

Research Article

Multistrategy Repeated Game-Based Mobile Crowdsourcing Incentive Mechanism for Mobile Edge Computing in Internet of Things

Chuanxiu Chi ^{1,2}, Yingjie Wang ^{1,2}, Yingshu Li ^{1,3} and Xiangrong Tong ^{1,2}

¹School of Computer and Control Engineering, Yantai University, Yantai 264005, China

²Yantai Key Laboratory of High-end Ocean Engineering Equipment and Intelligent Technology, Yantai University, Yantai 264005, China

³Department of Computer Science, Georgia State University, Atlanta, GA 30303, USA

Correspondence should be addressed to Yingjie Wang; wangyingjie@ytu.edu.cn

Received 11 November 2020; Revised 10 December 2020; Accepted 4 January 2021; Published 27 January 2021

Academic Editor: Shaohua Wan

Copyright © 2021 Chuanxiu Chi et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

With the advent of the Internet of Things (IoT) era, various application requirements have put forward higher requirements for data transmission bandwidth and real-time data processing. Mobile edge computing (MEC) can greatly alleviate the pressure on network bandwidth and improve the response speed by effectively using the device resources of mobile edge. Research on mobile crowdsourcing in edge computing has become a hot spot. Hence, we studied resource utilization issues between edge mobile devices, namely, crowdsourcing scenarios in mobile edge computing. We aimed to design an incentive mechanism to ensure the long-term participation of users and high quality of tasks. This paper designs a long-term incentive mechanism based on game theory. The long-term incentive mechanism is to encourage participants to provide long-term and continuous quality data for mobile crowdsourcing systems. The multistrategy repeated game-based incentive mechanism (MSRG incentive mechanism) is proposed to guide participants to provide long-term participation and high-quality data. The proposed mechanism regards the interaction between the worker and the requester as a repeated game and obtains a long-term incentive based on the historical information and discount factor. In addition, the evolutionary game theory and the Wright-Fisher model in biology are used to analyze the evolution of participants' strategies. The optimal discount factor is found within the range of discount factors based on repeated games. Finally, simulation experiments verify the existing crowdsourcing dilemma and the effectiveness of the incentive mechanism. The results show that the proposed MSRG incentive mechanism has a long-term incentive effect for participants in mobile crowdsourcing systems.

1. Introduction

The increasing data demand in the 5G era is a huge challenge for IoT devices with limited computing power and resources. All data are transmitted through the network to the traditional cloud platform for centralized processing. This method is not conducive to data security and privacy and has poor real-time performance and high bandwidth pressure and energy consumption [1, 2]. The emergence of MEC can effectively reduce the risk of privacy leakage and system delays and relieve the pressure on network bandwidth and data center energy consumption [3, 4]. Due to the limited

resources and computing power of a single device in MEC, a large amount of sense data is needed in practical applications [5, 6]. Mobile crowdsourcing is considered a promising method for obtaining massive amounts of data [7].

Mobile crowdsourcing provides a new model for solving problems by gathering the wisdom of groups. Numerous crowdsourcing websites show the collaborative nature of crowdsourcing, such as Yahoo! Answers [8] and Wikipedia [9, 10], which can be viewed as systems where small tasks are performed in exchange for rewards. At present, crowdsourcing has been applied in many fields, such as environmental quality inspection issues, noise level detection

projects, and commercial map services [11–14]. Workers use various devices to sense the environment in MEC, and edge nodes replace traditional cloud platforms to process these transaction data [15, 16]. The success of a crowdsourcing system largely depends on whether users are actively involved and their willingness to make efforts for sensing tasks [17, 18]. Due to insufficient participants in perception tasks and data quality issues, crowdsourcing systems need to adopt appropriate incentive mechanisms to actively guide participants to participate in tasks and provide high-quality services [19, 20]. Since the number of workers cannot maintain long-term participation, it is important to encourage workers to maintain higher enthusiasm under the condition that the number of workers does not decrease. Besides, the issue of completion quality is also an important research issue of the incentive mechanism. Since the platform and the requester cannot directly observe the worker's status, there is no guarantee that the task will be completed with high quality [21–23]. Completing a task requires resources such as battery power, computing resources, and data traffic [24]. To obtain greater benefits, workers may be lazy to reduce their cost consumption, which will seriously affect the completion quality of the task.

In this paper, the incentive mechanism is added in the crowdsourcing process, and a reasonable pricing plan is formulated based on the contribution of workers to complete the task. To compensate for the user's direct sensing cost when performing a particular sensing task, we call it short-term sensing incentive, but long-term participation of workers requires long-term incentives. Therefore, this paper researches the long-term incentive mechanism to motivate workers to continuously improve their efforts and ensure that workers continuously provide high-quality data.

In order to solve the problem, this paper models the interaction between workers and requesters as repeated games and calculates the specific discount factor value that could maintain the equilibrium. In the repeated game, according to the worker's data contribution, the degree of effort is divided into five intervals, i.e., the worker's strategy is the five different degrees of effort. In this paper, an evolutionary game is introduced to simulate the evolution of players' strategic choices, and the evolutionary stability strategy (ESS) of both players is analyzed [25]. The simulation results show that the requester and the worker will eventually adopt the strategy of high effort and employment to maintain the long-term optimal benefits. In summary, the main contributions of this article are as follows:

- (1) This paper uses a game to simulate the interaction between requester and workers, shows the conflict of interest between the two parties, and proves that there is a crowdsourcing dilemma in a dynamic game
- (2) Dynamic strategy selection is modeled as an evolutionary game. The evolutionary game theory and the Wright-Fisher model are combined to analyze the evolution trend, and the Wright-Fisher model is used to calculate the adaptability of different populations to calculate the adaptability of users with different strategies

- (3) This paper designs an incentive mechanism based on the multistrategy repeated game model to solve the dilemma. Using the discount factor of the repeated game and historical data, the current behavior of the two sides of the game is proposed, and an effective algorithm is proposed to obtain the Nash equilibrium and find the optimal payoff
- (4) Both theoretical analysis and simulation experiments show that the proposed incentive mechanism could more effectively motivate workers to continue to provide high-quality data to improve the performance of the platform. The experimental results also show that under the proposed incentive mechanism, the steady state of the evolution of the two parties' strategies is no longer a dilemma state

The rest of the paper is organized as follows. Section 2 reviews the related works and presents the motivation for our work. Section 3 gives an overview of the crowdsourcing system and three definitions of the system. Section 4 describes the crowdsourcing dilemma and uses an evolutionary game to analyze the changes in the strategies of both players. Section 5 proposes a multistrategy model based on a repeated game and describes the algorithm. Section 6 presents our simulation results and analysis of the results. Conclusions and future work are discussed at the end.

2. Related Work

From the earlier discussion, we can discover that the workers will be unwilling to contribute information unless they receive enough compensation for their cost of resources. So the research on incentive mechanisms is necessary. In this section, we investigate and study the related works of incentive mechanism, repeated game theory, and evolutionary game theory for strategy analysis.

2.1. Incentive Mechanisms. An incentive mechanism is commonly used in crowdsourcing applications as a key part of the crowdsourcing system. Incentives are achieved by solving the problems of quality, payment control, and energy consumption when maximizing the utilities. There are many different incentive methods in the perception of swarm intelligence, which can be divided into monetary incentives and nonmonetary incentives [26–30]. Incentive mechanisms based on monetary rely on monetary or matching rewards in the form of micropayment to motivate workers to provide high-quality services [31]. The monetary incentive mechanism is mainly based on the game theory and auction mechanism [32, 33]. A reverse auction is different from the traditional auctions because the reverse auction includes one buyer and multiple sellers. During the reverse auction [25, 34–38], the requester selects a subset of workers with a lower cost under a certain budget consideration, and the sense data of workers is purchased at their bid prices. Considering that specific tasks are limited by budget and require workers with one or more skills, Zhang et al. [25] proposed a reverse auction-based incentive mechanism, assigning tasks to competent workers and rewarding workers for completing tasks.

Samad and Kanhere [34] proposed the Modified Reverse Auction (MRA), the winning probability of the participants is estimated and revealed individually before the auction is closed, and then, they are allowed to improve their winning probability by reducing their bid or increasing their contribution via moving to a different location. Zhou et al. [35] designed a novel delay-constraint and reverse auction-based incentive mechanism (DRAIM). In DRAIM, authors modeled the reverse auction-based incentive problem as a nonlinear integer problem, aiming to maximize the revenue and jointly consider the delay constraint in the optimization problem. To avoid malicious competition and select high-quality crowd workers to improve the utility of the crowdsourcing system, Hu et al. [36] proposed an incentive mechanism based on the combination of the reverse auction and multiattribute auction in mobile crowdsourcing. This mechanism adopts a dynamic threshold to make an online decision for whether to accept a crowd worker according to its attributes. To investigate the joint problem of sensing task assignment and schedule, Cai et al. [37, 38] proposed two distributed auction schemes (CPAS and TPAS). Luo et al. [39] designed an incentive mechanism for scenarios involving heterogeneous types of workers (and the beliefs about their respective types) using an asymmetric all-pay auction model.

In addition to the incentive mechanism based on auction theory, Wu et al. [40] designed a mechanism based on the Stackelberg game to encourage participants to compete and participate in tasks, where the requester determines a certain total payment amount from the beginning. Taking the prior knowledge as a specific incentive, Lan et al. [41] proposed a novel classifier that can accurately recognize different categories. An incentive mechanism for the platform-centric mobile crowdsourcing was designed by Zhan et al. [42], which considers the resource requirements of the users and resource constraints of smart devices. They formulated the interaction between the requester and workers as one-to-many bargaining; then, they studied the bargaining solutions under ordered bargaining and simultaneous bargaining systematically.

Although the above monetary incentive mechanisms are simple and effective, they also have some shortcomings. Due to the selfishness of individuals, their purpose is to maximize their benefits [43]. References [44, 45] pointed out that there are distrust problems such as free-riding problems and false-reporting problems. To encourage workers to choose a trust strategy, Wang et al. [44] proposed an online incentive mechanism based on a reputation for mobile crowdsourcing systems and established a reputation updating method. In order to prevent the free-riding problem of workers and motivate workers to contribute their efforts, Zhang et al. [45] proposed a novel class of incentive protocols based on social norms.

