Quality Control in Crowdsourcing Using Sequential Zero-Determinant Strategies

Qin Hu, Shengling Wang, *Member, IEEE*, Peizi Ma, Xiuzhen Cheng, *Fellow, IEEE*, Weifeng Lv, and Rongfang Bie, *Member, IEEE*

Abstract—Quality control in crowdsourcing is challenging due to the heterogeneous nature of the workers. The state-of-the-art solutions attempt to address the issue from the technical perspective, which may be costly because they function as an additional procedure in crowdsourcing. In this paper, an economics based idea is adopted to embed quality control into the crowdsourcing process, where the requestor can take advantage of the market power to stimulate the workers for submitting high-quality jobs. Specifically, we employ two sequential games to model the interactions between the requestor and the workers, with one considering binary strategies while the other taking continuous strategies. Accordingly, two incentive algorithms for improving the job quality are proposed to tackle the sequential crowdsourcing dilemma problem. Both algorithms are based on a sequential zero-determinant (ZD) strategy modified from the classical ZD strategy. Such a revision not only provides a theoretical basis for designing our incentive algorithms, but also enlarges the application space of the classical ZD strategy itself. Our incentive algorithms have the following desired features: 1) they do not depend on any specific crowdsourcing scenario; 2) they leverage economics theory to train the workers to behave nicely for better job quality instead of filtering out the unprofessional workers; 3) no extra costs are incurred in a long run of crowdsourcing; and 4) fairness is realized as even the requestor (the ZD player), who dominates the game, cannot increase her utility by arbitrarily penalizing any innocent worker.

Index Terms—Crowdsourcing, quality control, sequential game, zero-determinant strategies

1 Introduction

The advent of crowdsourcing has created new opportunities that can facilitate the accomplishment of labor-intensive jobs that are difficult for machines alone [1], [2], [3], [4]. In crowdsourcing, a crowdsourcer (requestor) recruits multiple crowdsourcees (workers) online to complete jobs that can be easily done through gathering dispersed human resources. However, workers recruited through crowdsourcing usually have different skills, intents, and backgrounds. This heterogeneous nature leads to the diverse submission quality of the completed tasks, pressing an urgent need for quality control.

The state-of-the-art quality control in crowdsourcing can be categorized into two classes: worker-based [5], [6], [7], [8], [9], [10], [11], [12], [13] and job-based [14], [15], [16], [17], [18], with the former considering that the quality of a completed task is closely related to the workers and hence the job quality can be guaranteed through selecting good (professional) workers, and the latter developing different methodologies or tools to directly evaluate or monitor the job quality. These existing methods might be costly in terms of operation, because they usually function as an additional procedure in crowdsourcing.

In this paper, we take a drastically different approach in which quality control is embedded in a crowdsourcing process itself. Our novel approach is enlightened by the essence of paid crowdsourcing, which is actually a trade between a requestor and the workers. Hence, even though a worker can reduce his[1] cost through offering low-quality contributions or even behave maliciously to get filthy lucre, the requestor could punish such behaviors by lowering payment. Thus, through a suitable pricing scheme according to the job quality, the requestor can take advantage of the market power to stimulate workers to submit high-quality jobs. This method is not restricted to any specific crowdsourcing scenario; instead, it is an economic means proposed according to the common trait of all crowdsourcing scenarios, which renders it suitable for an extensive application domain.

Another outstanding property of our quality control approach lies in that it does not filter out unprofessional workers; instead, it trains the workers. More specifically, we use a monetary reward or penalty to encourage or force the workers to behave nicely, turning bad guys into good ones.

- *Q. Hu is with the College of Information Science and Technology, Beijing Normal University, Beijing 100875, China, and the Department of Computer Science, The George Washington University, Washington DC 20052. E-mail: qinhu@gwu.edu.*
- *S. Wang, P. Ma, and R. Bie are with the College of Information Science and Technology, Beijing Normal University, Beijing 100875, China. E-mail: {wangshengling, rfbie}@bnu.edu.cn, mapeizi@mail.bnu.edu.cn.*
- *X. Cheng is with the Department of Computer Science, The George Washington University, Washington, DC 20052. E-mail: cheng@gwu.edu.*
- *W. Lv is with the School of Computer Science and Engineering, Beihang University, Beijing 100191, China. E-mail: lwf@buaa.edu.cn.*

Manuscript received 16 May 2018; revised 27 Dec. 2018; accepted 28 Jan. 2019. Date of publication 1 Feb. 2019; date of current version 1 Apr. 2020. (Corresponding author: Shengling Wang.)
Recommended for acceptance by R. Cheng.
Digital Object Identifier no. 10.1109/TKDE.2019.2896926

1. In this paper, we denote the requestor as "she" and a worker as "he" for easy differentiation.

Compared to the existing worker selection approach, such a mechanism can call on all possible human resources, which is also a key trait of crowdsourcing that results in its success. However, this seemingly simple method needs to address the constraint problem of incentive cost. In this paper, we aim at achieving the following challenging objective: *incentivizing a worker to submit high-quality jobs without a long-term extra payment.*

To that aim, we make use of the zero-determinant (ZD) strategy [19] and revise it to obtain a variant that is suitable for a sequential game adapting to our scenario, in which the requestor acts first and the worker reacts according to the requestor's strategy. ZD is a probabilistic and conditional strategy whose adopter (i.e., the ZD player) can unilaterally set the opponent's absolute or relative expected payoff. By virtue of a sequential ZD strategy, we propose two incentive algorithms in this paper that can improve the job quality of the workers. Our contributions are summarized as follows:

- We employ two sequential games to model the interactions between the requestor and a worker when their strategies are either binary or continuous. We also prove that the sequential crowdsourcing dilemma phenomenon exists in both cases. Note that players may take strategies from their corresponding discrete spaces; but this can be treated as special cases of the continuous-strategy model.
- The original zero-determinant strategy is modified in this paper to get a variant applicable to sequential games. Such a variant (i.e., the sequential zero-determinant strategy) not only provides a theoretical basis for designing our incentive algorithms, but also enlarges the application domain of the original zero-determinant strategy.
- Based on the sequential zero-determinant strategy, two incentive algorithms for improving the job quality of the workers are designed when both players take either binary or continuous strategies. The simulation results demonstrate that the proposed algorithms can encourage the workers to become cooperative without a long-term extra payment. More importantly, such an incentive mechanism is fair because even though the requestor dominates the game, she cannot increase her utility by arbitrarily penalizing innocent workers.

The rest of the paper is organized as follows. The most related work is summarized in Section 2. Section 3 investigates the sequential dilemmas in crowdsourcing when the strategies are either binary or continuous. The original zero-determinant strategy is extended to the sequential scenario in Section 4. We propose two incentive algorithms in Sections 5 and 6 respectively considering the cases of binary and continuous strategies. Simulation results of our algorithms are reported in Section 7 and we conclude this paper in Section 8.

2 Related Work

As mentioned earlier, existing studies on quality control in crowdsourcing can be classified into two categories: worker-based and job-based.

Since the quality of a crowdsourcing job is closely related to the recruited workers, worker-based quality control regulates the quality of a crowdsourced job by identifying the characteristics of the workers [5], [6], [7], [8], [9], [10], [11], [12], [13]. This can be done by either considering quality detection and worker selection seperately [5], [6] or combining these two processes to filter out low-quality workers at the very beginning [7], [8]. Wang et al. [5] utilized the random forest model, a classic machine learning method, to build a filter that was experimentally proved to have more than 95 percent of accuracy for detecting workers' quality; Xuan et al. [6] employed a number of machine learning algorithms and proposed their own adversarial detecting framework, in which they found that the support vector machine (SVM) model outperforms all the rest. Folorunso et al. [7] took advantage of the trust-based access control (TBAC) and fuzzy-expert systems to develop a TBAC-fuzzy algorithm for filtering out the malicious workers in crowdsourcing computing. A trust evaluation model was constructed in [8], by which the trustworthy worker selection process was converted to a multi-objective combinatorial optimization problem and finally resolved by an evolutionary algorithm. In [9], Wang et al. utilized the historical information of the workers to accurately estimate their long-term quality in a dynamic manner, based on which they further proposed a reverse auction based incentive mechanism. In [10], a knowledge-based domain estimation approach was proposed to model the quality of workers, based on which an online task assignment algorithm was developed to appropriately match tasks and workers. To better describe the behavior of workers in crowdsourcing, Cao et al. [11] formulated two versions of the Jury Selection Problem (JSP) and proposed efficient algorithms to solve them. Focusing on crowdsourced filtering and rating tasks, Das Sarma et al. [12] developed a novel pruning and search-based approach to find the global maximum likelihood solution for estimating the ground truth and worker quality. To adaptively evaluate diverse accuracies of a worker on various tasks, Fan et al. [13] proposed a crowdsourcing framework, termed *iCrowd*, which could assign the tasks to the most appropriate workers on the fly.

