform. (b) Utility of ISP.

In Press

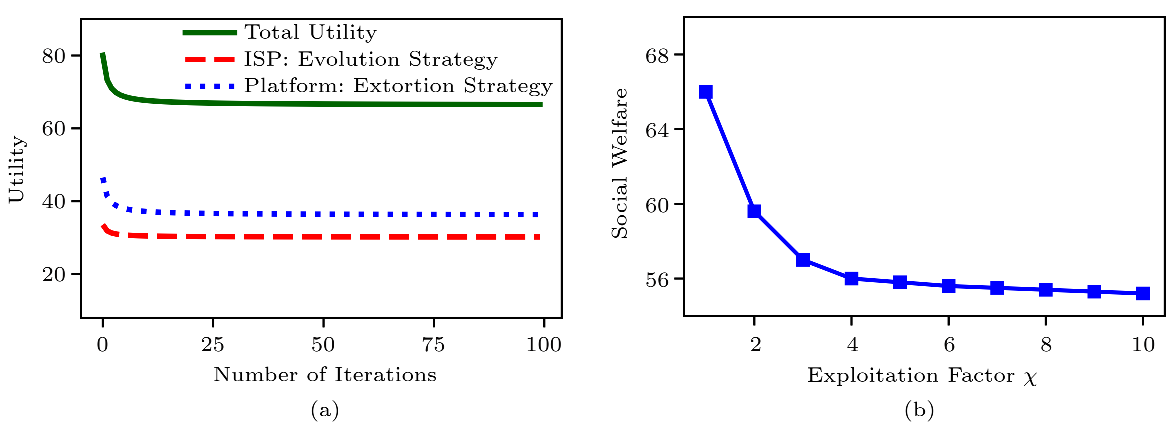


Fig.13. Utility and social welfare when the platform adopts an extortion strategy in the discrete model. (a) Utility of platform and ISP in different numbers of iterations when extortion factor $\chi = 2$. (b) Social welfare under different extortion factors.

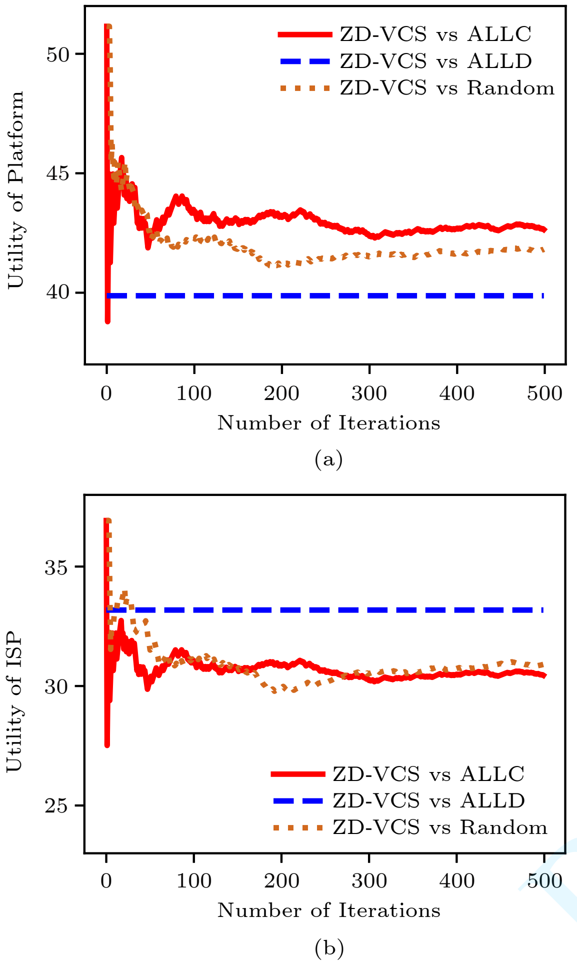


Fig.11. Utility of platform and ISP when the platform adopts the ZD-VCS strategy, and ISP adopts the ALLC/ALLD/Random strategy in the continuous model. (a) Utility of platform. (b) Utility of ISP.