Most of the existing incentive mechanisms are short-term incentives that directly pay workers. These incentive mechanisms cannot attract users to participate in crowd tasks for a long time. When workers leave the crowdsourcing platform because they lose interest, the performance of the platform will significantly decrease. Gao et al. [46] proposed a Lyapunov-based Vickrey-Clarke-Groves auction policy for

online sensor selection, which is aimed at maximizing social welfare and ensure the long-term participation incentive of users. Gao et al. [47] proposed a long-term quality perception incentive model in a crowdsourcing environment with budget constraints. The long-term incentive mechanism is to motivate crowd workers to provide long-term continuous services for crowd tasks. However, the above methods ignored the problem of workers' effort.

2.2. Repeated Game Theory. Inspired by game theory, the study of repeated games in incentive mechanisms has become a hot topic [48–51]. In repeated games, the player's goal changes from the current maximum profit to the maximum profit of multiple games, which means that future returns are closely related to the current behavior. Therefore, participants can be forced to avoid selfish behavior, and repeated games can also solve the dilemma. Mailath et al. [48] gathered and analyzed a metadata set of experiments on prisoners' dilemma games. They used experiments to prove that cooperation is affected by infinite repetition and that high cooperation rates are more likely to arise when it can be supported in equilibrium. To deal with the selfish behaviors of workers, Gao et al. [49] proposed an enhanced cooperative authentication protocol. For this designed protocol, an infinitely repeated game was proposed to analyze the utility of all users to help analyze the threat of selfish behavior. Hu et al. [50] proposed to model the interaction between the requester and crowd workers as a game process under the theoretical framework of repeated games; requesters adopted sequential zero-determination strategies to solve the crowdsourcing dilemma. Yin et al. [51] used a repeated game with incomplete information to motivate nodes to forward an advertisement.

2.3. Evolutionary Game Theory. In recent decades, the researches on evolutionary game theory (EGT) has become increasingly widespread [52]. Evolutionary stable strategy (ESS) and replication dynamics have been put forward successively in the course of its development, which laid the theoretical foundation for subsequent research. In EGT, if the later mutation behavior cannot shake a certain strategy executed by the population, that strategy is considered to be stable in the evolution process. ESS ensures stability and identifies robustness against mutations. Although the EGT was originally developed for biology, many exciting works have utilized EGT to model problems [53–55]. For example, Yin et al. [53] proposed an incentive mechanism based on EGT to inspire entities to select strategies that have high trustworthiness. To address the strategic uncertainty that users may face, EGT provided an excellent means. Wang et al. [54] used an evolutionary game framework to answer the question of "how to collaborate" in multiuser decentralized cooperative spectrum sensing. An evolutionary game is used to simulate the behavior of nodes in a network; Fang et al. [55] designed a budget allocation mechanism to encourage cooperation between adjacent nodes. The replicator dynamic can construct the gradual evolution of strategies to show the process of ESS. It is useful for investigating the trajectory of the strategies of players while adapting their

behaviors to reach the solution. Therefore, this paper combined the evolutionary game with the Wright-Fisher model for strategy evolution.

Therefore, according to the above problems, this paper proposed a multistrategy model to enhance the attractiveness of the platform under the framework of repeated games, which can not only ensure long-term participation of users but also guide workers to continuously improve their efforts.

3. System Overview

In this section, a generalized model in crowdsourcing systems is given, and then, the workflow of the crowdsourcing system is described in detail. Secondly, considering the historical behavior data of participants, according to different levels of effort, a multistrategy model is proposed.

3.1. System Description. A crowdsourcing system generally includes a crowdsourcing platform and users. Users could be divided into workers and requesters. Requesters could publish tasks through the crowdsourcing platform, and workers accept tasks and complete specific tasks with the help of mobile smart devices such as mobile phones. The crowdsourcing system could be elaborated from three aspects.

3.1.1. Requester. The requester publishes the task on the crowdsourcing platform; gives information about the task's time constraint, space constraint, and price; and then waits for feedback from the platform. The set of requesters is denoted by $R \triangleq \{1, 2, \dots, j, \dots, N\}$.

3.1.2. Worker. Workers could choose to complete tasks distributed by the platform, or they can choose tasks independently. To complete a specific task, workers need to provide specific data and information and then wait for feedback from the platform. The set of the workers is denoted by $W \triangleq \{1, 2, \dots, i, \dots, N\}$.

3.1.3. Platform. The crowdsourcing platform mainly includes two important parts, i.e., the tasks processing server and the payment server. The task processing server is responsible for task allocation, distributes tasks from requesters to appropriate workers, and promptly feeds back information about tasks required by both parties during the transaction. The payment server determines the worker's payment for this task based on the completion of the worker's task and distributes the payment to the designated worker.

According to the above three roles, the workflow of the crowdsourcing system includes the following steps. First of all, the requester releases the task information, the task processing server assigns the task, the worker chooses whether to accept the task, and if the task is accepted, the description of the available results will be returned to the task server. Then, the task server sends the task assignment result to the requester. Next, after receiving the result, the requester calculates the actual payment and sends the actual payment result feedback to the payment server. Finally, after receiving the information, the payment server informs the payment information to the worker. After the worker knows the actual pay-

ment, she submits the real data to the platform. The task server forwards the data to the requester, and the payment server sends the payment to the worker.

3.2. System Definition. According to the data contribution of workers, the efforts of workers are divided into different levels. The higher the level of effort workers provide, the greater the amount of contribution the requester will provide. Conversely, for workers with low effort levels who cannot meet the task requirements, the system will cancel their cooperation and achieve the purpose of punishment. According to different levels of effort, this paper adjusts the strategies of the participating parties in a targeted manner, to achieve different levels of income calculation functions.

When workers provide services, the level of effort of the worker $i, i \in W$, determines the amount of contribution Q_i the workers make in a specific task. In reality, workers can decide the contribution they want to provide. The relationship between the effort e of workers and the amount of contribution is shown by

$$Q_i(e_i) = e_i \times \varepsilon_i. \quad (1)$$

The random variable ε_i obeys a probability distribution function with an expectation of 0 and probability density function $f(\varepsilon)$.

Since the mobile devices used by workers in crowdsourcing consume resources such as power, memory, and time, a resource consumption cost $C(q_{ij})$ will be incurred when a certain amount of contribution is provided:

$$C(q_{ij}) = \frac{1}{2} \times \beta \times q_{ij}^2, \quad q_{ij} \in Q_i, \quad (2)$$

where q_{ij} represents the contribution provided by worker i to requester j ; it belongs to the set of workers' contributions. $C(q_{ij})$ represents the negative utility to complete the task, i.e., cost consumption, which is an increasing convex function, $C'(q_{ij}) > 0$ and $C''(q_{ij}) > 0$.

Definition 1 (utility of worker). Workers will have negative utility when they use mobile devices to provide data needed for tasks, and they will be paid after completing the tasks. The utility of workers is the difference between the payment and the negative utility:

$$U_i^j(q_{ij}) = P(q_{ij}) - C(q_{ij}), \quad (3)$$

where $U_i^j(q_{ij})$ represents the utility of the transaction between worker i and requester j . $P(q_{ij})$ means the payment to user i for data contribution q_{ij} .

Definition 2 (utility of requester). The information provided by the worker can bring a certain profit to the requester, and the requester needs to provide the worker with corresponding payment. The utility of the requester is the difference between the profit and the payment:

$$U_j^i(q_{ij}) = L(q_{ij}) - P(q_{ij}), \quad (4)$$

where U_j^i represents the utility of the transaction between worker i and requester j . $L(q_{ij})$ is the value of data contribution q_{ij} ; the calculation method is shown in

$$L(q_{ij}) = \varsigma \times q_{ij}, \quad q_{ij} \in Q_i. \quad (5)$$

Definition 3 (effort level of worker). Although the requester cannot directly observe the real effort of the worker, he can know the amount of data contribution that the worker can provide. Therefore, the effort of the worker is evaluated according to the amount of data contribution that the worker can provide during the completion of a task. To ensure the quality of the collected data and ensure that workers have to work hard for each task, the requester can formulate a measurement method for the quality of the submitted data. This paper divides the interval according to the contributions submitted by the workers, which is calculated by

$$\text{Level} = \tanh \left(k \times \ln \left(\frac{q_{ij}}{q_{\text{normal}}} \right) \right), \quad (6)$$

where q_{normal} is the standard contribution of this task, i.e., the minimum contribution required to complete this task. The parameter k is an adjustable parameter. The reason for using these two functions is to quantify the data. When the level is less than 0, it indicates that the worker has not worked hard enough, and the data provided cannot meet the needs of the task.

In this paper, according to the actual data contribution of workers, the degree of effort is determined. The corresponding amount of data contribution provided by different degrees of effort is shown in Table 1.

Workers with different levels of effort have different utility functions. After analyzing Equation (6), it can be seen that the higher the level, the higher the level of effort of workers. Therefore, the utility calculation function of these five levels is obtained by

$$\begin{cases} U_{\text{level}_d}^i(q_{ij}) = (a + 1.5) \times P(q_{ij}) - C(q_{ij}), & \text{level}_d < 0, \\ U_{\text{level}_d}^i(q_{ij}) = P(b, q_{ij}) - C(q_{ij}), & \text{level}_d \geq 0, \quad d = \{1, 2, 3, 4, 5\}, \end{cases} \quad (7)$$

where a and b are the left and right boundary values of different degree intervals, respectively.

4. Model and Formulation

In this section, dynamic games are used to model the interaction between workers and requesters; relevant rules of the game are given, and the existence of a dilemma is proposed. With the help of evolutionary games, it verifies that there are indeed dilemmas.

TABLE 1: Levels of effort.

Degrees of effort	Levels of effort
[-1, -0.5)	level ₁
(-0.5, 0)	level ₂
0	level ₃
(0, 0.5)	level ₄
(0.5, 1]	level ₅

4.1. Game Formulation. In this section, the theoretical background of this model and related assumptions are given before modeling. In this model, the requester and the worker pursue profit maximization and individual utility maximization. Under the condition of information asymmetry, the requester needs to pay a certain amount to obtain information, which is to bring more benefits. Because the requester does not fully understand the worker's effort, the worker may be lazy. The task needs to be assigned to a series of workers who meet the requirements. It should be noted that from the requester's perspective, the actions of the workers required for a specific task follow the same pattern. Therefore, the success incentive for any worker means that it has a very high probability of success in this type of worker. This paper regards an interaction between the requester and any worker as a dynamic game process.