On the other hand, observing that the quality of a job directly affects the quality of the whole crowdsourcing task, researchers aimed at achieving job-based quality control by employing different approaches or criteria to evaluate the job quality [14], [15], [16], [17], [18]. Based on the classical gold standard with fixed gold units and the corresponding answers, Oleson et al. [17] put forward a programmatic gold creation mechanism to automatically generate new gold units with known answers so as to assure each submitted job's quality. In [14], a crowdsourcing quality control model was proposed for parallelly distributed jobs by assessing the quality of each worker's contribution. Vuurens et al. [15] proposed an approach to detect spammers for the crowdsourced relevance learning problem of information retrieval (IR) systems, by comparing the obtained value with an expert value. In [16], Zheng et al. devised a quality-aware job assignment scheme in crowdsourcing with known ground truth of each question. Qiu et al. [18] considered the feedback from other workers and/or the public as a quality factor, and utilized it to

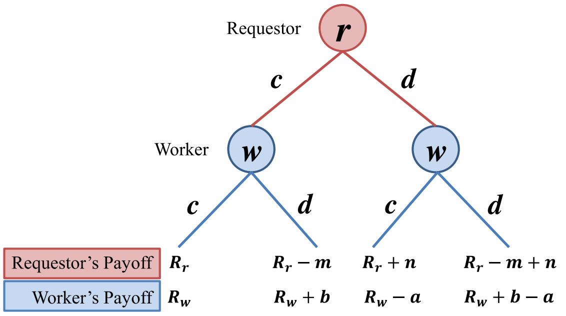

Fig. 1. The game tree of the sequential game between the requestor and the worker in one round under the binary model.

propose a Stackelberg game based scheme for heterogeneous contract design.

Note that most existing schemes require an additional component such as the worker selection procedure in worker-based approaches or the job evaluation procedure in job-based methods to guarantee the crowdsourcing quality. In contrast, our work aims to embed quality control into the crowdsourcing process, which does not need any extra supporting mechanism.

3 Game Formulation

In this section, we present our system model for the sequential game between a requestor and a worker.

We focus on the following multi-round crowdsourcing scenario with one requestor and multiple workers. To launch one round of crowdsourcing, the requestor first presents the required jobs and the corresponding payments; then the workers choose what they can accomplish so as to obtain the corresponding payments. Within each round, the requestor can assign different payments to the same job while the workers can choose to make different levels of effort to accomplish the jobs after witnessing the payments, resulting in various job quality. More seriously, there may exist extreme circumstances where the workers may behave maliciously to get filthy lucre such as submitting fabricated data to the requestor and/or attacking other peering workers to steal their data or even destroy their outcomes. The interactions between the workers and the requestor under the above scenario can be depicted by a sequential game. When the requestor recruits the same worker for multiple rounds, this game becomes an iterated one. Note that from the perspective of the requestor, the actions of the workers are similar or follow the same pattern; thus a successful incentive to any worker implies a high possibility of a large-scale success when it is applied to all workers. In light of this, we focus on analyzing the game between the requestor and an arbitrary worker in this paper.

In the iterated sequential game mentioned above, the strategy of the requestor is the amount of payment she offers to the worker for a specific task, and that of the worker is the quality of the claimed job he achieves. Obviously, the strategies of both the requestor and the worker could be discrete (selected from a discrete space) or continuous (selected from a continuous space). Nevertheless, we only consider the binary and continuous cases as the discrete case can be treated as a special one of the continuous case. When the strategies are binary, the players perform either extremely friendly or extremely maliciously.

3.1 Binary Model

In the binary model, the requestor offers either the highest or the lowest payment for the same job, and the worker decides to provide the highest or the lowest quality when completing the job. We denote the strategy of the requestor as $\tilde{x} \in \{c, d\}$, where c indicates her cooperation behavior of providing the highest payment to the worker and d refers to the defection behavior with the lowest payment. Similarly, the worker's strategy is denoted as $\tilde{y} \in \{c, d\}$, where c and d respectively refer to cooperation and defection, i.e., providing either the highest or the lowest job quality.

Since the requestor moves first and the worker makes his decision later, we can depict their sequential interactions in one round as a game tree (see Fig. 1). Denote the payoff vector of the requestor (r) by $\mathbf{S}_r = (R_r, R_r + n, R_r - m, R_r - m + n)$ and that of the worker (w) by $\mathbf{S}_w = (R_w, R_w - a, R_w + b, R + b - a)$, where R_r and R_w are respectively the normal payoffs of the requestor and the worker when they both cooperate; n is the increase of the requestor's normal payoff and a is the reduction of the worker's normal payoff when the requestor defects while the worker cooperates; similarly, when the worker defects while the requestor cooperates, the worker gets an increment b on his normal payoff and the requestor receives a payoff decrement m. Note that $n < m$ and $b < a$ since the requestor's lowest payment and the worker's lowest job quality should result in less payoff for both players when compared with the case of mutual cooperation.

It is obvious that no matter what the requestor's strategy is, the worker's best strategy is d; with backward induction [20], the requestor can also derive her best strategy d. Thus, the only equilibrium of the above sequential game is (d, d), where the payoffs of both players are obviously less than those in state (c, c). This comes into the sequential crowdsourcing dilemma.

3.2 Continuous Model

When the continuous-strategy model is adopted, the requestor can choose to offer any amount of payment in her strategy space while the worker can provide any job quality in his strategy domain. Here we denote the strategy of the requestor (r) as $x \in [l_r, h_r]$ and that of the worker (w) as $y \in [l_w, h_w]$, where l_r, l_w and h_r, h_w are respectively the lower and upper bounds of the strategy spaces of the players, i.e., the lowest/highest payment that the requestor can offer and the lowest/highest job quality that the worker can provide.

Hence, the utility of the requestor can be defined as

$$w_r(x, y) = A_r \phi(y) - B_r x, \tag{1}$$

where the first term is the profit she can obtain from the job completed by the worker while the second term reflects the payment she makes; $A_r > 0$ and $B_r > 0$ are scaling parameters; and $\phi(y)$ is monotonically increasing with the worker's job quality y.

With a similar structure, the utility of the worker can be defined as

$$w_w(x, y) = A_w x - B_w \psi(y), \tag{2}$$

where the first term is the payment he can obtain by completing a specific job while the second one represents his cost for accomplishing the job; $A_w > 0$ and $B_w > 0$ are scaling parameters; and $\psi(y)$ is positively proportional to y since a higher quality implies more effort needed.

Theorem 3.1. *The sequential crowdsourcing dilemma exists in the continuous-strategy model.*

Proof. Since $\psi(y)$ increases as y increases, we have $\psi'(y) > 0$. According to (2), we get $\frac{\partial w_w(x,y)}{\partial y} = -B_w \psi'(y)$. Combining with the condition of $B_w > 0$, we obtain $\frac{\partial w_w(x,y)}{\partial y} < 0$, which means that w_w is inversely proportional to y. Thus, the worker's best strategy is $y^* = l_w$.

Similarly, considering that $B_r > 0$, we have $\frac{\partial w_r(x,y^*)}{\partial x} = -B_r < 0$, which implies that the requestor's best strategy with respect to y^* is $x^* = l_r$. Thus, the stable equilibrium of the continuous sequential game is $(x^*, y^*) = (l_r, l_w)$, where the payoffs of both players are less than those in the state of $(x, y) = (h_r, h_w)$. Therefore a sequential crowdsourcing dilemma exists under the continuous-strategy model. □

Remark. Note that it is possible for the two players to take discrete strategies within their personalized strategy spaces that can be heterogeneous to each other due to various decision preferences of the game players. Nevertheless, we claim that this can be regarded as a special case of the continuous model for the following two reasons: i) both the upper and the lower bounds of the two players' continuous strategies can be solely decided by the players themselves; and ii) a continuous strategy space can be discretized to a customized discrete one according to the player's personalized requirements. In this case, the game analysis and algorithm design for the continuous model in sequel can be easily revised to adapt to the heterogeneous discrete-strategy situation. So we omit the detailed description for brevity and for avoiding redundancy.

4 Extension of the Sequential Zero-Determinant Strategy

According to our analysis in Section 3, one can conclude that sequential crowdsourcing dilemmas always exist whether the strategies are binary or continuous. In the short term, a dilemma can lead to low utilities of both players; while in the long run, it can result in low efficiency and low effectiveness of the whole crowdsourcing system or even breakdown the system. To address this problem, it is reasonable for the requestor to work out a scheme that can encourage the worker to be cooperative. Here are the two underlying reasons: first, the requestor is dominant in the game since she has a global observation on all the participating workers and thus can estimate their strategies from a statistical perspective; second, the requestor moves first in the sequential game, which can influence the actions of the workers. As we all know, players in a game are utility-driven; thus, if the requestor wants to drive the worker cooperate, she has to make the worker perceive the positive correlation between his cooperation and his utility. The ZD strategy proposed in [19] brings us a significant inspiration for realizing such a correlation.

Generally speaking, a ZD strategy can enforce a linear relationship between the payoffs of two players in an iterated game by setting appropriate values for one player's mixed strategy [21]. Particularly, facilitated with the ZD strategy, a player can unilaterally set the expected payoff of its opponent according to some arbitrary property. This motivates us to consider a mechanism in which the requestor may take advantage of ZD to reward the worker's cooperation while penalize his defection so as to achieve a win-win situation and finally get rid of the dilemma. However, the classical ZD strategy is derived for simultaneous games where the two players act at the same time without knowing the action of the opponent in the current round. Therefore, we need to extend the classical ZD and make it fit our sequential game.