6.4 Performance of the Extortion Strategy

To evaluate the influence of social welfare when the platform adopts an extortion strategy, we do experiments in the iterated game with the platform adopting an extortion strategy and the ISP adopting an evolution strategy. Specifically, we set the extortion factor $\chi = 2$, and the results are shown in Fig.13(a). The red dot line refers to the utility of ISP adopting an evolution strategy, and the blue line refers to the utility of the platform adopting an extortion strategy. The platform's utility is approximately two times more than the ISP's utility when both are minus a constant value. The solid green line represents the total utility (i.e., social welfare) of the platform and ISP, which is becoming stable as the number of iteration increases. However, the stable value is less than that of when the platform adopts ZD-VCS strategy. Furthermore, we discuss the influence of different extortion factors on social welfare. Fig.13(b) shows that social welfare first decreases sharply and then slowly as the extortion factor increases. When the extortion factor $\chi = 1$, we set $\alpha = \beta = 1$, which means the platform has no extortion to ISP. Therefore the extortion strategy turns to a ZD-VCS strategy used in Subsection 5.2, and the social welfare is the same as the stable value in Fig.5(a) and Fig.6(a).

6.5 Discussion

In some scenarios, each day's traffic varies differently, therefore the payoff matrices vary each day. Therefore, how to use ZD-VCS strategy in practice is an issue. Usually, the traffic in a city or an area obeys a regular distribution in the periods, therefore we can predict each day's traffic according to the traffic histories (such as fitting a periodic function for prediction). Then, the predicted payoff matrix of each day could be calculated. Therefore, we can pre-calculate the ZD-VCS strategy adopted by the platform each day when the action space is discrete. Thus, the platform controls the social welfare with a high and stable value in the long term, which is shown in Fig.5(a) and Fig.6(a). Furthermore, when the platform and the ISP's utilities are represented by continuous functions, the parameters of tasks and RSUs are continuous (parameters only make sense if they are integers), and then the ZD-VCS strategy is represented by a piece-wise function. The platform also controls the social welfare with a high and stable value, which is shown in Fig.9(a) and Fig.10(a).

In short, the utility function is appropriately represented by the utility matrix or the utility function. When an action space is discrete or continuous, the ZD-VCS strategy can be pre-calculated directly or represented by a piece-wise function. Then, the platform can adopt ZD-VCS strategy and control the social welfare without considering the ISP's strategy.

7 Conclusion

In this paper, we formulated the interaction between the platform and ISP under vehicular crowdsensing as an iterated game, and addressed the problem of social welfare control in this game. We proposed a zero-determinant strategy for the vehicular crowdsensing platform strategy to control the social welfare without considering ISP's strategy. We theoretically analyzed that the platform can achieve stable and possible maximal social welfare regardless of the ISP's strategy. Additionally, we investigated the influence of the extortion strategy on social welfare. Experimental results verify that the platform using the ZD-VCS strategy unilaterally controls the social welfare. In the future, we will study the different variants of the zero-determinant strategy and apply them in other scenarios.

Conflict of Interest The authors declare that they have no conflict of interest.

References

[1] Song C, Wu J. Vehicular ad hoc/sensor networks in smart cities. In *Mission-Oriented Sensor Networks and Systems*: *Art and Science*: *Volume 2*: *Advances*, Ammari H M (ed.), Springer, 2019, pp.287–310. DOI: 10.1007/978-3-319-92384-0_9.

[2] Eriksson J, Girod L, Hull B, Newton R, Madden S, Balakrishnan H. The pothole patrol: Using a mobile sensor network for road surface monitoring. In *Proc. the 6th International Conference on Mobile Systems*, *Applications*, *and Services*, Jun. 2008, pp.29–39. DOI: 10.1145/1378600.1378605.

[3] Ganti R K, Ye F, Lei H. Mobile crowdsensing: Current state and future challenges. *IEEE Communications Magazine*, 2011, 49(11): 32–39. DOI: 10.1109/MCOM.2011.6069707.

[4] Guo B, Chen H, Yu Z, Xie X, Huangfu S, Zhang D. FlierMeet: A mobile crowdsensing system for cross-space public information reposting, tagging, and sharing. *IEEE Trans. Mobile Computing*, 2015, 14(10): 2020–2033. DOI: 10.1109/TMC.2014.2385097.

[5] Campioni F, Choudhury S, Salomaa K, Akl S G. Improved recruitment algorithms for vehicular crowdsensing networks. *IEEE Trans. Vehicular Technology*, 2019, 68(2): 1198–1207. DOI: 10.1109/TVT.2018.2885403.

[6] Ding R, Yang Z, Wei Y, Jin H, Wang X. Multi-agent reinforcement learning for urban crowd sensing with for-hire vehicles. In *Proc. the 2021 IEEE Conference on Computer Communications*, May 2021, pp.1–10. DOI: 10.1109/INFOCOM42981.2021.9488713.