Specifically, a stage game is that after the requester publishes the task, the worker chooses the data contribution strategy to maximize her utility. Then, the requester decides whether to hire the worker by observing the strategy of the worker and decides how much to pay her. In this process, we assume the following:

Hypothesis 1: the platform has historical information about workers, and workers in the previous stage can participate in the tasks of the next stage after completing the tasks.

Hypothesis 2: ignore the cost consumption of supervision.

Hypothesis 3: the total duration of the game consists of a series of discrete stage games; each worker performs only one task in a stage.

Hypothesis 4: every game follows the same rules and processes.

Hypothesis 5: enough hard workers choose all tasks uniformly and randomly, and tasks have the same probability of being selected. This assumption does not reduce the number of potential workers. The mechanism proposed in this paper can actively guide workers to increase their efforts.

4.2. Game Model. In the stage game, the strategy of the worker is the amount of data contribution submitted to complete the task, while the strategy of the requester is whether to hire the worker. In this model, workers can use different contribution strategies. If the contribution is within the acceptance range of the requester, the requester chooses to hire workers and provide corresponding payment. The set of workers' strategies is defined as (level₁, level₂, level₃, level₄, and level₅); the higher the level, the greater the contribution. The set of requesters' strategies is defined as (Y, N) . Y and N

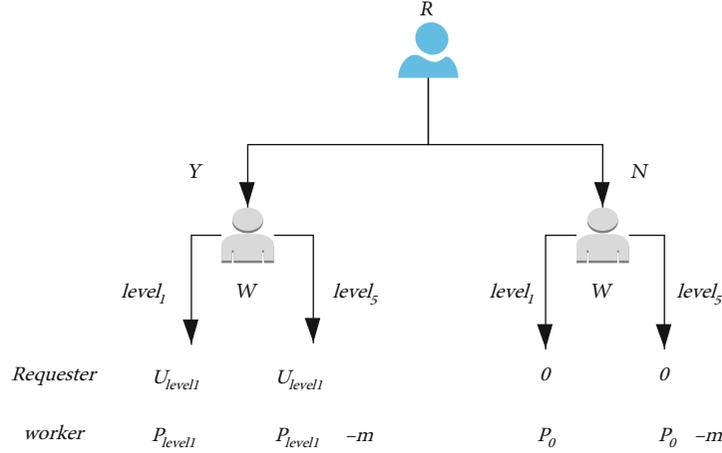


FIGURE 1: The game tree of a one-stage game between a requester and a worker.

indicate that the requester chooses to hire workers and the requester refuses to hire workers, respectively.

A stage game G is defined by $G = (N, S_k, U_k)$, $k \in N$, where $N = \{i, j\}$, $i \in W$, $j \in R$, is a finite set of players, S_k is the set of player k 's actions, the strategy set of worker is $S_i = \{\text{level}_1, \text{level}_2, \text{level}_3, \text{level}_4, \text{level}_5\}$, the strategy set of the requester is $S_j = \{Y, N\}$, and U_k is player k 's payoff function.

Definition 4 (Nash equilibrium solution of stage game). When S_i and S_j are fixed, if S_j^* satisfies $U_j(S_j^*) \geq U_j(S_j)$ and S_i^* satisfies $U_i^w(S_i^*) \geq U_i^w(S_i)$, we called (S_i^*, S_j^*) as the Nash equilibrium strategy in the proposed game.

Backward induction could be used to solve this Nash equilibrium. If the worker refuses this task, the worker becomes a self-employed person and only retains self-employment income p_0 . Otherwise, workers choose their contribution strategy (level₁, level₂, level₃, level₄, and level₅). Providing data will bring negative utility to workers, and the greater the effort, the greater the negative utility. This paper first discusses the situation where workers have only two extreme strategies, level₁ and level₅, and extends it to a multi-strategy game. Through the game tree ($m > 0$, $U_{level_1} < 0$) shown in Figure 1, we can get the game result of backward induction. In the second step, workers will accept tasks and choose the level₁ strategy to reduce their negative utility. When the worker chooses the level₁ strategy, the requester will benefit more from the nonemployment strategy, so the requester will choose N . Therefore, the Nash equilibrium of this game is (level₁, N).

In this paper, we extend the two strategies to multiple strategies. Assuming that the requester hires worker i with a payment P^i , when the worker's effort is high, the benefits of the requester and the worker are the following: the requester's income is the difference between the profit obtained from the worker's data and the payment paid to worker i , and the worker's income is the difference between the payment from the requester and the negative utility. If

TABLE 2: Benefit matrix of worker.

Strategy	Y	N
Level ₁	$U_W(\text{level}_1, Y)$	$U_W(\text{level}_1, N)$
Level ₂	$U_W(\text{level}_2, Y)$	$U_W(\text{level}_2, N)$
Level ₃	$U_W(\text{level}_3, Y)$	$U_W(\text{level}_3, N)$
Level ₄	$U_W(\text{level}_4, Y)$	$U_W(\text{level}_4, N)$
Level ₅	$U_W(\text{level}_5, Y)$	$U_W(\text{level}_5, N)$

the worker is lazy (that is, the effort level is level₁ or level₂), there is no negative effect.

If the game is played only once at this stage, the benefits of both parties are shown in Tables 2 and 3; the calculation formulas are

$$\begin{cases} U_W(\text{level}_d, Y) = P^i, & d = \{1, 2\}, \\ U_W(\text{level}_d, Y) = P^i - C(q_{ij}), & d = \{3, 4, 5\}, \\ U_W(\text{level}_d, N) = P_0^i(q_{ij}), & d = \{1, 2, 3, 4, 5\}, \end{cases} \quad (8)$$

$$\text{s.t. } L(q_{ij}) - C(q_{ij}) > P_0 > \rho \times L(q_{ij}),$$

$$\begin{cases} U_R(\text{level}_d, Y) = \rho \times L(q_{ij}) - P^i, & d = \{1, 2\}, \\ U_R(\text{level}_d, Y) = L(q_{ij}) - P^i, & d = \{3, 4, 5\}, \\ U_R(\text{level}_d, N) = 0, & d = \{1, 2, 3, 4, 5\}, \end{cases} \quad (9)$$

$$\text{s.t. } L(q_{ij}) - C(q_{ij}) > P_0 > \rho \times L(q_{ij}).$$

After the above analysis, without any incentives, the result of the game is (level₁, N). This paper refers to this phenomenon as a crowdsourcing dilemma. Then, evolution games are used to evolve users' strategy changes. The existence of the crowdsourcing dilemma and the effectiveness

TABLE 3: Benefit matrix of requester.

Strategy	Y	N
Level ₁	$U_R(\text{level}_1, Y)$	$U_R(\text{level}_1, N)$
Level ₂	$U_R(\text{level}_2, Y)$	$U_R(\text{level}_2, N)$
Level ₃	$U_R(\text{level}_3, Y)$	$U_R(\text{level}_3, N)$
Level ₄	$U_R(\text{level}_4, Y)$	$U_R(\text{level}_4, N)$
Level ₅	$U_R(\text{level}_5, Y)$	$U_R(\text{level}_5, N)$

of the incentive mechanism proposed in this paper have also been verified.

4.3. Evolution Analysis

4.3.1. Strategy Stability Analysis of Workers. Suppose n workers are participating in a task, and the number of workers in the five strategies are n_{level_d} , $d \in \{1, 2, 3, 4, 5\}$. The sum of the quantities is n . According to the income matrix, combined with the calculation method of the adaptability of different populations in the Wright-Fisher model, the adaptabilities of different strategies are calculated by

$$F_{\text{level}_d} = \frac{(n_{\text{level}_d} - 1) \times U_W(\text{level}_d, Y)}{n - 1}. \quad (10)$$

Since the Wright-Fisher process is to perform the N -fold Bernoulli experiment in the offspring set, the perceived worker obeys the binomial distribution. Assuming that n_i^m represents the number of workers at level i in the m th generation in the group, the probability that the number of workers at level i in the $m+1$ th generation is n_i^{m+1} is shown by

$$p\left(n_{\text{level}_d}^{m+1} \mid n_{\text{level}_d}^m\right) = \binom{n}{n_{\text{level}_d}^{m+1}} \prod_{d=1}^5 \left(\frac{n_{\text{level}_d}^m \times F_{\text{level}_d}}{\sum_{d=1}^5 n_{\text{level}_d}^m \times F_{\text{level}_d}} \right)^{n_{\text{level}_d}^{m+1}}. \quad (11)$$

In the Wright-Fisher process, individuals are updated synchronously; $E(\Delta x)/\Delta t$ in the game on the graph can be used to approximately replace the copy dynamic equation d_x/d_t in the evolutionary game. $E(\Delta x)$ represents the change in the individual frequency of the worker of strategy level_d ; Δt indicates the compensation step of the update time. The calculation method $E(\Delta x)$ is shown by

$$\begin{aligned} E(\Delta x) &= \frac{\sum_{n_{\text{level}_d}^{m+1}=0}^n (n_{\text{level}_d}^{m+1} - n_{\text{level}_d}^m) \times p\left(n_{\text{level}_d}^{m+1} \mid n_{\text{level}_d}^m\right)}{n} \\ &= \frac{n_{\text{level}_d}^m \times F_{\text{level}_d}}{\sum_{*} n_{*}^m \times F_{*}} - x. \end{aligned} \quad (12)$$

In this paper, the proportion of workers with five types of strategies is set x_d , $d \in \{1, 2, 3, 4, 5\}$, and the evolution prediction model of perceiving workers in the mobile

crowdsourcing system can be obtained. Equation (13) is the evolution prediction model of the strategies:

$$\frac{dx_d}{dt} = \frac{x_d \times \overline{u_{\text{level}_d}} / \bar{u} - x_d}{\Delta t}. \quad (13)$$

The benefit calculation method is shown by Equations (14) and (15), where y means the proportion of requesters who decides to hire workers, $\overline{u_{\text{level}_d}}$ represents the expected benefits of workers at different levels of effort, and \bar{u} represents the average expected benefits of all workers in the system:

$$\overline{u_{\text{level}_d}} = y \times U_W(\text{level}_d, Y) + (1 - y) \times U_W(\text{level}_d, N), \quad (14)$$

$$\bar{u} = \sum_{d=1}^5 x_d \times \overline{u_{\text{level}_d}}. \quad (15)$$

According to the evolution prediction model, the evolution trend of different types of workers can be predicted. To find the ESS, that is, the Nash equilibrium, two conditions $F(x) = d_x/d_t = 0$ and $F(x)' < 0$ need to be met at the same time. The Nash equilibrium point is obtained through computing $x_d = 0$, $d \in \{2, 3, 4, 5\}$. It can be seen that due to the selfishness of the workers, the ESS of the workers is the strategy level_1 , which also corresponds to the crowdsourcing dilemma in the previous section.