As proved in [19], a long-memory player has no priority against a short-memory player in an iterated game. Thus to extend the classical ZD strategy, here we assume that the requestor has only one round of memory, and her mixed strategy at each round is the conditional probabilities she chooses strategy c under all possible states of the previous round. We denote the requestor's mixed strategy at round t as $\mathbf{p}^t = (p_1^t, p_2^t, p_3^t, p_4^t)$, where p_1^t is the probability of choosing c when the outcome of round $t-1$ is $\tilde{x}\tilde{y} = cc$; thus the probability of choosing d is $1 - p_1^t$. Similarly, p_2^t, p_3^t, and p_4^t are the probabilities of adopting c when the previous state is $\tilde{x}\tilde{y} = cd$, dc, and dd, respectively. The worker makes his decision at any round t after seeing the requestor's action in that round; thus his strategy is the conditional probability of choosing c given the requestor's possible action in the current round. We denote the worker's strategy at round t as $\mathbf{q}^t = (q_1^t, q_2^t)$, where q_1^t and q_2^t are the probabilities of choosing c when the requestor's strategies are $\tilde{x} = c$ and d, respectively.

With the above definitions of $\mathbf{p}^t$ and $\mathbf{q}^t$, we can construct a Markov matrix,

$$\mathbf{M} = \begin{bmatrix} p_1^t q_1^t & p_1^t(1-q_1^t) & (1-p_1^t)q_2^t & (1-p_1^t)(1-q_2^t) \\ p_2^t q_1^t & p_2^t(1-q_1^t) & (1-p_2^t)q_2^t & (1-p_2^t)(1-q_2^t) \\ p_3^t q_1^t & p_3^t(1-q_1^t) & (1-p_3^t)q_2^t & (1-p_3^t)(1-q_2^t) \\ p_4^t q_1^t & p_4^t(1-q_1^t) & (1-p_4^t)q_2^t & (1-p_4^t)(1-q_2^t) \end{bmatrix},$$

where each element denotes the state transition probability from round $t-1$ to t. Taking the first row of $\mathbf{M}$ as an example, one can see that the four elements denote the transition probabilities from state $\tilde{x}\tilde{y} = cc$ at round $t-1$ to the four possible states $\tilde{x}\tilde{y} = cc, cd, dc, dd$ at round t. Similarly, the probabilities in the other three rows correspond to the states $\tilde{x}\tilde{y} = cd, dc, dd$ at round $t-1$.

Let $\mathbf{v}$ be the stable vector of the above transition matrix; then $\mathbf{v}^{\mathrm{T}}\mathbf{M} = \mathbf{v}^{\mathrm{T}}$. Let $\mathbf{M}' = \mathbf{M} - \mathbf{I}$, where $\mathbf{I}$ is the unitary matrix; then we have $\mathbf{v}^{\mathrm{T}}\mathbf{M}' = \mathbf{0}$. According to the Cramer's rule, we have $Adj(\mathbf{M}')\mathbf{M}' = det(\mathbf{M}'\mathbf{I}) = 0$, where $Adj(\mathbf{M}')$ is the adjugate matrix of $\mathbf{M}'$. Comparing the above two equations, one can see that $\mathbf{v}$ is proportional to each row of $Adj(\mathbf{M}')$. Therefore, when computing the dot product of the stable vector $\mathbf{v}$ and any vector $\mathbf{f} = (f_1, f_2, f_3, f_4)$, we have

$$\mathbf{v}\cdot\mathbf{f} = D(\mathbf{p}^t,\mathbf{q}^t,\mathbf{f}) = \det\begin{bmatrix} p_1^t q_1^t - 1 & p_1^t - 1 & (1-p_1^t)q_2^t + p_1^t q_1^t - 1 & f_1 \\ p_2^t q_1^t & p_2^t - 1 & (1-p_2^t)q_2^t + p_2^t q_1^t & f_2 \\ p_3^t q_1^t & p_3^t & (1-p_3^t)q_2^t + p_3^t q_1^t - 1 & f_3 \\ p_4^t q_1^t & p_4^t & (1-p_4^t)q_2^t + p_4^t q_1^t & f_4 \end{bmatrix}. \tag{3}$$

Notably, the second column is only related to the strategy of the requestor, denoted as $\widetilde{\mathbf{p}}^t = (p_1^t - 1, p_2^t - 1, p_3^t, p_4^t)^{\mathrm{T}}$. Besides, the expected payoff of the requestor $\tilde{E}_r^t$ and that of the worker $\tilde{E}_w^t$ at round t can be calculated as follows:

$$\tilde{E}_r^t = \frac{\mathbf{v}\cdot\mathbf{S}_r}{\mathbf{v}\cdot\mathbf{1}} = \frac{D(\mathbf{p}^t,\mathbf{q}^t,\mathbf{S}_r)}{D(\mathbf{p}^t,\mathbf{q}^t,\mathbf{1})}, \tag{4}$$

$$\tilde{E}_w^t = \frac{\mathbf{v}\cdot\mathbf{S}_w}{\mathbf{v}\cdot\mathbf{1}} = \frac{D(\mathbf{p}^t,\mathbf{q}^t,\mathbf{S}_w)}{D(\mathbf{p}^t,\mathbf{q}^t,\mathbf{1})}. \tag{5}$$

Hence, when computing a linear combinations of the above two expected payoffs, with α, β, and γ being constant parameters, we have

$$\alpha\tilde{E}_r^t + \beta\tilde{E}_w^t + \gamma = \frac{\mathbf{v}\cdot(\alpha\mathbf{S}_r + \beta\mathbf{S}_w + \gamma\mathbf{1})}{\mathbf{v}\cdot\mathbf{1}} = \frac{D(\mathbf{p}^t,\mathbf{q}^t,\alpha\mathbf{S}_r + \beta\mathbf{S}_w + \gamma\mathbf{1})}{D(\mathbf{p}^t,\mathbf{q}^t,\mathbf{1})}. \tag{6}$$

Therefore, when the requestor sets her strategy $\mathbf{p}^t$ satisfying $\widetilde{\mathbf{p}}^t = (p_1^t - 1, p_2^t - 1, p_3^t, p_4^t)^{\mathrm{T}} = \alpha\mathbf{S}_r + \beta\mathbf{S}_w + \gamma\mathbf{1}$, the second and the fourth columns in (3) are proportional to each other, which means that the right side of the above equation is zero. In this case, what the requestor adopts is called the sequential *zero-determinant* strategy.

When $\alpha = 0$, the requestor's strategy $\mathbf{p}^t$ meets $\widetilde{\mathbf{p}}^t = \beta\mathbf{S}_w + \gamma\mathbf{1}$, and we then have $\tilde{E}_w^t = -\frac{\gamma}{\beta}$. With several steps of variable substitutions, the worker's expected payoff at round t can be written as $\tilde{E}_w^t = \frac{(1-p_1^t)(R_w+b-a)+p_4^t R_w}{1-p_1^t+p_4^t}$, which resides in $[R_w + b - a, R_w]$ since $p_1^t, p_4^t \in [0,1]$.

The above analysis implies that with the adoption of the sequential ZD strategy, the requestor has a unilateral control on the worker's expected payoff, i.e., setting it to a fixed value, which provides the requestor a powerful tool to encourage the worker's cooperation in a sequential crowdsourcing game.

Note that in our sequential ZD strategy, only the first mover (the requestor) can act as the ZD player since only one column (the second one) in (3) is determined by the first strategy maker; while in the classical ZD strategy [19] derived for the simultaneous games, both players can adopt the ZD strategy because the value of the corresponding determinant can be exclusively determined by either of the players. Also note that the proposed extension to the classical ZD strategy for the sequential crowdsourcing game scenario not only solves our problem but also contributes to enlarge the ZD application space.

5 Sequential ZD Strategy based Incentive Algorithm for the Binary Model

With the help of the sequential ZD strategy, the requestor can unilaterally set the expected payoff of the worker in our sequential game. In this section, we utilize the power of the sequential ZD strategy to design an algorithm for the requestor to drive the cooperation of the worker. In fact, there is hardly a universal strategy that can beat all variety of strategies from the opponent. Hence, we need to first consider the worker's strategy before starting the design of our algorithm.

Obviously, if the worker's strategy is fixed, no incentive can change his strategy. Therefore, we consider the worker's strategy to be adaptive that may change with the game result of the requestor or himself. Note that in the sequential game, the worker stays in an information-lacking position compared with the requestor who has the global information about all the workers. Consequently, to maximize his payoff, it seems to be rational for the worker to adopt an evolutionary strategy that is inspired by the idea of "survival of the fittest" in biological evolution. Here we give a loose definition of the evolutionary strategy as follows.

Definition 5.1 (Evolutionary strategy). *The evolutionary strategy adopted by a game player indicates that the player can adjust its strategy to maximize its payoff regardless of the strategy or payoff of its opponent.*

Here we give an example evolutionary strategy [22]. Suppose that q_w^t is the worker's cooperation probability at round t. If he adopts the evolutionary strategy in [22], his cooperation probability in the next round, denoted by q_w^{t+1}, may evolve as follows:

$$q_w^{t+1} = q_w^t \frac{W_c^t}{\tilde{E}_w^t}, \tag{7}$$

where W_c^t is the expected payoff when he cooperates and can be calculated by $W_c^t = p_r^t R_w + (1-p_r^t)(R_w - a)$, with p_r^t being the cooperation probability of the requestor at round t that can be statistically calculated according to the cooperation frequency in practice; and $\tilde{E}_w^t$ is the expected payoff of the worker that can be calculated by $\tilde{E}_w^t = q_w^t W_c^t + (1-q_w^t)W_d^t$, with $W_d^t = p_r^t(R_w + b) + (1-p_r^t)(R_w + b - a)$ being the expected payoff when the worker defects. Thus, it is obvious that the evolutionary worker's cooperation probability can increase only if he perceives that cooperation is profitable, i.e., $W_c^t > \tilde{E}_w^t$; otherwise, his cooperation probability keeps unchanged or decreased, implying that he tends to be more defective. Note that there could be other evolutionary strategies [23], but we demonstrate only this one here and claim that our basic idea is suitable for other variants of the evolutionary strategies.