[7] Nash J F Jr. Equilibrium points in *n*-person games. *Proceedings of the National Academy of Sciences of the United States of America*, 1950, 36(1): 48–49. DOI: 10.1073/pnas.36.1.48.

[8] Wang E, Yang Y, Wu J, Wang H. Multi-round bidding strategy based on game theory for crowdsensing task. In *Proc. the 12th International Conference on Security*, *Privacy*, *and Anonymity in Computation*, *Communication*, *and Storage*, Jul. 2019, pp.196–210. DOI: 10.1007/978-3-030-24907-6_16.

[9] Liang C, He Y, Yu F R, Zhao N. Energy-efficient resource allocation in software-defined mobile networks with mobile edge computing and caching. In *Proc. the 2017 IEEE Conference on Computer Communications Workshops*, May 2017, pp.121–126. DOI: 10.1109/INFCOMW.2017.8116363.

[10] Okic A, Zanzi L, Sciancalepore V, Redondi A, Costa-Pérez X. π-ROAD: A learn-as-you-go framework for on-demand emergency slices in V2X scenarios. In *Proc. the 2021 IEEE Conference on Computer Communications*, May 2021, pp.1–10. DOI: 10.1109/INFOCOM42981.2021.9488677.

[11] Ozkaptan C D, Ekici E, Altintas O. Enabling communication via automotive radars: An adaptive joint waveform design approach. In *Proc. the 2020 IEEE Conference on Computer Communications*, Jul. 2020, pp.1409–1418. DOI: 10.1109/INFOCOM41043.2020.9155527.

[12] Hu Q, Wang S, Bie R, Cheng X. Social welfare control in mobile crowdsensing using zero-determinant strategy. *Sensors*, 2017, 17(5): 1012. DOI: 10.3390/s17051012.

[13] Hu Q, Wang S, Ma L, Bie R, Cheng X. Anti-malicious crowdsourcing using the zero-determinant strategy. In *Proc. the 37th IEEE International Conference on Distributed Computing Systems*, Jun. 2017, pp.1137–1146. DOI: 10.1109/ICDCS.2017.46.

[14] Tang C, Li X, Cao M, Zhang Z, Yu X. Incentive mechanism for macrotasking crowdsourcing: A zero-determinant strategy approach. *IEEE Internet of Things Journal*, 2019, 6(5): 8589–8601. DOI: 10.1109/JIOT.2019.2921348.

[15] Hu Q, Wang S, Ma P, Cheng X, Lv W, Bie R. Quality control in crowdsourcing using sequential zero-determinant strategies. *IEEE Trans. Knowledge and Data Engineering*, 2020, 32(5): 998–1009. DOI: 10.1109/TKDE.2019.2896926.

[16] Hu Q, Wang S, Cheng X. A game theoretic analysis on block withholding attacks using the zero-determinant strategy. In *Proc. the 2019 International Symposium on Quality of Service*, Jun. 2019, Article No.41. DOI: 10.1145/3326285.3329076.

[17] Zhang H, Niyato D, Song L, Jiang T, Han Z. Equilibrium analysis for zero-determinant strategy in resource management of wireless network. In *Proc. the 2015 IEEE Wireless Communications and Networking Conference*, Mar. 2015, pp.2002–2007. DOI: 10.1109/WCNC.2015.7127775.

[18] Chakeri A, Wang X, Goss Q, Akbas M I, Jaimes L G. A platform-based incentive mechanism for autonomous vehicle crowdsensing. *IEEE Open Journal of Intelligent Transportation Systems*, 2021, 2: 13–23. DOI: 10.1109/OJITS.2021.3056925.

[19] Di Stefano A, Scatá M, Attanasio B, La Corte A, Lió P, Das S K. A novel methodology for designing policies in mobile crowdsensing systems. *Pervasive and Mobile Computing*, 2020, 67: 101230. DOI: 10.1016/j.pmcj.2020.101230.

[20] Xiao L, Chen T, Xie C, Dai H, Poor H V. Mobile crowdsensing games in vehicular networks. *IEEE Trans. Vehicular Technology*, 2018, 67(2): 1535–1545. DOI: 10.1109/TVT.2016.2647624.

[21] Dai Z, Wang H, Liu C H, Han R, Tang J, Wang G. Mobile crowdsensing for data freshness: A deep reinforcement learning approach. In *Proc. the 2021 IEEE Conference on Computer Communications*, May 2021, pp.1–10. DOI: 10.1109/INFOCOM42981.2021.9488791.