4.3.2. Strategy Stability Analysis of Requesters. When the worker's effort is low, the requester will adopt a N strategy. If this situation persists for a long time, the number of task transactions in the system will drop significantly, resulting in system performance degradation. To analyze the evolution trend of task requesters with different strategies, a trust evolution prediction model for task requesters is also established. Similarly, the requester's strategies include employment and nonemployment, the proportion of these two types of requesters is y_d , $d \in \{1, 2\}$, and the evolution prediction model of the two strategies is

$$\frac{dy_d}{dt} = \frac{y_d \times \overline{u_Y} / \bar{u} - y_d}{\Delta t}. \quad (16)$$

The calculation method of income is $\overline{u_Y}$ representing the expected return of the requester who chooses strategy Y . $\overline{u_N}$ indicates the expected return of the requester who chooses strategy N ; \bar{u} represents the average expected return of all requesters in the system:

$$\begin{aligned} \overline{u_Y} &= y_1 \times \left(\sum_{d=1}^5 x_d \times U_R(Y, \text{level}_d) \right), \\ \overline{u_N} &= y_2 \times \left(\sum_{d=1}^5 x_d \times U_R(N, \text{level}_d) \right), \\ \bar{u} &= y_1 \times \overline{u_Y} + y_2 \times \overline{u_N}. \end{aligned} \quad (17)$$

According to the evolution prediction model, the evolution trend of different types of requesters can be predicted. To find the ESS, that is, the Nash equilibrium, two conditions $F(y) = d_y/d_t = 0$ and $F(y)' < 0$ need to be met at the same time. The Nash equilibrium point is obtained through computing $y_1 = 0$. When the number of low-effort workers in the group is the majority, the requester's ESS in the system is not hired, which is consistent with the previous question of the crowdsourcing dilemma. To solve the above problems, a multistrategy repeated game incentive model was established to motivate workers. Requesters hire workers with appropriate remuneration and maintain the system to provide long-term and efficient services.

5. Design of Incentive Mechanism

To solve this dilemma, ensure the long-term participation of users and improve the quality of workers' completion. The interaction between workers and requesters is modeled as a repeated game, and a multistrategy incentive mechanism is proposed in this section.

5.1. Analysis of Multistrategy Repeated Game Model. In the repeated game, the game of the same structure is repeated many times, even infinite times. Players will adopt a confrontational strategy to maximize personal gains in a one-shot game. For example, the incident of a stranger grabbing a seat on a bus can be considered a game. If the two parties are not strangers, and they give up their immediate interests because of future interaction, there will be no quarrel. Therefore, repeated games can solve the dilemma problem. If it is repeated for a limited number of N times, then in the N th game, the participants know that if they choose a low price in this game, they will benefit themselves and will not leave the opponent with any chance of revenge.

When the workers cannot predict when the game will end, the workers will not adopt a confrontation strategy in the last game; that is, there will be no critical state in which the worker's effort is low in the last game. Therefore, to eliminate the dilemma caused by selfishness, although personal working ability, time, and other conditions are limited conditions, the game between the two parties can still be regarded as an infinitely repeated game in the analysis process. Under a certain discount factor, the trigger strategy in the infinite repeat game can form a subgame Nash equilibrium, which can maintain a long-term high-level service.

The stage game is repeated continuously in time, and the historical information of the game can be obtained before the next game starts. Simply put, a repeated game is a stage game with the same structure repeated many times. Suppose G is a stage game. When G is repeated T times, it is called a repeated game $G(T)$; when T is limited, it is called a finite repeated game; and when T is infinite, it is called an infinite repeated game.

Definition 5 (infinite repeated game). For all $\delta \in [0, 1)$, the repeated game $G(T)$ consists of an infinite sequence of repetitions of G with a common discount factor δ . We denote that the definition of a repeated game is

$$G(T) = G(\infty) = (N, S_k, U_k, T, \delta), \quad k \in N, \quad (18)$$

$$T = 0, 1, 2, \dots, t, \dots, \infty.$$

In the repeated game $G(\infty)$, the strategy selected by the participants in stage t is defined as $S_k^t, S_k^t \in S_k$; then, the strategy combination in stage t is defined as $S^t = (S_1^t, S_2^t, \dots, S_k^t)$. Supposed that the benefits of a certain player at each stage are $U_k(S^t)$; the present value of the total return in an infinitely repeated game is obtained by

$$\pi_k = \sum_{t=1}^T \delta^{t-1} U_k(s^t), \quad t \in T, \quad (19)$$

where $\delta \in [0, 1)$ is the discount factor, and the discount factor is the importance of future earnings in the current stage. The discount factor determines a credible threat. This threat makes no participant willing to violate the trigger strategy alone, which follows the principle of Nash equilibrium. The long-term utility is the normalized sum of the discounted expected stage utilities, which can also be called the average return. Assuming that the long-term utility of each stage is $\bar{\pi}_k$, the present value can be expressed as $\bar{\pi}_k/(1 - \delta)$. According to Equation (19), the average return is obtained by

$$\bar{\pi}_k = (1 - \delta) \sum_{t=1}^{\infty} \delta^{t-1} U_k(s^t), \quad t \in T. \quad (20)$$

Theorem 6 (folk theorem). $G(T)$ is an infinite repeated game with G as the stage game, S^* is the Nash equilibrium of stage game G , $U_k(S^t)$ is a set of payment determined by S^* , and $U_k(S)$ is a set of viable payment. For any $k = 1, 2, \dots, N$ satisfying $U_k(S^*) > U_k(S)$, there exists $\delta^* < 1$, so that all $\delta \geq \delta^*$, and S^* is a subgame perfect Nash equilibrium in each round of the game.

The above theorem means that in an infinitely repeated game, if the participants have enough patience (the discount factor satisfies certain conditions), then any feasible payout vector that satisfies individual rationality can be obtained through a specific subgame Nash equilibrium.

5.2. Workflow of the Proposed Incentive Mechanism. In the proposed incentive mechanism, if a worker chooses a low-effort strategy in the t th transaction, she will never be able to participate in such crowdsourcing tasks after

completing this task. The crowdsourcing process with an embedded incentive mechanism is as follows. First of all, the requester publishes the task and the price of the task; then, the worker considers whether to accept the job or not. If she refuses the job, she only has the benefit of self-employment. To accept the job, an available data declaration needs to be submitted. Different contribution strategies have different negative utility, and higher effort will result in higher negative utility.

The requester considers whether to hire the worker for the task based on historical data. If the requester decides to hire the worker, the requester will calculate the corresponding payment after receiving the statement of the worker's contribution. The requester feeds back a series of decisions to the platform. The platform feeds back employment information to workers. Workers who are successfully employed submit real data and are paid when platform validation is passed. The trigger strategies of the two parties in the t th stage are the following.

5.2.1. Requester. If the effort level is greater than 0 in the previous $t-1$ stages and greater than 0 in the t th stage, then the worker will be hired and paid correspondingly in the t th stage; otherwise, it will not be hired.

5.2.2. Worker. Workers accept jobs when the price of the task is greater than the self-employment income. If the payment in the previous $t-1$ stages can compensate for the negative utility and there are additional benefits, the worker chooses to provide a high degree of effort.

The trigger strategy makes both parties have a threat. The threat of the requester is once the worker does not work hard, he will not hire the worker in the next stage. The threat of workers is if the payment is less than the self-employment income, they will not work hard. How much payment should the requester pay to the worker so that it is beneficial to the requester and meets the requirements of the worker? The following is a discussion of payment.

- (1) First, from the perspective of workers, this paper assumes that the total income when workers do not violate the trigger strategy is π_e :

$$\begin{aligned} \pi_e &= \left(P(q_{ij}) - C(q_{ij}) \right) + \delta \times \left(P(q_{ij}) - C(q_{ij}) \right) \\ &\quad + \delta^2 \times \left(P(q_{ij}) - C(q_{ij}) \right) + \dots + \delta^{t-1} \\ &\quad \times \left(P(q_{ij}) - C(q_{ij}) \right) + \dots \\ &= \frac{1}{1-\delta} \times \left(P(q_{ij}) - C(q_{ij}) \right). \end{aligned} \quad (21)$$

Assuming that the worker chooses to violate the trigger strategy in the t period, ρ represents the probability

of workers providing high contribution at the low effort level; the total income is

$$\begin{aligned} \pi_s &= \left(P(q_{ij}) - C(q_{ij}) \right) + \delta \times \left(P(q_{ij}) - C(q_{ij}) \right) + \dots + \delta^{t-2} \\ &\quad \times \left(P(q_{ij}) - C(q_{ij}) \right) + \dots + \delta^{t-1} \times P(q_{ij}) + \delta^t \\ &\quad \times \left(\rho \times P(q_{ij}) + P_0 \times (1-\rho) \right) + \delta^{t+1} \\ &\quad \times \left(\rho \times P(q_{ij}) + P_0 \times (1-\rho) \right) + \dots. \end{aligned} \quad (22)$$

The sufficient condition to drive workers not to violate the trigger strategy is

$$\pi_e - \pi_s > 0. \quad (23)$$

Therefore, the constraint conditions for workers not to violate the trigger strategy are

$$P > P_0 + C(q_{ij}) + \frac{1-\delta}{\delta(1-\rho)} \times C(q_{ij}). \quad (24)$$

- (2) Second, from the perspective of requesters, if the requester chooses not to hire workers, his profit is 0; therefore, the sufficient condition for the requester to follow the trigger strategy in the infinite repeat game is

$$\frac{1}{1-\delta} \left(L(q_{ij}) - P(q_{ij}) \right) > 0. \quad (25)$$

That is,

$$\left(L(q_{ij}) - P(q_{ij}) \right) > 0. \quad (26)$$

Through combining Equations (24) and (26), we get the final result:

$$L(q_{ij}) > P > P_0 + C(q_{ij}) + \frac{1-\delta}{\delta(1-\rho)} \times C(q_{ij}). \quad (27)$$

After the above analysis, inequality is obtained. The payment given under this constraint makes the trigger strategy of both parties constitute a subgame perfect Nash equilibrium. The calculation method of the discount factor in this paper is shown by

$$\delta \geq \delta^* = \frac{C(q_{ij})}{C(q_{ij}) + (1-\rho) \left(L(q_{ij}) - P_0 - C(q_{ij}) \right)}. \quad (28)$$

Input: tasks set A
Output: A new set of tasks Γ with a discount factor
1: **For** $a \in A$ **do**
2: Suppose the triggering strategy is adopted in the first time of infinitely repeated game
3: calculate δ^* the according to Eq.(28)
4: Add the discount factor to the task set
5: **End for**

ALGORITHM 1: Discount factor.