Next, we propose an algorithm for the requestor confronting a worker who adopts the evolutionary strategy described above to induce his cooperation. The main idea is to reward the cooperation of the worker, i.e., providing the highest job quality, and penalize his defection, i.e., accomplishing job with the lowest quality, with the help of the sequential ZD strategy. The pseudo-code of the algorithm is presented in Algorithm 1, which includes a preparatory stage of N_0 rounds to get the necessary initial values of the state transition probability vector $\mathbf{P}_s = (P_{cc}, P_{cd}, P_{dc}, P_{dd})$, where $P_{ij},\ i,j \in \{c,d\}$, is the probability of the worker changing from state i to j. More specifically, P_{ij} is set to be the ratio of the number of rounds the worker's action changing from state i to j to the total number of rounds. Note that $\mathbf{P}_s$ can also be set to (0.5, 0.5, 0.5, 0.5) or any other reasonable set of initial values and then be updated gradually to approximate the real values as much as possible. Obviously, this process is similar to the preparatory stage mentioned above.

Algorithm 1. Sequential ZD based Incentive Algorithm for the Binary Model

Require: $\mathbf{p}^i = \{p_1^i, p_2^i, p_3^i, p_4^i\}$: the requestor's strategy at round i and its initial values are the ones used in the N_0^{th} round; $\mathbf{P}_s = (P_{cc}, P_{cd}, P_{dc}, P_{dd})$: the state transition probabilities of the worker, and their initial values are calculated statistically through the preparatory N_0 rounds; N: the total number of rounds.

1: Initialize($\tilde{E}_w^0$)
2: **for** $i = 1$ to N **do**
3:   **if** The worker's last move is c **then**
4:     **if** $P_{cc} > P_{cd}$ **then**
5:       Set $\{p_1^i, p_2^i, p_3^i, p_4^i | \frac{(1-p_1^i)(R_w+b-a)+p_4^i R_w}{1-p_1^i+p_4^i} = R_w \wedge 0 \le p_1^i, p_2^i, p_3^i, p_4^i \le 1\}$
6:       $\tilde{E}_w^i \leftarrow R_w$
7:     **else** ▷ $P_{cc} \le P_{cd}$.
8:       Set $\{p_1^i, p_2^i, p_3^i, p_4^i | \frac{(1-p_1^i)(R_w+b-a)+p_4^i R_w}{1-p_1^i+p_4^i} = R_w + b - a \wedge 0 \le p_1^i, p_2^i, p_3^i, p_4^i \le 1\}$
9:       $\tilde{E}_w^i \leftarrow R_w + b - a$
10:     **end if**
11:   **else** ▷ The worker's last move is d.
12:     **if** $P_{dc} > P_{dd}$ **then**
13:       Set $\{p_1^i, p_2^i, p_3^i, p_4^i | \frac{(1-p_1^i)(R_w+b-a)+p_4^i R_w}{1-p_1^i+p_4^i} = R_w \wedge 0 \le p_1^i, p_2^i, p_3^i, p_4^i \le 1\}$
14:       $\tilde{E}_w^i \leftarrow R_w$
15:     **else** ▷ $P_{dc} \le P_{dd}$.
16:       Set $\{p_1^i, p_2^i, p_3^i, p_4^i | \frac{(1-p_1^i)(R_w+b-a)+p_4^i R_w}{1-p_1^i+p_4^i} = R_w + b - a \wedge 0 \le p_1^i, p_2^i, p_3^i, p_4^i \le 1\}$
17:       $\tilde{E}_w^i \leftarrow R_w + b - a$
18:     **end if**
19:   **end if**
20:   **if** The current round ends **then**
21:     Update $\mathbf{P_s}$
22:   **end if**
23: **end for**

After initializing $\tilde{E}_w^0$ (Step 1) during the preparatory stage, the requestor adjusts the worker's expected payoff $\tilde{E}_w^i$ according to the prediction on the worker's action at each round i. More specifically, when the worker's last move is cooperation (Step 3), the prediction of his current move is determined by the relationship of P_{cc} and P_{cd}. If $P_{cc} > P_{cd}$, the requestor regards that the worker is friendly and sets the highest expected payoff $\tilde{E}_w^i = R_w$ to him (Steps 4-6) ; if not, the lowest $\tilde{E}_w^i = R_w + b - a$ is given to the worker (Steps 7-10). Similarly, when the worker's last move is defection, it comes to the comparison between P_{dc} and P_{dd}. If the former is greater, the requestor gives the worker the highest payoff $\tilde{E}_w$ (Steps 12-14), and vice versa (Steps 15-18). At the end of each round, the values of the state transition probabilities are updated (Steps 20-22). When the total number of rounds is reached, the whole process terminates (Step 23).

According to the above description, one can see that once the requestor predicts that the worker would cooperate in this round, she should give the worker a reward of R_w, equaling to the payoff obtained by the worker when both cooperate, which implies that the requestor does not sacrifice any extra cost for incentivizing the worker. It is also worthy of noting that to stimulate the high-quality submissions from multiple workers, the requestor can implement the above algorithm with different initial state transition probabilities for different workers, calculating and storing different values for different workers separately.

Note that other binary strategy situations with $\Pi \ge 3$ are very similar to the above situation with $\Pi = 2$, except for the increased cardinality of the strategy and payoff sets. In this case, the derivation of sequential ZD strategy and incentive algorithm design can be conducted following the same basic idea. Thus, due to the page length limitation, we omit the detailed description.

6 Sequential ZD Strategy based Incentive Algorithm for the Continuous Model

In order to take advantage of the sequential ZD to encourage the cooperation of the worker adopting a continuous strategy, we make use of an idea similar to that proposed in [24] to construct the continuous-sequential ZD strategy. To that aim, we first calculate the expected utilities of the requestor and the worker at round t with their utility functions $w_r(x, y)$ and $w_w(x, y)$ are respectively defined as follows,

$$E_r^t = \int_{l_w}^{h_w} \int_{l_r}^{h_r} v^t(x, y) w_r(x, y) dx dy, \tag{8}$$

$$E_w^t = \int_{l_w}^{h_w} \int_{l_r}^{h_r} v^t(x, y) w_w(x, y) dx dy, \tag{9}$$

where $v^t(x, y)$ is the joint probability function of the requestor adopting strategy x and the worker adopting y at round t. To calculate $v^t(x, y)$, we define the mixed strategy of the requestor $p^t(x|x_{-1}, y_{-1})$ as the conditional probability at which the requestor chooses to offer a payment of x at round t with the outcome $x_{-1}y_{-1}$ at round $t-1$, where $x_{-1}, x \in [l_r, h_r]$ and $y_{-1} \in [l_w, h_w]$. With the above definitions, we have

$$\int_{l_r}^{h_r} p^t(x_{-1}, y_{-1}, x) dx = 1. \tag{10}$$

For the worker, the trait of a sequential game makes him choose a strategy according to the requestor's current strategy; thus we define the mixed strategy of the worker $q^t(y|x)$ as the conditional probability at which the worker provides the job quality y when the requestor offers him the payment x at the current round t, where $x \in [l_r, h_r], y \in [l_w, h_w]$. Also, we have

$$\int_{l_w}^{h_w} q^t(y|x) dy = 1. \tag{11}$$

The transition function $M(x_{-1}, y_{-1}, x, y)$ denoting the state transition probability from round $t-1$ to round t can be expressed as follows:

$$M(x_{-1}, y_{-1}, x, y) = p^t(x|x_{-1}, y_{-1}) q^t(y|x). \tag{12}$$

It shapes the relationship between the state probability at round $t-1$ and that at round t as shown below:

$$v^{t-1}(x_{-1}, y_{-1}) \cdot M(x_{-1}, y_{-1}, x, y) = v^t(x, y). \tag{13}$$

Similar to the sequential ZD strategy derived under the binary model, we can deduce its counterpart in the continuous model, which is summarized by Lemma 6.1.

Lemma 6.1. *When the mixed strategy of the requestor $p^t(x|x_{-1}, y_{-1})$ satisfies $\widetilde{p}^t(h_r|x_{-1}, y_{-1}) = \alpha w_r(x, y) + \beta w_w(x, y) + \gamma$, the expected utilities of the requestor and the worker have the following linear relationship,*

$$\alpha E_r^t + \beta E_w^t + \gamma = 0, \tag{14}$$

where $\widetilde{p}^t(h_r|x_{-1}, y_{-1})$ is defined as follows:

$$\widetilde{p}^t(h_r|x_{-1}, y_{-1}) = \begin{cases} p^t(h_r|x_{-1}, y_{-1}), \ x < h_r, \\ p^t(h_r|x_{-1}, y_{-1}) - 1, \ x = h_r. \end{cases}$$

The detailed proof of the above lemma is elaborated in Appendix A.