[22] Keremoğlu E, Weidmann N B. How dictators control the

In Press

internet: A review essay. *Comparative Political Studies*, 2020, 53(10/11): 1690–1703. DOI: 10.1177/0010414020912278.

[23] Press W H, Dyson F J. Iterated prisoner's dilemma contains strategies that dominate any evolutionary opponent. *Proceedings of the National Academy of Sciences of the United States of America*, 2012, 109(26): 10409–10413. DOI: 10.1073/pnas.1206569109.

[24] Pu L, Chen X, Mao G, Xie Q, Xu J. *Chimera*: An energy-efficient and deadline-aware hybrid edge computing framework for vehicular crowdsensing applications. *IEEE Internet of Things Journal*, 2019, 6(1): 84–99. DOI: 10.1109/JIOT.2018.2872436.

[25] Morselli F, Zabini F, Conti A. Environmental monitoring via vehicular crowdsensing. In *Proc. the 29th IEEE Annual International Symposium on Personal, Indoor and Mobile Radio Communications*, Sept. 2018, pp.1382–1387. DOI: 10.1109/PIMRC.2018.8580783.

[26] Stewart A J, Plotkin J B. From extortion to generosity, evolution in the iterated prisoner's dilemma. *Proceedings of the National Academy of Sciences of the United States of America*, 2013, 110(38): 15348–15353. DOI: 10.1073/pnas.1306246110.

[27] Roemheld L. Evolutionary extortion and mischief zero determinant strategies in iterated 2x2 games[Bachelor thesis]. Heidelberg University, Heidelberg, 2013. **（查阅网上资料, 无法确认论文属性是否正确, 请确认）**

[28] Hu Q, Wang S, Cheng X, Ma L, Bie R. Solving the crowdsourcing dilemma using the zero-determinant strategies. *IEEE Trans. Information Forensics and Security*, 2020, 15: 1778–1789. DOI: 10.1109/TIFS.2019.2949440.

[29] Zhang H, Niyato D, Song L, Jiang T, Han Z. Zero-determinant strategy for resource sharing in wireless cooperations. *IEEE Trans. Wireless Communications*, 2016, 15(3): 2179–2192. DOI: 10.1109/TWC.2015.2499185.

[30] He X, Dai H, Ning P, Dutta R. Zero-determinant strategies for multi-player multi-action iterated games. *IEEE Signal Processing Letters*, 2016, 23(3): 311–315. DOI: 10.1109/LSP.2016.2517640.

[31] McAvoy A, Hauert C. Autocratic strategies for iterated games with arbitrary action spaces. *Proceedings of the National Academy of Sciences of the United States of America*, 2016, 113(13): 3573–3578. DOI: 10.1073/pnas.1520163113.

[32] Tan R, Su Q, Wu B, Wang L. Payoff control in repeated games. In *Proc. the 33rd Chinese Control and Decision Conference*, May 2021, pp.997–1005. DOI: 10.1109/CCDC52312.2021.9602002.

[33] Li L, Yu X, Cai X, He X, Liu Y. Contract-theory-based incentive mechanism for federated learning in health crowdsensing. *IEEE Internet of Things Journal*, 2023, 10(5): 4475–4489. DOI: 10.1109/JIOT.2022.3218008.

[34] Jiang C, Gao L, Duan L, Huang J. Exploiting data reuse in mobile crowdsensing. In *Proc. the 2016 IEEE Global Communications Conference*, Dec. 2016, pp.1–6. DOI: 10.1109/GLOCOM.2016.7841828.

[35] Liu W, Yang Y, Wang E, Wu J. Dynamic user recruitment with truthful pricing for mobile crowdsensing. In *Proc. the 2020 IEEE Conference on Computer Communications*, Jul. 2020, pp.1113–1122. DOI: 10.1109/INFOCOM41043.2020.9155242.

[36] Xiao M, Gao G, Wu J, Zhang S, Huang L. Privacy-preserving user recruitment protocol for mobile crowdsensing. *IEEE/ACM Trans. Networking*, 2020, 28(2): 519–532. DOI: 10.1109/TNET.2019.2962362.

[37] Lyamin N, Vinel A, Jonsson M, Loo J. Real-time detection of denial-of-service attacks in IEEE 802.11p vehicular networks. *IEEE Communications Letters*, 2014, 18(1): 110–113. DOI: 10.1109/LCOMM.2013.102213.132056.

[38] Wu T J, Liao W, Chang C J. A cost-effective strategy for road-side unit placement in vehicular networks. *IEEE Trans. Communications*, 2012, 60(8): 2295–2303. DOI: 10.1109/TCOMM.2012.062512.100550.