Input: tasks set Γ , workers set W
Output: payoff set P_p
1: Initialize(P_p, σ)
2: **for** $t = 1$ to T **do**
3: **For** $\gamma \in \Gamma$ **do**
4: Posts task information on the platform, including the price tag P_{tag} and contribution q_{normal}
5: Waiting for platform feedback
6: **for** $w \in W$ **do**
7: Calculate the degree of effort σ of worker w by Eq.(6)
8: Traverse worker's history information
9: **If** $\sigma(t) > 0$ and $\sigma(1, \dots, t-1) > 0$ **then**
10: Decide to hire the worker and give feedback to the platform
11: Calculate payoff $P_p(t)$ of worker by Eq.(29)
12: **else** Decide not to hire
13: **end if**
14: **end for**
15: Update (P_p, σ)
16: **end for**

ALGORITHM 2: MSRSG for requesters.

Input : tasks set Γ , workers set W
Output : payoff set P_p
1: **for** $t = 1$ to T **do**
2: **for** $\gamma \in \Gamma$ **do**
3: **If** $P_{tag} > P_0$ **then**
4: Submit the task contribution amount description to the platform
5: **If** worker are not hired **then**
6: Choose another task
7: **else**
8: Traverse worker's history information
9: **If** $P_p(1, \dots, t-1) > P_0(1, \dots, t-1) + C(1, \dots, t-1)$ **then**
10: Choose high level ,and submit real data to the platform
11: **end if**
12: **end if**
13: **else** Choose another task
14: **end if**
15: **end for**
16: **end for**

ALGORITHM 3: MSRSG for workers.

Based on the above analysis, the calculation method of workers' payment is given, where b means the right boundary of different effort levels:

$$P = P_0 + C(q_{ij}) + \frac{1 - \delta}{\delta(1 - \rho)} \times C(q_{ij}) \times (1 + b). \quad (29)$$

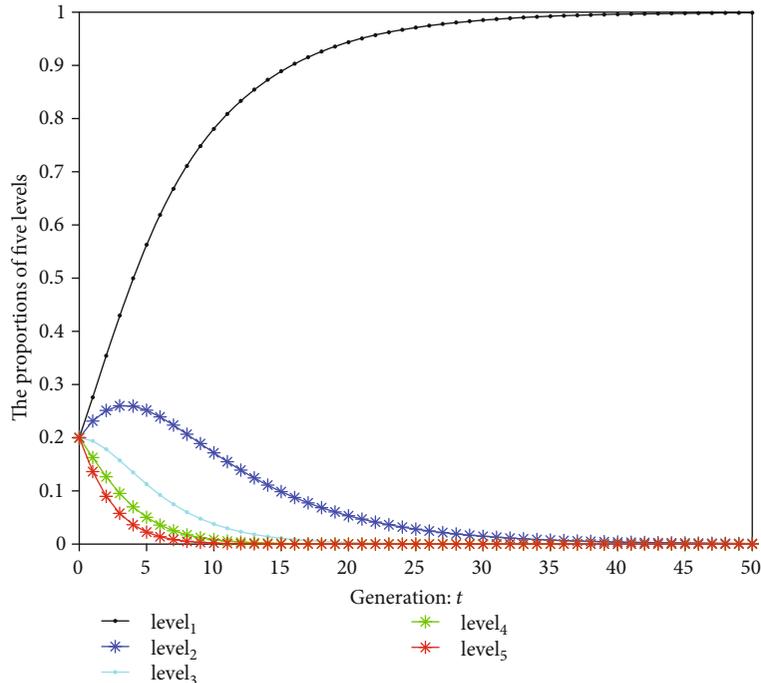


FIGURE 2: The evolution of the worker's strategy under the crowdsourcing dilemma.

Based on the above analysis, this paper proposes a multi-strategy repeated game algorithm—the MSRG algorithm. Algorithms can be implemented in a decentralized manner, which are shown by Algorithms 2 and 3. First, Algorithm 1 is used to process the task set to get the discount factor. Then, from lines 6 to 15 in Algorithm 2, the requester calculates the effort of the workers based on the amount of contribution and decides whether to hire workers and how much they are paid based on historical information. The workers from lines 3 to 12 in Algorithm 3 decide whether to accept the job and determine the degree of effort for this task based on historical information.

6. Simulation Experiment Analysis

In order to verify the effectiveness of the multistrategy repeated game incentive mechanism proposed in this paper, four sets of simulation experiments are designed. The first set of experiments verifies the existing crowdsourcing dilemma. The second set of experiments verifies the optimal discount factor in the range of discount factor satisfying equilibrium. The third set of experiments verifies that the incentive mechanism can effectively solve the existing crowdsourcing dilemma. The fourth set of experiments compares the MSRG incentive mechanism with the simple crowdsourcing mechanism (general crowdsourcing mechanism without incentive mechanism) to illustrate the effectiveness of multistrategy repeated games.

The simulation experiment environment in this paper is the Windows 10 operating system, Intel® Core™ i5-6500 CPU @3.20 GHz 8 GB memory, Matlab2018a, and JetBrains PyCharm 2018.3.3 simulation platform. We simulate the algorithm by setting $\beta = 0.1$, $k = 2$, $q_{\text{normal}} = 8$, and $\zeta = 1$.

Other parameter settings will be introduced separately in each group of simulation experiments.

6.1. Verify the Existence of the Crowdsourcing Dilemma. According to the game payoff matrixes in Tables 2 and 3, we conduct the experiments to verify the crowdsourcing dilemma phenomenon in mobile crowdsourcing systems. In the experiments, we set the initial value of the workers with five levels of effort to account for 0.2, 0.2, 0.2, 0.2, and 0.2, respectively. It can be seen from Figure 2 that the four strategies except level₁ have a short rise, but they all converge to 0 in the end. After 23 iterations, level₁ finally converges to 1. Therefore, the experimental results verify that the workers in the crowdsourcing dilemma have a low level of effort due to selfishness.

Figure 3 presents the evolution of the requester's strategy under the crowdsourcing dilemma, with different worker settings. The proportions of low-effort workers in the experimental setup system are 0.8, 0.7, and 0.4, respectively. One can see that when there are more low-effort workers, no matter what the initial value of the requester who chooses Y , the final strategy of the requester is not to hire. This is because the data provided by the workers cannot meet the needs of the task and cannot bring ideal benefits to the requester. Therefore, the best strategy for the requester is not to hire. This result obviously justifies the dilemma raised in crowdsourcing.

6.2. Optimal Discount Factor. If the discount factor is too large, Equation (27) shows that the current pay is low, which will make the current task less attractive to workers. Therefore, when the discount factor is too large, workers cannot be guided to a high level of effort. If the discount factor is

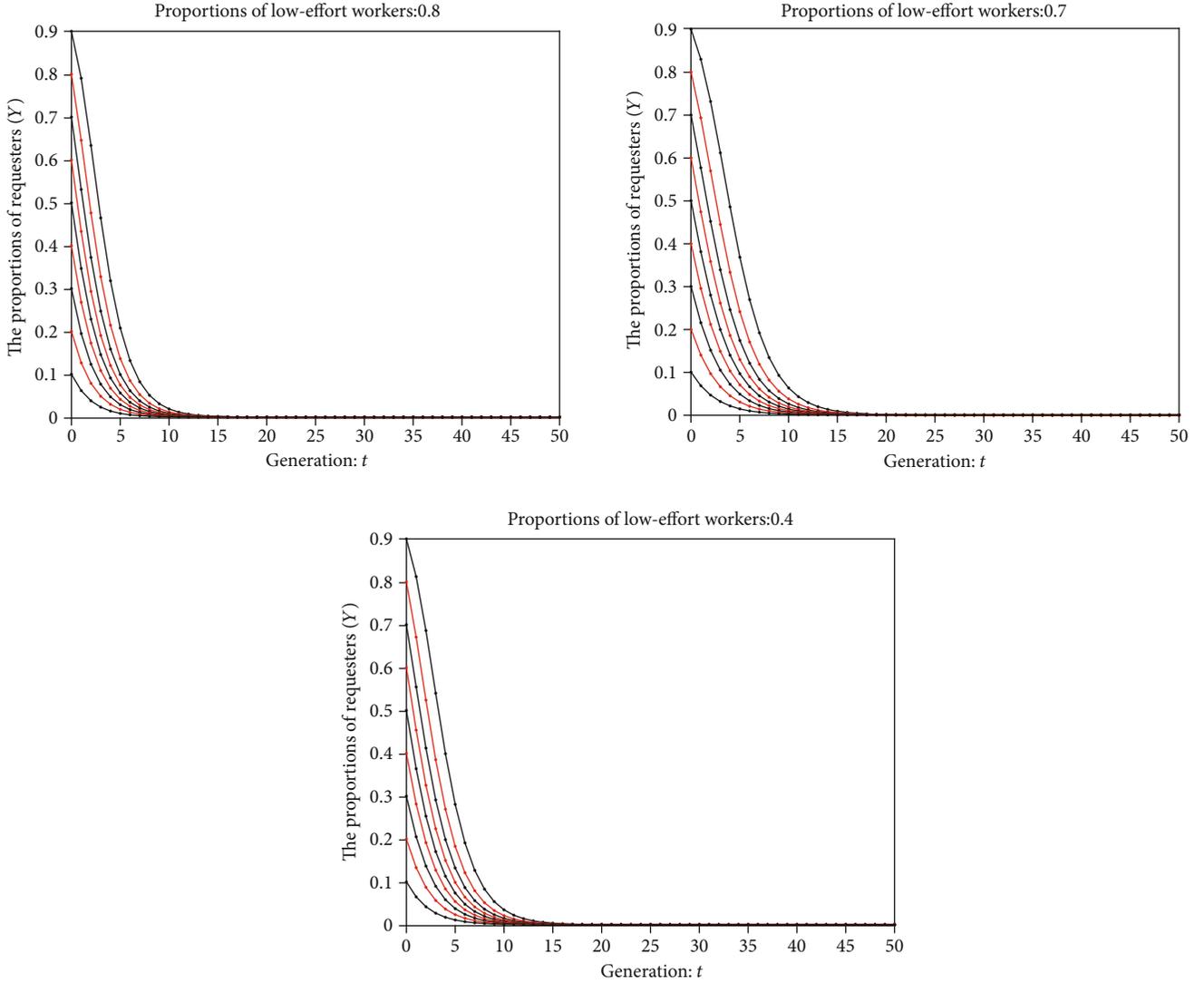


FIGURE 3: The evolution of the requester's strategy under the crowdsourcing dilemma.

too small, or even less than δ^* , then the triggering strategy cannot constitute a perfect Nash equilibrium of subgame, resulting in the failure to solve the crowdsourcing dilemma. In addition, the discount factor is too small to cause workers to not pay attention to future tasks, thus cannot maintain long-term incentive. In order to verify the influence of different discount factors on system performance, the range of discount factors is obtained according to Equation (28). The experiments are carried out to find the optimal discount factor.