According to Lemma 6.1, when $\alpha = 0$, we have $E_w^t = -\frac{\gamma}{\beta}$; thus the strategy of the requestor $p^t(x|x_{-1}, y_{-1})$ can be calculated, and one can consequently figure out the maximum and minimum of the worker's expected utility, denoted by $\max(E_w^t)$ and $\min(E_w^t)$, respectively. To design an algorithm for the requestor to drive the worker to provide high job quality under the continuous model, we need to choose a metric to quantify the willingness of the worker's good behavior. Motivated by the approach used in Section 5, one can focus on the probability of the worker providing the highest job quality. However, since the strategies the requestor and the worker can adopt are continuous, the probability of a point in a continuous interval is zero; thus we employ the probability density of providing the highest job quality to measure the willingness of the worker behaving friendly. Therefore we define f_h^t as the probability density of the worker choosing h_w as the job quality at round t. As mentioned above, a rational worker would take an evolutionary strategy to obtain a high utility in an iterated sequential game; thus, f_h^t evolves as follows,

$$f_h^{t+1} = f_h^t \frac{W_h^t}{E_w^t}, \tag{15}$$

where W_h^t is the expected utility of the worker when he adopts $y = h_w$, which can be calculated by $W_h^t = \int_{l_r}^{h_r} g_x^t w_w(x, h_w) dx$, and E_w^t is the expected utility of the worker, which can be calculated by $E_w^t = \int_{l_w}^{h_w} f_y^t W_y^t dy$, with g_x^t being the probability density of the requestor offering the payment x in its value range $[l_r, h_r]$ at round t, f_y^t being the probability density of the worker providing the quality y in its value range $[l_w, h_w]$ at round t, and W_y^t being the expected utility of the worker when he provides the job quality y. Note that W_y^t can be calculated by $W_y^t = \int_{l_r}^{h_r} g_x^t w_w(x, y) dx$. Thus, one can conclude that only if the worker's expected utility of adopting $y = h_w$ is the maximum compared to those obtained when using other strategies, would the worker choose or maintain the job quality $y = h_w$; otherwise, the worker's job quality would come to a lower level.

In light of the above evolution characteristic of the worker, we design a sequential ZD based incentive algorithm (summarized in Algorithm 2) for the requestor to stimulate the worker's provision of the highest job quality in a continuous crowdsourcing game. The core idea of this algorithm is similar to that of the binary one, i.e., rewarding the high job quality provided by the worker and penalizing his low job quality. Note that there is also a preparatory stage including N_0 rounds to initialize the parameters. Also note that in order to better control the evolution path of the worker's probability density f_h^t, the requestor needs to maintain a state transition probability matrix $\mathbf{P}_s^c = \{P_{ij}^c\}_{\eta\times\eta}$ to predict the worker's action, where P_{ij}^c is the statistical probability of the worker's state changing from interval $[l_w + (i-1)\delta, l_w + i\delta]$ to $[l_w + (j-1)\delta, l_w + j\delta]$, with δ being a sufficiently small number specifically defined in Appendix A and satisfying $l_w + \eta\delta = h_w$.

Algorithm 2. Sequential ZD Strategy based Incentive Algorithm for the Continuous Model

Require: $p^t(x|x_{-1}, y_{-1})$: the requestor's strategy and its initial value is the one used in the N_0^{th} round; $\mathbf{P}_s^c = \{P_{ij}^c\}_{\eta\times\eta}$: the state transition probability of the worker, and its initial value is calculated statistically through the preparatory N_0 rounds; N: the total number of rounds for algorithm termination.

```
1:  Initialize(E_w^0)
2:  for i = 1 to N do
3:    if y^{i-1} ∈ [l_w + (κ-1)δ, l_w + κδ] then
4:      if P^c_{κ-η} ≥ P^c_{κ-j}, ∀j ∈ {1, w, ..., η} then
5:        Set p^i(x|x_{-1}, y_{-1}) to let
          E_w^i ← max(E_w^t)
6:      else                                   ▷ P^c_{κ-η} < P^c_{κ-j}.
7:        Set p^i(x|x_{-1}, y_{-1}) to let
          E_w^i ← min(E_w^t)
8:      end if
9:    end if
10:   if The current round ends then
11:     Update P_s^c
12:   end if
13: end for
```

The initialization of E_w^0 is completed through the preparatory N_0 rounds (Step 1). After that, the requestor adjusts her strategy to make the worker's expected utility change with his action at each round i. Specifically, according to the state transition probability matrix $\mathbf{P}_s^c$, the requestor can predict the worker's current action. When the worker's last move is located at interval $[l_w + (\kappa-1)\delta, l_w + \kappa\delta]$ (Step 3), the requestor determines the worker's tendency of providing the highest job quality by comparing the transition probability from the state interval $[l_w + (\kappa-1)\delta, l_w + \kappa\delta]$ to the interval $[l_w + (\eta-1)\delta, h_w]$. If $P_{\kappa-\eta}^c$ is the largest, the requestor regards that the worker is likely to provide the highest quality in this round and thus she would set the highest expected utility for the worker (Steps 4-6); if not, the requestor would set the lowest E_w^i (Steps 7-10). At the end of each round, the requestor updates the state transition probability matrix $\mathbf{P}_s^c$ for a more precise estimation of the worker's next action (Steps 12-14). The algorithm terminates when certain number of rounds are performed (Step 15). Similar to Algorithm 1, at each

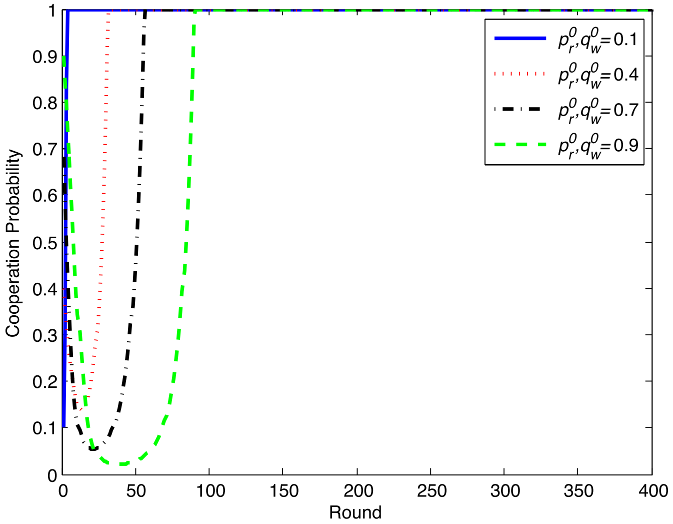

Fig. 2. Cooperation probability evolution of the worker in the binary model.

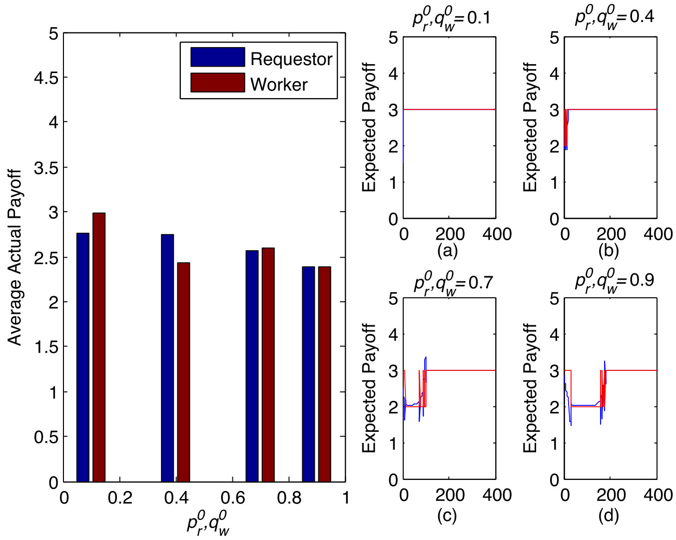

Fig. 3. Payoffs of the requestor and the worker in the binary model.

round, the reward that the requestor gives to the worker if he cooperates is equivalent to that obtained by the worker when both players cooperate, which means that the requestor does not give extra payment for incentivizing the worker to be cooperative.

7 Simulation Results

In this section, we present our simulation study for the proposed incentive algorithms under the two different game models. First, we simulate the algorithm in a binary sequential crowdsourcing game by setting $R_w = 3, R_r = 3, a = 3, b = 2, m = 3, n = 2$, which satisfies the parameter value relationship claimed in Section 3; thus we have $\mathbf{S}_r = (3, 0, 5, 2)$ and $\mathbf{S}_w = (3, 5, 0, 2)$. Note that we also simulate other parameter settings satisfying the aforementioned parameter constraint and obtain very similar results, which are omitted here due to redundancy. Besides, we set $N_0 = 100$, the number of rounds in the preparatory stage, and $N = 400$, the number of rounds for the algorithm execution. Here we assume that both the requestor and the worker adopt the evolutionary strategy in the preparatory stage; thus the initial cooperation probabilities at the beginning of Algorithm 1 are the same, i.e., $p_r^0 = q_w^0$. Each simulation has been repeated 30 times to obtain the average value with sufficient statistical confidence.

Fig. 2 presents the evolution of the worker's cooperation probability under the binary model, with different initial cooperation probabilities p_r^0 and q_w^0. One can see that the worker's cooperation probability q_w^t always reaches 1 in the end no matter what value of q_w^0 he selects at the beginning, which implies that the worker could finally become cooperative whether or not he cooperates at the beginning of the game. This result obviously justifies the effectiveness of our proposed ZD based algorithm. Note that there exists obvious difference between the case of $p_r^0 = q_w^0 = 0.1$ and the other three cases. The concave parts under the cases of the three larger initial cooperation probability values are resulted from the higher initial expected payoff, which leads to a smaller rewarding space and a larger penalty space when q_w^t drops down for a while.

