Fifty-second Annual Allerton Conference
Allerton House, UIUC, Illinois, USA
October 1 - 3, 2014

Incentive Mechanism and Protocol Design for Crowdsourcing Systems

Hong Xie*, John C.S. Lui*, Joe Wenjie Jiang†, Wei Chen‡
*Department of Computer Science & Engineering, The Chinese University of Hong Kong
‡Microsoft Research Asia

Abstract— **Crowdsourcing systems such as Amazon Mechanical Turk, Yahoo! Answers, and Google Helpouts have attracted extensive attention over the past few years. In a crowdsourcing system, a large group of "workers" solve the tasks outsourced by "requesters". To make a crowdsourcing system sustainable, it is vital to attract users (both requesters and workers) to participate, and incentivize high-quality solutions. To achieve this objective, we design an effective incentive mechanism and reputation protocol. Our design incorporates various important elements of a crowdsourcing system such as workers having heterogeneous skill sets (i.e., some are "experts" while others are "novices"), and task assignment process, rating system, etc. Our incentive mechanism is composed of a rating system and a reward dividing scheme, and requires the system administrator to divide the reward based on requesters' rating on the solution quality. We derive the minimum reward needed so that "expert" workers are guaranteed provide high-quality solutions. We show that "novice" workers provide low-quality solutions, and our *reputation protocol* eliminates this undesirable behavior by tracking a worker's solution history and penalizing him when his reputation is poor. We apply repeated game-theoretic frameworks to quantify the impact of this reputation protocol on requesters' cost in guaranteeing high quality solutions.**

I. Introduction

Over the past decade, we have seen an increasing popularity of crowdsourcing systems. Numerous successful crowdsourcing websites have emerged on the Internet [27]. For example, the Amazon Mechanical Turk [1] is an online labor marketplace that allows requesters to oursource various tasks, e.g., image labeling, transcription, to a large group of workers. The Google Helpouts [7] is an online collaboration website that enables workers to provide online help, e.g., teaching guitar, teaching cooking, via online video, mobile devices, or computers, or requesters to seek real-time help. The Yahoo! Answers [26] is a question-and-answer website that allows users to both answer questions and post questions. Coined in 2005 [13], crowdsourcing is a promising production model to obtain needed services, content, or ideas, by eliciting the collective intelligence of a group of people, rather than from specialized employees in a company. Crowdsourcing has emerged as a promising paradigm to gather data as well. Typical examples include Linux, Wikipedia and Youtube. Crowdsourcing has achieved many successes, and it is considered to drive the future Internet businesses [14].

A typical crowdsourcing system is composed of *requesters* and *workers*. Usually a user can register to be a requester, a worker, or both of them. Requesters post tasks, and set a reward for each task. Workers solve tasks and return the solutions to requesters. Workers may have different skills, i.e., some are "experts" being good at solving tasks and others are "novices" being lack in desired knowledge. Finally, a requester selects a subset of the solution (usually the highest quality one) and grants the reward to contributing workers. After a task is finished, a requester also pays a transaction fee to the system. The reward attracts workers to participate, and it can be monetary [1], [5], non-monetary [24], or altruism [26], etc. High-quality solutions by workers encourage requesters to participate. To make a crowdsourcing website sustainable, it is of vital importance to consistently attract participation from both requesters and workers, and incentivize high-quality solutions [11], [20]. Low quality solutions, or lacking of user (requesters and workers) participation, may lead to the collapse of a crowdsourcing system. The main objective of this paper is to design effective incentive mechanisms and reputation protocols to attract users to participate and guarantee high quality solutions.

There are several challenges in attracting user participation or incentivizing high quality solutions. First, determining the appropriate reward that makes both requesters and workers happy is challenging. Requesters seek to elicit high quality solutions by paying a reward as small as possible. However, a small reward may fail to attract workers resulting in a task unsolved. Second, it is challenging to design effective incentive mechanisms and reputation protocols tracking workers' solution history so that workers are guaranteed to provide high-quality solutions. Workers may strategically exert low effort in solving tasks, leading to low quality solutions. Furthermore, it is possible that only "novice" workers (without the desired skills to solve that task) participate, whose solutions bring little benefit to requesters, no matter how much effort they exert. The heterogeneity of workers' skill sets, i.e., some are "novices", while others are "experts" (with the desired skills to solve a task), complicates the design of incentive mechanism and reputation protocol. In this paper, we address the above challenges. Our contributions are:

- We present a model to capture various important elements of a crowdsourcing system such as workers having heterogeneous skill sets (i.e., some are "experts" while others are "novices"), a probabilistic model for task assignment process, etc.
- We design an effective incentive mechanism composed of a rating system and a reward dividing scheme, and it

† Joe Wenjie Jiang is now working at Google

requires the system administrator to divide the reward based on requesters' rating on the solution quality. We derive the minimum reward needed so that "expert" workers are guaranteed provide high-quality solutions.

- Since "novice" workers will provide low quality solutions, we design a class of efficient *reputation protocols* to eliminate undesirable behavior. Our reputation protocol tracks a worker's solution history and penalizes him when his reputation is poor. We apply repeated game-theoretic frameworks to quantify the impact of this reputation protocol on requesters' overhead (i.e., cost in guaranteeing high quality solutions).