The parameter δ^* is set to be 0.7, 0.8, and 0.9, respectively. The experimental results are shown in Figure 4; it shows that when δ^* is 0.7, the population of level₅ converges to 1 fastest. According to Equation (27), it is found that the smaller the discount factor, the higher the reward. Therefore, within the range of satisfying the equilibrium, the smaller the discount factor, the greater the benefits of workers. That is to say, when the discount factor is 0.7, the convergence rate is the fastest.

Figure 5 shows the evolution of the requester strategy under different discount factors as the number of evolutionary generations increases. It can be seen from Figure 4 that when the discount factor is 0.5 and 0.3, the final strategy of the requester is N . This is because the excessively high reward cost cannot bring the desired benefits to the requester, so strategy N is the optimal choice for the requester. From Equation (27), it is obvious that the smaller the discount factor, the higher the current reward given by the requester. It will cause workers to no longer value long term, resulting in an increasing proportion of low-quality workers in the system, which will also cause the requester to eventually converge to strategy N .

6.3. Effectiveness of Incentive Mechanism. In the experiments, we set the initial values of the five worker strategies to account for 0.2, 0.2, 0.2, 0.2, and 0.2, respectively. The discount factor is set to be 0.7. The evolution of the worker

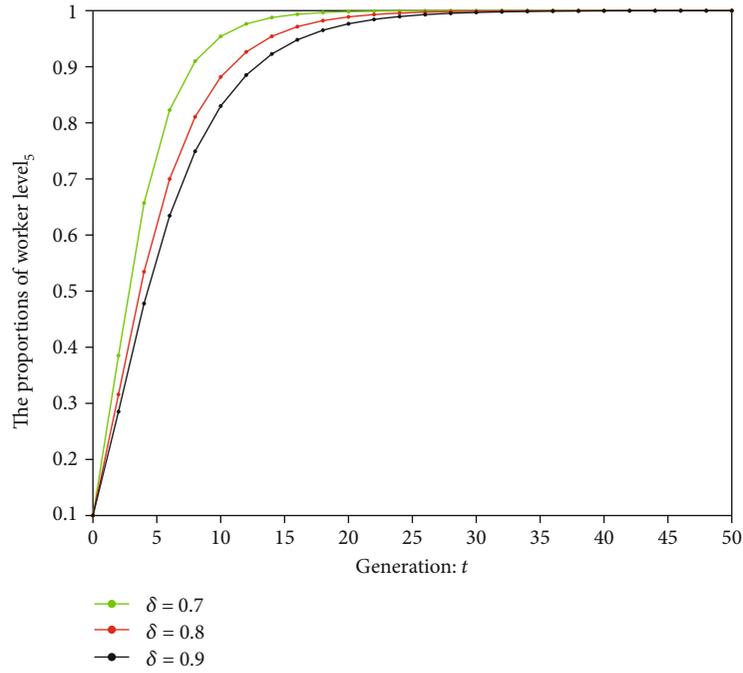


FIGURE 4: The evolution of the worker’s strategy under different discount factors.

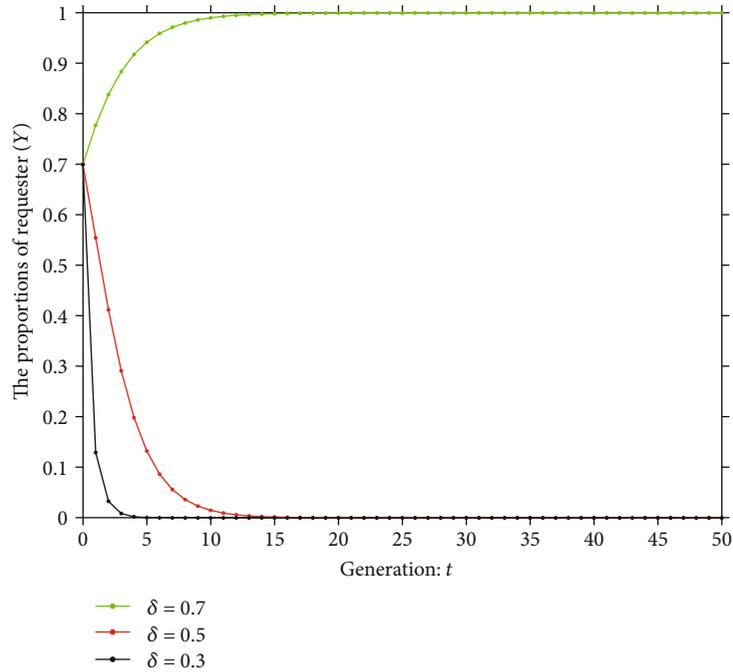


FIGURE 5: The evolution of the requester’s strategy under different discount factors.

and requester under the MSRGI incentive mechanism is presented in Figures 6 and 7.

In Figure 6, one can clearly observe that when incentives are added, the worker population will eventually adopt the level₅ strategy, which shows that the crowdsourcing dilemma has been resolved. This is because the higher the worker’s level of effort,

the higher the task’s payment benchmark and rewards they will receive. Out of individual rationality, workers will adopt the level₅ strategy if they want to maximize their own benefits.

The level of worker’s effort with the incentive mechanism increases continuously, and the change of the requester’s strategy is shown in Figure 7. Figure 7 shows

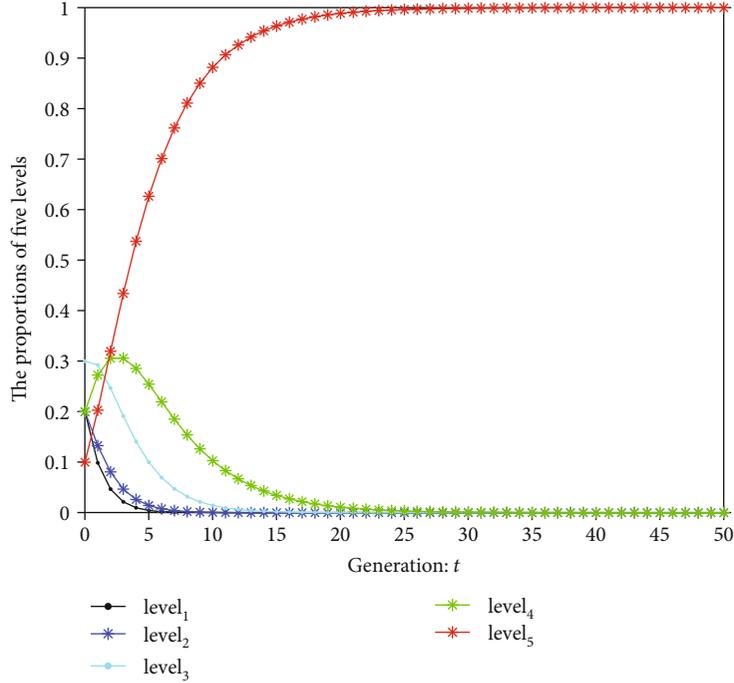


FIGURE 6: The evolution of worker’s strategies under MSRГ incentive mechanism.

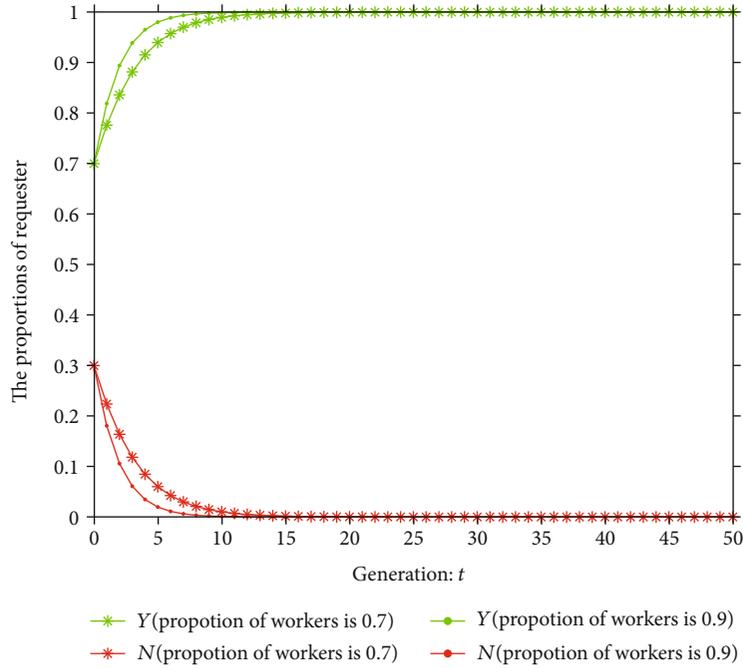


FIGURE 7: The evolution of the requester’s strategy under MSRГ incentive mechanism.

the changes of the requester’s strategies when the proportion of high-effort workers is 0.7 and 0.9, respectively. It can be seen that the higher the proportion of workers with a high-effort level, the faster the requester’s convergence to the employment strategy, and no matter what the initial value of the requester is, it will eventually converge to employment. This is because with the MSRГ incentive

mechanism, the data contribution of workers is kept at a high level, so the requester will gain more by employing the strategy.