Fig. 3 plots the actual payoffs and the expected payoffs of the requestor and the worker with different initial cooperation probabilities p_r^0 and q_w^0 in the binary model. In the left side of Fig. 3, the bar graph shows that both players get relatively equivalent average actual payoffs under different situations, which indicates that our proposed algorithm is fair to both players. In other words, even though it is the requestor who dominates the sequential crowdsourcing game, she cannot get a higher than the normal payoff value by reducing the worker's payment. In addition, as the initial cooperation probabilities increase, the payoffs decrease slightly. The reason lies in that a larger q_w^0 causes a trough on the evolution of q_w, during which the payoffs are lower. Figures (a) (b) (c) (d) in the right side report the expected payoffs of the requestor and the worker corresponding to different values of p_r^0 and q_w^0, which are coincident with the average actual payoffs in the left side.

We also compare our proposed ZD based algorithm for the binary model (labeled by ZD) with five other classical strategies, i.e., win-stay-lose-shift (WSLS) [25], tit-for-tat (TFT) [26], all-cooperation (ALLC), all-defection (ALLD), and random (Random), by simulating the cases where the requestor adopts different strategies while the worker always adopts the evolutionary strategy. We mainly focus on the worker's cooperation probability evolution and the average actual payoffs of the requestor and the worker. Fig. 4 demonstrates the change of the cooperation probability of the worker when more rounds are performed. Note that only the first 50 rounds are displayed in Fig. 4 for a clear observation on the differences among these strategies, though in total we perform 400 rounds. It is obvious that only when the requestor adopts the ZD strategy based method can she drive the worker to be finally cooperative, and the other five classical strategies are not able to stimulate the worker's cooperation.

The actual payoffs of both the requestor and the worker are presented in Fig. 5. One can see that the ZD based algorithm leads to a relatively commensurate payoffs of around 3 for both players, which corresponds to the payoffs of mutual cooperation; while WSLS, ALLC, and Random make the requestor stay in a disadvantage position, getting far less payoffs than the evolutionary worker, which means

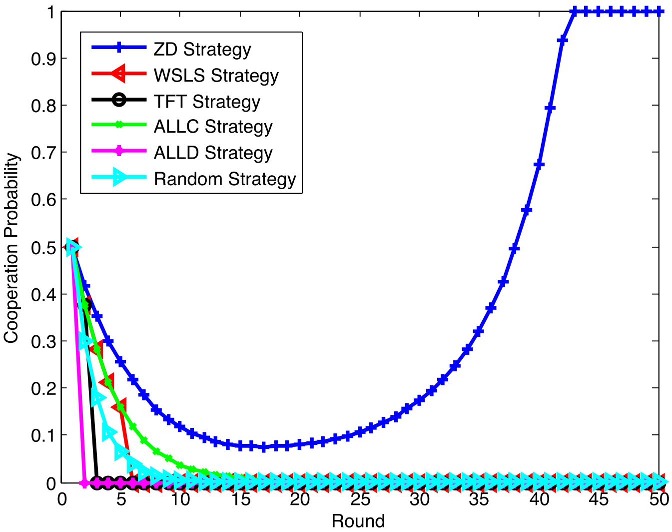

Fig. 4. Cooperation probabilities of the worker when the requestor adopts different strategies in the binary model.

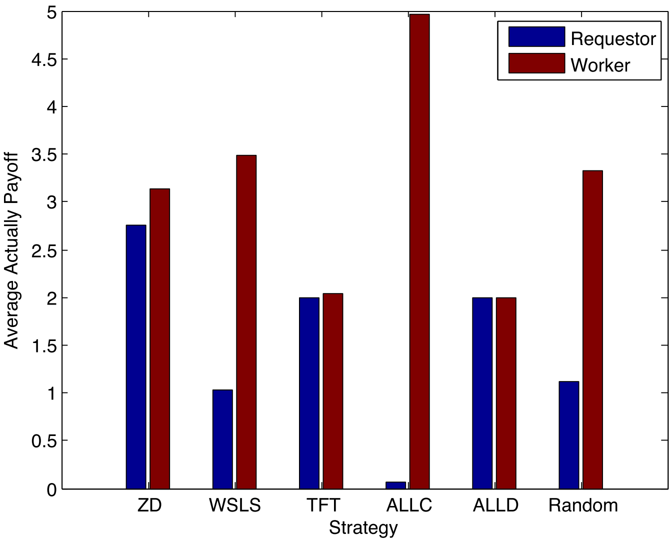

Fig. 5. Payoffs of the requestor and the worker when the requestor takes different strategies in the binary model.

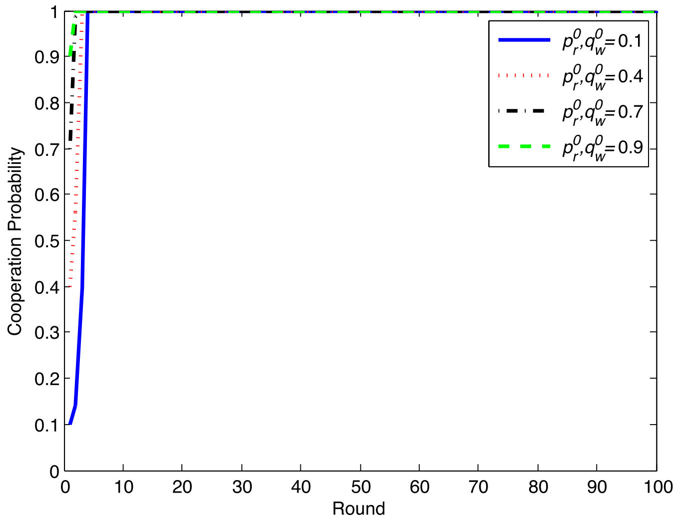

Fig. 6. Cooperation probability evolution of the worker in the continuous model.

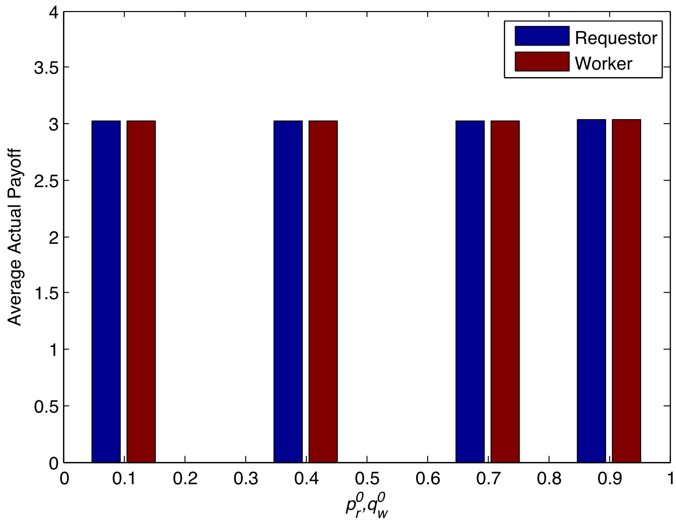

Fig. 7. Payoffs of the requestor and the worker in the continuous model.

that there are too many rounds at state cd resulted from the requestor lacking the capability to incentivize the worker's cooperation; when the requestor adopts either TFT or ALLD, she has an equivalent payoff of 2 with the worker, which means that their state is mostly dd during the whole simulation process, failing to encourage the final cooperation of the worker.

To simulate Algorithm 2, we assume that both the requestor and the worker can choose their strategies from the interval [0, 10], i.e., $l_r = l_w = 0, h_r = h_w = 10$. We further assume that $\phi(y) = \frac{1}{1+exp(-y)}$, $A_r = 4$, and $B_r = 0.1$, and that $\psi(y) = \frac{1}{1+exp(-y)} - 1$, $A_w = 0.3$, and $B_w = 4$; then we have $w_r(0,0) = w_w(0,0) = 2$ and $w_r(10,10) = w_w(10,10) = 3$. Note that multiple sets of parameters and monotonically increasing functions have been tested but we only present the simulation results of the above ones to avoid redundancy as the results are very similar. Besides, to have a better statistical calculation of the state probability transition matrix $\mathbf{P}_s^c$, we divide the continuous strategy space into 10 sub-spaces. We also set the preparatory stage to $N_0 = 100$ rounds and the whole simulation process takes $N = 400$ rounds; but we only display the outcomes of the first 100 rounds for a clear observation on the experimental results. Each simulation has been repeated 30 times to obtain the average value for sufficient statistical confidence.

The worker's cooperation probability and the actual payoffs of both players in the continuous model are presented in Figs. 6 and 7. In Fig. 6, one can clearly observe that our ZD strategy based algorithm can facilitate the requestor to force the cooperation probability of the worker to increase to 1, no matter what the initial cooperation probability of the worker is. The average actual payoffs in Fig. 7 demonstrate that both players have payoffs of 3, which means that they adopt $(x, y) = (10, 10)$ most of the time during the simulation process. This is an indication of fairness of our proposed algorithm, where the requestor has no way to obtain an income higher than the normal payoff received from the mutual cooperation state.