[39] Mehmeti F, Spyropoulos T. Is it worth to be patient? Analysis and optimization of delayed mobile data offloading. In *Proc. the 2014 IEEE Conference on Computer Communications*, May 2014. DOI: 10.1109/INFOCOM.2014.6848181.

[40] Zhou C, Tham C K, Motani M. Online auction for scheduling concurrent delay tolerant tasks in crowdsourcing systems. *Computer Networks*, 2020, 169: 107045. DOI: 10.1016/j.comnet.2019.107045.

[41] Mantri D, Prasad N R, Prasad R. Grouping of clusters for efficient data aggregation (GCEDA) in wireless sensor network. In *Proc. the 3rd IEEE International Advance Computing Conference*, Feb. 2013, pp.132–137. DOI: 10.1109/IAdCC.2013.6514208.

[42] Xu J, Yang S, Lu W, Xu L, Yang D. Incentivizing for truth discovery in edge-assisted large-scale mobile crowdsensing. *Sensors*, 2020, 20(3): 805. DOI: 10.3390/s20030805.

[43] Zheng Z, Peng Y, Wu F, Tang S, Chen G. Trading data in the crowd: Profit-driven data acquisition for mobile crowdsensing. *IEEE Journal on Selected Areas in Communications*, 2017, 35(2): 486–501. DOI: 10.1109/JSAC.2017.2659258.

[44] Sheng B, Li Q, Mao W. Data storage placement in sensor networks. In *Proc. the 7th ACM International Symposium on Mobile ad Hoc Networking and Computing*, May 2006, pp.344–355. DOI: 10.1145/1132905.1132943.

[45] Burkhardt T, Leopold A, Leopold-Wildburger U. Markov simulation of an iterated prisoners' dilemma experiment. In *Proc. the 2011 Selected Papers of the International Conference on Operations Research on Operations Research Proceedings*, Sept. 2012, pp.223–228. DOI: 10.1007/978-3-642-29210-1_36.

[46] Lacey N. The Prisoners' Dilemma. Cambridge University Press, 2008: 9–17. DOI: 10.1017/CBO9780511819247.

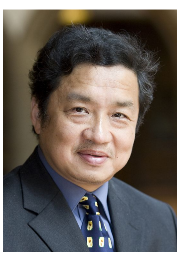
[47] Akbay A B, Zhang J. Distributed learning with strategic users: A repeated game approach. In *Proc. the 36th AAAI Conference on Artificial Intelligence*, Jun. 2022, pp.5976–5983. DOI: 10.1609/aaai.v36i6.20543.

[48] Imhof L A, Fudenberg D, Nowak M A. Tit-for-tat or win-stay, lose-shift?. *Journal of Theoretical Biology*, 2007, 247(3): 574–580. DOI: 10.1016/j.jtbi.2007.03.027.

In Press

Ji-Qing Gu received her Ph.D. degree from the School of Computer Science and Engineering, University of Electronics Science and Technology of China (UESTC), Chengdu, in 2022. She is currently a lecture in the School of Computer Science, Chengdu University of Information Technology, Chengdu. Her research interests include crowdsensing, mobile computing, distributed computing, and graph stream mining.

Chao Song received his Ph.D. Degree in Computer Science from University of Electronic Science and Technology of China (UESTC), Chengdu, in 2009. During 2013, he was a visiting scholar at Temple University, under the supervision of Dr. Jie Wu. He is currently an associate professor in the School of Computer Science and Engineering at UESTC. His main research interests include mobile computing, vehicular networks, and distributed computing.



Jie Wu is the director of the Center for Networked Computing and Laura H. Carnell professor at Temple University, Philadelphia. 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. Dr. Wu is a CCF Distinguished Speaker and a Fellow of the IEEE. He is the recipient of the 2011 China Computer Federation (CCF) Overseas Outstanding Achievement Award.

Li Lu received the Ph.D. degree from the Key Laboratory of Information Security, Chinese Academy of Science, Beijing, in 2007. He is a professor with the School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu. His current research interests include battery-free systems and RFID technology, wireless networks, and network security.

Ming Liu received his M.S. degree from the University of Electronic Science and Technology of China, Chengdu, in 2003, and his Ph.D. degree from Nanjing University, Nanjing, in 2006. He is currently a full professor with the School of Computer Science and Engineering, University of Electronic Science and Technology of China. His research interests include mobile computing, wireless sensor networks, distributed computing, deep learning, and big data analysis.

In Press