6.4. *Budget and Contribution.* The experiment set the proportion of workers of each level in the initial group as 0.2, 0.2, 0.2, 0.2, and 0.2. $\ln(t)$ is a function related to the stage t ,

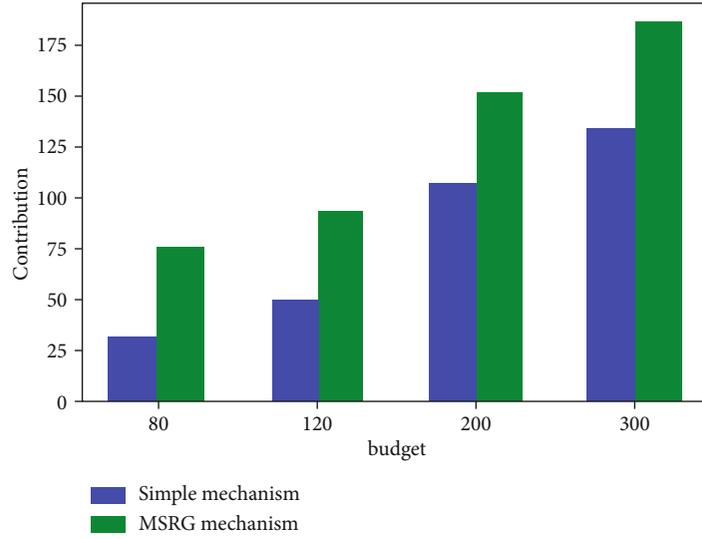


FIGURE 8: Comparison of the contributions of the two mechanisms.

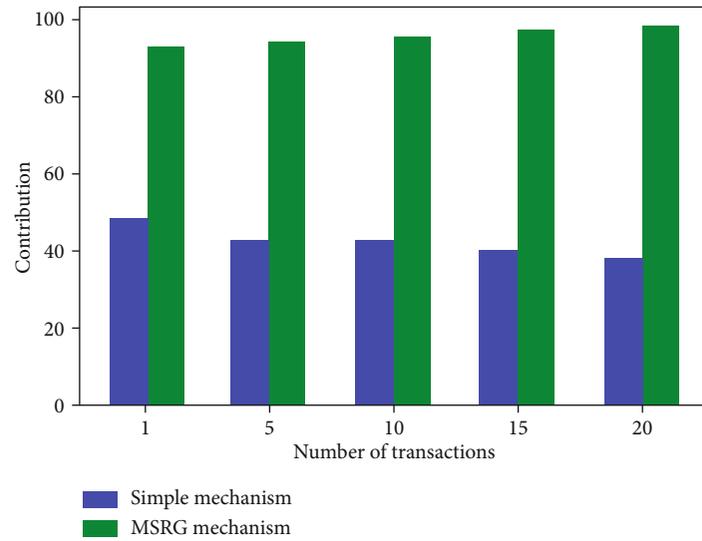


FIGURE 9: Comparison of the contributions of the two mechanisms.

which can be used as an adjustment parameter of the worker's strategy. The contributions of the five levels were $\ln(t) + 3$, $\ln(t) + 5$, $\ln(t) + 8$, $\ln(t) + 8.5$, and $\ln(t) + 9$. According to the evolutionary rules in this mechanism, the number of high-level workers is constantly increasing. The evolution of the worker strategy follows the evolution rules proposed by the MSRG mechanism.

Figure 8 shows the contribution of the two mechanisms under different budgets. It can be seen that under the same budget, requesters in the MSRG incentive mechanism can get more data contributions. This is because the MSRG mechanism can make judgments on the level of workers' effort, thus avoiding the hiring of low-level workers, which means that the payment will not be paid to workers with low contributions.

Assuming a task's budget is 120, the total contributions of the 1st, 5th, 10th, 15th, and 20th tasks are computed in the experiment. Figure 9 describes the amount of data contribution for the same budget in different trading periods. As the number of transactions increases, the amount of contribution received by the requester also increases and remains at a high level after 20 transactions. This is because after 20 transactions have been conducted, the proportion of workers with the highest level of effort in the group could reach 1.

As the number of task transactions increases, the advantages of this mechanism become more obvious. In Figure 10, one can see that as the number of transactions increases, the difference in contribution between the two mechanisms gradually increases. It is obvious that workers need to continuously improve their level of effort to get a higher payment.

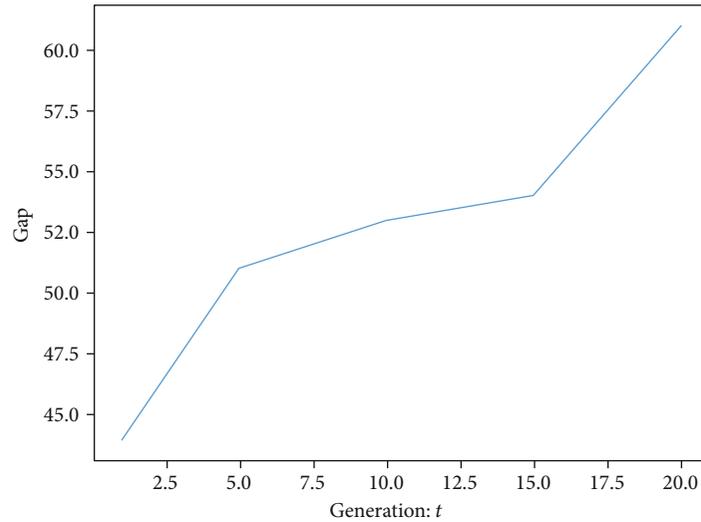


FIGURE 10: Change of the contribution gap.

Therefore, the contribution provided by the workers in the system remains at a high level. This experiment can prove that the MSRSG incentive mechanism could maintain the long term and inspire workers to provide high-quality sensing data.

7. Conclusion

In this paper, we propose an incentive mechanism based on multistrategy repeated games to encourage long-term participation of workers. The discount factor and historical data of repeated games are used to obtain the current optimal behaviors of both sides in the game, and the MSRSG algorithm is proposed to obtain the Nash equilibrium and find the optimal return. When the crowdsourcing dilemma is resolved, the long-term incentive of the system is also guaranteed. This paper also uses an evolutionary game to verify the effectiveness of this mechanism. The proposed MSRSG incentive mechanism could guide workers to continuously improve their level of effort through monetary incentives. The extensive simulation experiments demonstrate that our proposed algorithms can guide workers to maintain a high level of effort and maintain the long-term incentive of mobile crowdsourcing systems.

In future works, we will further investigate other issues related to service quality between transactions in MEC, such as privacy protection issues. Based on game theory, we further improve the incentive mechanism to solve the contradiction between privacy protection loss and service quality [56].

Data Availability

In order to verify the effectiveness of the multistrategy repeated game incentive mechanism proposed in this paper, four sets of simulation experiments are designed. Specific experimental parameter settings can be found in Simulation Experiment Analysis.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant 62072392, Grant 61822602, Grant 61772207, Grant 61802331, Grant 61602399, Grant 61702439, Grant 61773331, and Grant 62062034; the China Postdoctoral Science Foundation under Grant 2019T120732 and Grant 2017M622691; the Natural Science Foundation of Shandong Province under Grant ZR2016FM42; the Major scientific and technological innovation projects of Shandong Province under Grant 2019JZZY020131; and the Key projects of Shandong Natural Science Foundation under Grant NO. ZR2020KF019.

References

- [1] X. Xu, R. Mo, F. Dai, W. Lin, S. Wan, and W. Dou, "Dynamic resource provisioning with fault tolerance for data-intensive meteorological workflows in cloud," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 9, pp. 6172–6181, 2020.
- [2] L. Qi, C. Hu, X. Zhuang et al., "Privacy-aware data fusion and prediction with spatial-temporal context for smart city industrial environment," *IEEE Transactions on Industrial Informatics*, p. 1, 2020.
- [3] C. Chen, Y. Zhang, M. R. Khosravi, Q. Pei, and S. Wan, "An intelligent platooning algorithm for sustainable transportation systems in smart cities," *IEEE Sensors Journal*, p. 1, 2020.
- [4] W. Zhong, X. Yin, X. Zhang et al., "Multi-dimensional quality-driven service recommendation with privacy-preservation in mobile edge environment," *Computer Communications*, vol. 157, pp. 116–123, 2020.
- [5] L. Qi, Q. He, F. Chen, X. Zhang, W. Dou, and Q. Ni, "Data-driven web APIs recommendation for building web applications," *IEEE Transactions on Big Data*, p. 1, 2020.
- [6] Q. Duan, S. Wang, and N. Ansari, "Convergence of networking and cloud/edge computing: status, challenges, and