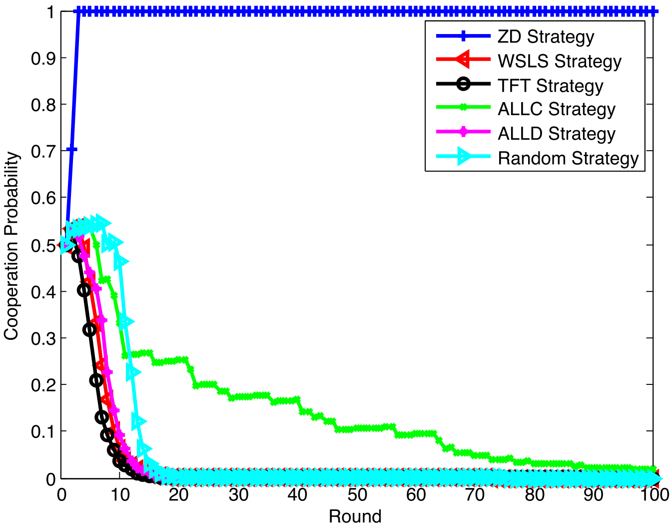

Fig. 8. Cooperation probability of the worker with different strategies of the requestor in the continuous model.

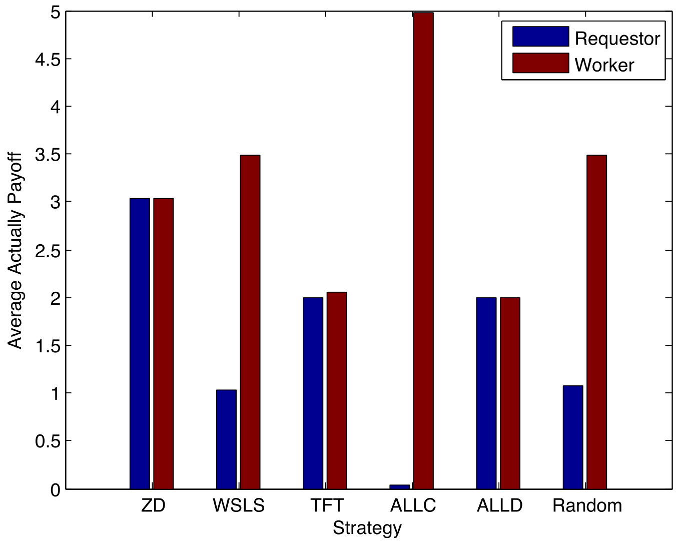

Fig. 9. Payoffs of the requestor and the worker with different strategies of the requestor in the continuous model.

We also compare our sequential continuous ZD strategy with the other five classical ones mentioned above. The cooperation probability's evolution of the worker and the payoffs of both players are presented in Figs. 8 and 9, respectively. The worker's cooperation probability demonstrates totally different evolution paths when the requestor takes different strategies. One can see that only the proposed sequential continuous ZD strategy can drive the worker to be finally cooperative, while the other five classical strategies fail to realize the ultimate goal of the requestor. In addition, the average payoffs are similar to those obtained in the binary model, where only the ZD strategy results in an equivalent payoff of 3, which corresponds to $w_r(10,10) = w_w(10,10)$; while the other strategies trap the requestor into an unfavorable position.

8 Conclusion

In this paper, we propose an economics-based quality control mechanism for crowdsourcing in which monetary reward and penalty are employed to encourage or force the workers to behave nicely. To that aim, we revise the ZD strategy so that its variant can be employed in our sequential games. By means of the sequential ZD, we design two incentive algorithms to improve the job quality of the worker when both players perform binary or continuous strategies. In both algorithms, the requestor can unilaterally set the expected payoff of the worker according to the predicted action of the worker. Thus, our algorithms can incentivize the cooperation of the workers without a long-term extra payment. Note that our methods do not involve specific crowdsourcing scenarios; they focus on training rather than simply filtering out unprofessional workers, and hence can call on all possible human resources, which is a key trait of crowdsourcing resulting in its success. The extensive simulation results demonstrate that our proposed algorithms can efficiently and effectively encourage the workers to become cooperative.

Appendix A
Proof of Lemma 6.1

Proof. For the continuous strategy space, we divide it into η parts. Thus, the requestor's strategy x can be one of $\{l_r, l_r+\delta, l_r+2\delta, \ldots, l_r+\eta\delta\}$, and the worker's strategy y can be one of $\{l_w, l_w+\delta, l_w+2\delta, \ldots, l_w+\eta\delta\}$, where δ is a sufficiently small number while η is a sufficiently large number, which satisfy $l_r+\eta\delta = h_r, l_w+\eta\delta = h_w$. Thus, when $\delta \to 0$, their strategies become continuous. Accordingly, the payoff of the requestor becomes $\mathbf{W}_r = \{w_r(l_r, l_w), \ldots, w_r(l_r, l_w+\eta\delta), \ldots, w_r(l_r+\eta\delta, l_w), \ldots, w_r(l_r+\eta\delta, l_w+\eta\delta)\}$, which can be denoted as $\mathbf{W}_r = \{w_{r00}, \ldots, w_{r0\eta}, \ldots, w_{r\eta 0}, \ldots, w_{r\eta\eta}\}$, and the payoff of the worker should be $\mathbf{W}_w = \{w_w(l_r, l_w), \ldots, w_w(l_r, l_w+\eta\delta), \ldots, w_w(l_r+\eta\delta, l_w), \ldots, w_w(l_r+\eta\delta, l_w+\eta\delta)\}$, denoted as $\mathbf{W}_w = \{w_{w00}, \ldots, w_{w0\eta}, \ldots, w_{w\eta 0}, \ldots, w_{w\eta\eta}\}$. Then we can denote the mixed strategy of the requestor at round t as p^t_{ij-k}, where $i, j, k \in \{0, 1, \ldots, \eta\}$, which is the probability of the requestor offering a payment of $l_r + k\delta$ at round t when she offers the payment of $l_r + i\delta$ and the worker provides the job quality of $l_w + j\delta$ at round $t-1$; similarly, we denote the mixed strategy of the worker at round t as q^t_{i-j}, where $i, j \in \{0, 1, \ldots, \eta\}$, which is the probability of the worker providing the job quality $l_w + j\delta$ at round t when the requestor offers the payment of $l_r + i\delta$ in the same round.

According to the above partition on the strategy spaces and the corresponding utility spaces, we can construct the following Markov state transition matrix,

$$\mathbf{M}_d = [\mathbf{M}_{00}, \ldots, \mathbf{M}_{0\eta}, \mathbf{M}_{11}, \ldots, \mathbf{M}_{1\eta}, \ldots, \mathbf{M}_{\eta 0}, \ldots, \mathbf{M}_{\eta\eta}], \quad (16)$$

where $\mathbf{M}_{ij}, \forall i, j \in \{0, 1, \ldots, \eta\}$ is a vector containing the transition probability from all the possible last state of xy to the current state $x = l_r + i\delta$ and $y = l_w + j\delta$, and it can be expressed as follows,

$$\begin{aligned}\mathbf{M}_{ij} =& [p^t_{00-i}q^t_{i-j}, \ldots, p^t_{0\eta-i}q^t_{i-j}, p^t_{11-i}q^t_{i-j}, \ldots, \\ & p^t_{1\eta-i}q^t_{i-j}, \ldots, p^t_{\eta 0-i}q^t_{i-j}, \ldots, p^t_{\eta\eta-i}q^t_{i-j}]^{\mathrm{T}}.\end{aligned} \quad (17)$$

If the stable vector of the above transition matrix is $\mathbf{v}_d$, we have $\mathbf{v}_d^{\mathrm{T}}\mathbf{M}_d = \mathbf{v}_d^{\mathrm{T}}$. Suppose $\mathbf{M}'_d = \mathbf{M}_d - \mathbf{I}$, we get $\mathbf{v}_d^{\mathrm{T}}\mathbf{M}'_d = 0$. With the similar calculation method for the binary model, we obtain that $\mathbf{v}_d$ is proportional to each row of $Adj(\mathbf{M}'_d)$. Therefore, for any vector $\mathbf{f} = [f_{00}, f_{01}, \ldots, f_{\eta\eta}]^{\mathrm{T}}$, with the known condition $\sum_{j=0}^{\eta} q_{i-j}^t = 1$, we can compute the dot product of $\mathbf{f}$ and $\mathbf{v}_d$ as follows,

$$\mathbf{v}_d \cdot \mathbf{f} = D(\mathbf{p}^t, \mathbf{q}^t, \mathbf{f})$$
$$= det \begin{bmatrix} p_{00-0}^t q_{0-0}^t & \cdots & p_{00-\eta}^t & f_{00} \\ \vdots & \vdots & \vdots & \vdots \\ p_{(\eta-1)\eta-0}^t q_{0-0}^t & \cdots & p_{(\eta-1)\eta-\eta}^t & f_{(\eta-1)\eta} \\ p_{\eta\eta-0}^t q_{0-0}^t & \cdots & p_{\eta\eta-\eta}^t - 1 & f_{\eta 0} \\ \vdots & \vdots & \vdots & \vdots \\ p_{\eta\eta-\eta}^t q_{0-0}^t & \cdots & p_{\eta\eta-\eta}^t - 1 & f_{\eta\eta} \end{bmatrix}. \tag{18}$$

It is obvious that the penultimate column of the above determinant is only determined by the requestor, which can be denoted as $\widetilde{\mathbf{p}}_{(\eta+1)(\eta+1)}^t$.