- opportunities,” *IEEE Network*, vol. 34, no. 6, pp. 148–155, 2020.
- [7] Y. Wang, Z. Cai, X. Tong, Y. Gao, and G. Yin, “Truthful incentive mechanism with location privacy-preserving for mobile crowdsourcing systems,” *Computer Networks*, vol. 135, pp. 32–43, 2018.
- [8] <https://answers.yahoo.com/2020>.
- [9] H. Hong, X. Li, D. He, Y. Zhang, and M. Wang, “Crowdsourcing incentives for multi-hop urban parcel delivery network,” *IEEE Access*, vol. 7, pp. 26268–26277, 2019.
- [10] S. Zhu, Z. Cai, H. Hu, Y. Li, and W. Li, “zkCrowd: a hybrid blockchain-based crowdsourcing platform,” *IEEE Transactions on Industrial Informatics*, vol. 16, no. 6, pp. 4196–4205, 2020.
- [11] J. Li, Z. Cai, M. Yan, and Y. Li, “Using crowdsourced data in location-based social networks to explore influence maximization,” in *IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications*, pp. 1–9, San Francisco, CA, 2016.
- [12] T. Liu, Y. Wang, Y. Li, X. Tong, L. Qi, and N. Jiang, “Privacy protection based on stream cipher for spatiotemporal data in IoT,” *IEEE Internet of Things Journal*, vol. 7, no. 9, pp. 7928–7940, 2020.
- [13] Y. Wang, Z. Cai, G. Yin, Y. Gao, X. Tong, and G. Wu, “An incentive mechanism with privacy protection in mobile crowdsourcing systems,” *Computer Networks*, vol. 102, pp. 157–171, 2016.
- [14] Y. Wu, J. Zeng, H. Peng, H. Chen, and C. Li, “Survey on incentive mechanisms for crowd sensing,” *Journal of Software*, vol. 27, pp. 2025–2047, 2016.
- [15] J. Li, Z. Cai, J. Wang, M. Han, and Y. Li, “Truthful incentive mechanisms for geographical position conflicting mobile crowdsensing systems,” *IEEE Transactions on Computational Social Systems*, vol. 5, no. 2, pp. 324–334, 2018.
- [16] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, “A survey on mobile edge computing: The communication perspective,” *IEEE Communications Surveys & Tutorials*, vol. 19, no. 4, pp. 2322–2358, 2017.
- [17] T. Shi, Z. Cai, J. Li, and H. Gao, “CROSS: a crowdsourcing based sub-servers selection framework in D2D enhanced MEC architecture,” in *The 40th IEEE International Conference on Distributed Computing Systems (ICDCS 2020)*, Singapore, 2020.
- [18] J. Lu, Y. Xin, Z. Zhang, S. Tang, C. Tang, and S. Wan, “Extortion and cooperation in rating protocol design for competitive crowdsourcing,” *IEEE Transactions on Computational Social Systems*, p. 1, 2020.
- [19] L. Zhang, Y. Ding, X. Wang, and L. Guo, “Conflict-aware participant recruitment for mobile crowdsensing,” *IEEE Transactions on Computational Social Systems*, vol. 7, no. 1, pp. 192–204, 2020.
- [20] A. Wang, M. Ren, H. Ma, L. Zhang, P. Li, and L. Guo, “Maximizing user type diversity for task assignment in crowdsourcing,” *Journal of Combinatorial Optimization*, vol. 40, no. 4, pp. 1092–1120, 2020.
- [21] Y. Wang, G. Yin, Z. Cai, Y. Dong, and H. Dong, “A trust-based probabilistic recommendation model for social networks,” *Journal of Network and Computer Applications*, vol. 55, pp. 59–67, 2015.
- [22] S. K. Goudos, Z. D. Zaharis, and K. B. Baltzis, “Particle swarm optimization as applied to electromagnetic design problems,” *International Journal of Swarm Intelligence Research*, vol. 9, no. 2, pp. 47–82, 2018.
- [23] L. Wang, X. Zhang, R. Wang, C. Yan, H. Kou, and L. Qi, “Diversified service recommendation with high accuracy and efficiency,” *Knowledge-Based Systems*, vol. 204, p. 106196, 2020.
- [24] Z. Cai and T. Shi, “Distributed query processing in the edge assisted IoT data monitoring system,” *IEEE Internet of Things Journal*, p. 1, 2020.
- [25] Y. Zhang, H. Qin, B. Li, J. Wang, S. Lee, and Z. Huang, “Truthful mechanism for crowdsourcing task assignment,” *Tsinghua Science and Technology*, vol. 23, no. 6, pp. 645–659, 2018.
- [26] Y. Wang, Y. Gao, Y. Li, and X. Tong, “A worker-selection incentive mechanism for optimizing platform-centric mobile crowdsourcing systems,” *Computer Networks*, vol. 171, p. 107144, 2020.
- [27] J. Xu, S. Wang, N. Zhang, F. Yang, and X. Shen, “Reward or penalty: aligning incentives of stakeholders in crowdsourcing,” *IEEE Transactions on Mobile Computing*, vol. 18, no. 4, pp. 974–985, 2019.
- [28] J. Lu, Z. Zhang, J. Wang, R. Li, and S. Wan, *A Green Stackelberg-Game Incentive Mechanism for Multi-Service Exchange in Mobile Crowdsensing*, ACM ToIT, 2020.
- [29] J. Liu, W. Wang, D. Li, S. Wan, and H. Liu, “Role of gifts in decision making: an endowment effect incentive mechanism for offloading in the IoV,” *IEEE Internet of Things Journal*, vol. 6, no. 4, pp. 6933–6951, 2019.
- [30] Y. Wang, Z. Cai, Z. Zhan, B. Zhao, X. Tong, and L. Qi, “Walrasian equilibrium-based multiobjective optimization for task allocation in mobile crowdsourcing,” *IEEE Transactions on Computational Social Systems*, vol. 7, no. 4, pp. 1033–1046, 2020.
- [31] J. Xu, S. Wang, B. K. Bhargava, and F. Yang, “A blockchain-enabled trustless crowd-intelligence ecosystem on mobile edge computing,” *IEEE Transactions on Industrial Informatics*, vol. 15, no. 6, pp. 3538–3547, 2019.
- [32] Z. Duan, W. Li, X. Zheng, and Z. Cai, “Mutual-preference driven truthful auction mechanism in mobile crowdsensing,” in *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*, pp. 1233–1242, Dallas, TX, USA, 2019.
- [33] Y. Wang, Z. Cai, Z. Zhan, Y. Gong, and X. Tong, “An optimization and auction-based incentive mechanism to maximize social welfare for mobile crowdsourcing,” *IEEE Transactions on Computational Social Systems*, vol. 6, no. 3, pp. 414–429, 2019.
- [34] S. Samad and S. S. Kanhere, “MRA: a modified reverse auction based framework for incentive mechanisms in mobile crowdsensing systems,” *Computer Communications*, vol. 145, pp. 137–145, 2019.
- [35] H. Zhou, X. Chen, S. He, J. Chen, and J. Wu, “DRAIM: a novel delay-constraint and reverse auction-based incentive mechanism for WiFi offloading,” *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 4, pp. 711–722, 2020.
- [36] Y. Hu, Y. Wang, Y. Li, and X. Tong, “An incentive mechanism in mobile crowdsourcing based on multi-attribute reverse auctions,” *Sensors*, vol. 18, no. 10, p. 3453, 2018.
- [37] Z. Cai, Z. Duan, and W. Li, “Exploiting multi-dimensional task diversity in distributed auctions for mobile crowdsensing,” *IEEE Transactions on Mobile Computing*, p. 1, 2020.
- [38] Z. Duan, W. Li, and Z. Cai, “Distributed auctions for task assignment and scheduling in mobile crowdsensing systems,”

- in *2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*, pp. 635–644, Atlanta, GA, 2017.
- [39] T. Luo, S. S. Kanhere, S. K. Das, and H. Tan, “Incentive mechanism design for heterogeneous crowdsourcing using all-pay contests,” *IEEE Transactions on Mobile Computing*, vol. 15, no. 9, pp. 2234–2246, 2016.
- [40] W. Wu, W. Wang, M. Li et al., “Incentive mechanism design to meet task criteria in crowdsourcing: how to determine your budget,” *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 2, pp. 502–516, 2017.
- [41] R. Lan, Y. Zhou, Z. Liu, and X. Luo, “Prior knowledge-based probabilistic collaborative representation for visual recognition,” *IEEE Transactions on Cybernetics*, vol. 50, no. 4, pp. 1498–1508, 2020.
- [42] Y. Zhan, Y. Xia, and J. Zhang, “Incentive mechanism in platform-centric mobile crowdsensing: a one-to-many bargaining approach,” *Computer Networks*, vol. 132, pp. 40–52, 2017.
- [43] Y. Zhang and D. S. M. Van, “Reputation-based incentive protocols in crowdsourcing applications,” in *2012 Proceedings IEEE INFOCOM*, pp. 2140–2148, Orlando, FL, 2012.
- [44] Y. Wang, Y. Li, Z. Chi, and X. Tong, “The truthful evolution and incentive for large-scale mobile crowd sensing networks,” *IEEE Access*, vol. 6, pp. 51187–51199, 2018.
- [45] X. Zhang, G. Xue, R. Yu, D. Yang, and J. Tang, “Keep your promise: mechanism design against free-riding and false-reporting in crowdsourcing,” *IEEE Internet of Things Journal*, vol. 2, pp. 562–572, 2017.
- [46] L. Gao, F. Hou, and J. Huang, “Providing long-term participation incentive in participatory sensing,” in *2015 IEEE Conference on Computer Communications (INFOCOM)*, pp. 2803–2811, Kowloon, 2015.
- [47] L. P. Gao, T. Jin, and C. Lu, “A long-term quality perception incentive strategy for crowdsourcing environments with budget constraints,” *International Journal of Cooperative Information Systems*, vol. 29, pp. 901–914, 2020.
- [48] G. J. Mailath, I. Obara, and T. Sekiguchi, “The maximum efficient equilibrium payoff in the repeated prisoners’ dilemma,” *Games and Economic Behavior*, vol. 40, pp. 99–122, 2001.
- [49] L. Gao, N. Ruan, and H. Zhu, “Efficient and secure message authentication in cooperative driving: a game-theoretic approach,” in *2016 IEEE International Conference on Communications (ICC)*, pp. 1–6, Kuala Lumpur, 2016.
- [50] Q. Hu, S. Wang, P. Ma, X. Cheng, W. Lv, and R. Bie, “Quality control in crowdsourcing using sequential zero-determinant strategies,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 32, no. 5, pp. 998–1009, 2020.
- [51] L. Yin, Y. Guo, F. Li, Y. Sun, J. Qian, and A. Vasilakos, “A game-theoretic approach to advertisement dissemination in ephemeral networks,” *World Wide Web*, vol. 21, no. 2, pp. 241–260, 2018.
- [52] X. Liu, B. Cheng, and S. Wang, “Availability-aware and energy-efficient virtual cluster allocation based on multi-objective optimization in cloud datacenters,” *IEEE Transactions on Network and Service Management*, vol. 17, no. 2, pp. 972–985, 2020.
- [53] G. Yin, Y. Wang, Y. Dong, and H. Dong, “Wright-Fisher multi-strategy trust evolution model with white noise for Internetware,” *Expert Systems with Applications*, vol. 40, no. 18, pp. 7367–7380, 2013.
- [54] B. Wang, K. J. R. Liu, and T. C. Clancy, “Evolutionary cooperative spectrum sensing game: how to collaborate?,” *IEEE Transactions on Communications*, vol. 58, no. 3, pp. 890–900, 2010.
- [55] L. Fang, G. Shi, L. Wang, Y. Li, S. Xu, and Y. Guo, “Incentive mechanism for cooperative authentication: an evolutionary game approach,” *Information Sciences*, vol. 527, pp. 369–381, 2020.
- [56] Z. Cai and Z. He, “Trading private range counting over big IoT data,” in *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*, pp. 144–153, Dallas, TX, USA, 2019.