When $\mathbf{f} = \alpha\mathbf{W}_r + \beta\mathbf{W}_w + \gamma\mathbf{1}$, we have $\mathbf{v}_d \cdot \mathbf{f} = \mathbf{v}_d(\alpha\mathbf{W}_r + \beta\mathbf{W}_w + \gamma) = \alpha E_r^t + \beta E_w^t + \gamma$. Therefore, if the $((\eta+1) \times (\eta+1))^{th}$ column $\widetilde{\mathbf{p}}_{(\eta+1)(\eta+1)}^t = \alpha\mathbf{W}_r + \beta\mathbf{W}_w + \gamma\mathbf{1}$, we have $\alpha E_r^t + \beta E_w^t + \gamma = 0$. When the small number δ approaches 0, we obtain the result of the lemma. □

Acknowledgments

This work is partially supported by the US NSF under grants IIS-1741279 and CNS-1704397, and the National Natural Science Foundation of China under grants U1811463, 61832012, 61772080, 61771289, 61571049, and 61472044.

References

[1] G. Li, J. Wang, Y. Zheng, and M. J. Franklin, "Crowdsourced data management: A survey," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 9, pp. 2296–2319, Sep. 2016.
[2] G. Li, Y. Zheng, J. Fan, J. Wang, and R. Cheng, "Crowdsourced data management: Overview and challenges," in *Proc. ACM Int. Conf. Manag. Data*, 2017, pp. 1711–1716.
[3] Y. Zheng, G. Li, Y. Li, C. Shan, and R. Cheng, "Truth inference in crowdsourcing: Is the problem solved?" *Proc. VLDB Endowment*, vol. 10, no. 5, pp. 541–552, 2017.
[4] Q. Hu, S. Wang, L. Ma, R. Bie, and X. Cheng, "Anti-malicious crowdsourcing using the zero-determinant strategy," in *Proc. IEEE 37th Int. Conf. Distrib. Comput. Syst.*, 2017, pp. 1137–1146.
[5] G. Wang, T. Wang, H. Zheng, and B. Y. Zhao, "Man versus machine: Practical adversarial detection of malicious crowdsourcing workers," in *Proc. Usenix Security Symp.*, 2014, pp. 239–254.
[6] Y. Xuan, Y. Chen, H. Li, P. Hui, and L. Shi, "Lbsnshield: Malicious account detection in location-based social networks," in *Proc. 19th ACM Conf. Comput. Supported Cooperative Work Social Comput. Companion*, 2016, pp. 437–440.
[7] O. Folorunso and O. A. Mustapha, "A fuzzy expert system to trust-based access control in crowdsourcing environments," *Appl. Comput. Informat.*, vol. 11, no. 2, pp. 116–129, 2015.
[8] B. Ye, Y. Wang, and L. Liu, "Crowd trust: A context-aware trust model for worker selection in crowdsourcing environments," in *Proc. IEEE Int. Conf. Web Serv.*, 2015, pp. 121–128.
[9] H. Wang, S. Guo, J. Cao, and M. Guo, "MeLoDy: A long-term dynamic quality-aware incentive mechanism for crowdsourcing," *IEEE Trans. Parallel Distrib. Syst.*, vol. 29, no. 4, pp. 901–914, Apr. 2018.
[10] Y. Zheng, G. Li, and R. Cheng, "Docs: a domain-aware crowdsourcing system using knowledge bases," *Proc. VLDB Endowment*, vol. 10, no. 4, pp. 361–372, 2016.
[11] C. C. Cao, J. She, Y. Tong, and L. Chen, "Whom to ask?: jury selection for decision making tasks on micro-blog services," *Proc. VLDB Endowment*, vol. 5, no. 11, pp. 1495–1506, 2012.
[12] A. Das Sarma, A. Parameswaran, and J. Widom, "Towards globally optimal crowdsourcing quality management: The uniform worker setting," in *Proc. Int. Conf. Manag. Data*, 2016, pp. 47–62.
[13] J. Fan, G. Li, B. C. Ooi, K.-l. Tan, and J. Feng, "iCrowd: An adaptive crowdsourcing framework," in *Proc. ACM SIGMOD Int. Conf. Manag. Data*, 2015, pp. 1015–1030.
[14] S. Zhu, S. Kane, J. Feng, and A. Sears, "A crowdsourcing quality control model for tasks distributed in parallel," in *Proc. Extended Abstracts Human Factors Comput. Syst.*, 2012, pp. 2501–2506.
[15] J. B. Vuurens and A. P. de Vries, "Obtaining high-quality relevance judgments using crowdsourcing," *IEEE Internet Comput.*, vol. 16, no. 5, pp. 20–27, Sep./Oct. 2012.
[16] Y. Zheng, J. Wang, G. Li, R. Cheng, and J. Feng, "QASCA: A quality-aware task assignment system for crowdsourcing applications," in *Proc. ACM SIGMOD Int. Conf. Manag. Data*, 2015, pp. 1031–1046.
[17] D. Oleson, A. Sorokin, G. P. Laughlin, V. Hester, J. Le, and L. Biewald, "Programmatic gold: Targeted and scalable quality assurance in crowdsourcing," *Human Comput.*, vol. 11, no. 11, pp. 43–48, 2011.
[18] C. Qiu, A. C. Squicciarini, S. M. Rajtmajer, and J. Caverlee, "Dynamic contract design for heterogenous workers in crowdsourcing for quality control," in *Proc. IEEE 37th Int. Conf. Distrib. Comput. Syst.*, 2017, pp. 1168–1177.
[19] W. H. Press and F. J. Dyson, "Iterated prisoner's dilemma contains strategies that dominate any evolutionary opponent," *Proc. Nat. Academy Sci. United States America*, vol. 109, no. 26, pp. 10 409–10 413, 2012.
[20] R. J. Aumann, "Backward induction and common knowledge of rationality," *Games Econ. Behavior*, vol. 8, no. 1, pp. 6–19, 1995.
[21] H. Dong, R. Zhi-Hai, and Z. Tao, "Zero-determinant strategy: An underway revolution in game theory," *Chin. Physics B*, vol. 23, no. 7, 2014, Art. no. 078905.
[22] J. M. Smith, *Evolution and the Theory of Games*. Cambridge, U.K.: Cambridge Univ. Press, 1982.
[23] R. M. Dawes, A. J. Van de Kragt, and J. M. Orbell, "Not me or thee but we: The importance of group identity in eliciting cooperation in dilemma situations: Experimental manipulations," *Acta Psychologica*, vol. 68, no. 1–3, pp. 83–97, 1988.
[24] H. Zhang, D. Niyato, L. Song, T. Jiang, and Z. Han, "Zero-determinant strategy for resource sharing in wireless cooperations," *IEEE Trans. Wireless Commun.*, vol. 15, no. 3, pp. 2179–2192, Mar. 2016.
[25] M. Nowak and K. Sigmund, "A strategy of win-stay, lose-shift that outperforms tit-for-tat in the prisoner's dilemma game," *Nature*, vol. 364, no. 6432, 1993, Art. no. 56.
[26] L. M. Andersson and C. M. Pearson, "Tit for tat? the spiraling effect of incivility in the workplace," *Academy Manage. Rev.*, vol. 24, no. 3, pp. 452–471, 1999.

Qin Hu received the BS degree in electronic engneering and the MS degree in computer science from Beijing Normal University, in 2014 and 2017, respectively. She is currently working toward the PhD degree in computer science at The George Washington University. Her research interests include Wireless network, crowdsourcing/crowdsensing, and game theory.

Shengling Wang received the PhD degree from Xi'an Jiaotong University, in 2008. She is an associate professor with the College of Information Science and Technology, Beijing Normal University. After that, she did her postdoctoral research with the Department of Computer Science and Technology, Tsinghua University. Then, she worked as an assistant and associate professor from 2010 to 2013 in the Institute of Computing Technology of the Chinese Academy of Sciences. Her research interests include mobile/wireless networks, game theory, and crowdsourcing. She is a member of the IEEE.

Peizi Ma received the BS degree in computer science and technology from Beijing Normal University, in 2016. She is currently working toward the MS degree in computer science at Beijing Normal University. Her research interests include wireless network, network science, and game theory.

Xiuzhen Cheng received the MS and PhD degrees in computer science from the University of Minnesota Twin Cities, in 2000 and 2002, respectively. She is a professor with the Department of Computer Science, The George Washington University, Washington, DC. Her current research interests focus on privacy-aware computing, wireless and mobile security, smart cities, and algorithm design and analysis. She has served on the editorial boards of several technical publications and the technical program committees of various professional conferences/workshops. She is a fellow of the IEEE.

Weifeng Lv received the PhD degree in computer science from Beihang University. His research interests include massive information system, urban cognitive computing, swarm intelligence, and smart cities. He is a professor of computer science, the dean of the School of Computer Science and Engineering, and the vice director of the State Key Laboratory of Software Development Environment, at Beihang University. He also serves as the secretary-general of the China Software Industry Association and the director of the National Engineering Research Center for Science and Technology Resources Sharing Service. He obtained multiple internationally renowned awards, including the second prize of the 2016 China National Science and Technology Invention Award and the first prize of the 2010 Beijing Science and Technology Award.

Rongfang Bie received the MS and PhD degrees from Beijing Normal University, in 1993 and 1996, respectively. She is a professor with the College of Information Science and Technology, Beijing Normal University. She was a visiting scholar of the University of Cambridge in 2003. Her current research interests include Internet of Things, data mining, and knowledge engineering. She is a member of the IEEE.