This is the outline of the paper. In §II, we present the system model and our incentive mechanism. In §III, we derive the minimum reward needed to guarantee that *expert* workers exert high effort. In §IV, we present the design and analysis of our reputation protocol. Related work is given in §V and we conclude in §VI.

II. Model

We present a model of crowdsourcing systems, our incentive mechanism, and an m-player game to characterize workers' strategic behavior under our incentive mechanism.

A. System Model

A typical crowdsourcing system operates with *requesters*, *workers*, *tasks* and *rewards*. Requesters post tasks and set a reward for each task, and workers solve tasks earning the reward. After a task is finished, a requester pays a transaction fee to the crowdsourcing system. It is common that crowdsourcing systems categorize tasks into different types. For example, Amazon Mechanical Turk categorizes tasks into "Transcription", "Object classification", "Website feedback", etc [15]. We consider a crowdsourcing system that contains $K \geq 1$ types of tasks. Consider a type $k \in \{1,\ldots,K\}$ task, a requester pays a transaction fee of $T_k > 0$ to the crowdsourcing system, and pays a reward of $r_k > 0$ to be distributed to contributing workers as well. In total, a requester pays $T_k + r_k$ in order to have a type k task completed. Without loss of generality, our analysis deals with one type of task, i.e., we drop the subscript k.

Workers have different skill sets. Specifically, some workers may be more knowledgeable than others in solving a task. Based on workers' skill level, we categorize workers into two types: *expert* indicating a worker having sufficient skill to complete a task, and *novice* indicating a worker's skill being insufficient. Note that this categorization is task sensitive, i.e., a worker can be an *expert* for one task, and be a *novice* for other tasks. We assume that this skill type is private information.

We propose a probabilistic model to describe task assignment process. Some crowdsourcing systems like Clickworker [4] assign tasks to workers by the system administrator, while others like Amazon Mechanical Turk allow workers to select tasks. We present a model that is general enough to supports all these cases. More concretely, we describe the interaction between workers and tasks via the following probabilistic model: in each task assignment, a task is assigned to a *novice* with probability $\beta \in [0,1]$, and it is assigned to an *expert*, with probability $1-\beta$. We assume that a task is assigned to different workers independently, and β is common (public) knowledge to all workers.

Our model considers that $m \geq 1$ workers work on a task denoted by $w_1,\ldots,w_m$. Let $\theta_i \in \{0,1\}$ denote the skill type of w_i, where $\theta_i = 1$ implies that w_i is an expert, and $\theta_i = 0$ implies that w_i is a novice. When a worker receives a task assignment, he can refuse (D) to solve it, or commit to solve this task by a high effort (H) or a low effort (L). Hence the action set of a worker is $\{H,L,D\}$. If a worker refuses a task, then this task will be reassigned until m workers commit to solve it. An *expert* worker exerting a high effort H is with cost c, and a *novice* exerting a high effort H is with cost c', where $c' > c > 0$. A worker (expert or novice) exerting a low effort L or refusing D a task is with zero cost. A high effort H solution by an *expert* brings a benefit of V to a requester, and a low effort solution, or one submitted by a *novice* (regardless of whatever he exerts H or L) brings zero benefit. Let $a_i \in \{H,L\}$ denote the action (or effort level) of w_i in solving a task. Then the solution submitted by w_i brings a benefit $V\mathbf{I}_{\{a_i=H,\theta_i=1\}}$, where $\mathbf{I}_{\{a_i=H,\theta_i=1\}} = 1$ if and only if $\theta_i = 1$ (i.e., w_i is an expert) and $a_i = H$ (i.e., w_i exerts high effort H), otherwise $\mathbf{I}_{\{a_i=H,\theta_i=1\}} = 0$. A requester receives m solutions in the end. We focus on the scenario that a requester only selects one with the highest quality. If there is a tie, a requester randomly picks one from this tie. Thus, these m solutions bring an overall benefit $V\max\left\{\mathbf{I}_{\{a_1=H,\theta_1=1\}},\ldots,\mathbf{I}_{\{a_m=H,\theta_m=1\}}\right\}$. Our model requires $V > r + T$ so that a crowdsourcing system is attractive to requesters.

It is possible that a requester only receives low quality solutions, i.e., low effort solutions by *expert* workers or solutions by novices. To eliminate this undesirable outcome, it is crucial to design effective incentive mechanisms and reputation protocols to guarantee that requesters receive high effort solutions by some *expert* workers.

B. Incentive Mechanism

We present an effective mechanism composed of a *binary rating system* and a *reward dividing scheme*, to incentivize high quality solutions. It operates as follows. Before a task begins, a requester submits its transaction fee T and associated reward r to the crowdsourcing system administrator. Then the system administrator allows m workers to work on a task, and workers submit solutions to the administrator. When all m solutions for a given task are submitted, the system administrator forwards them to the corresponding requester. A requester evaluates the quality of each solution, and then sends a "*feedback rating*" $\in \{-1,1\}$ to the administrator, indicating whether he is satisfied (i.e., positive rating 1), or unsatisfied (i.e., negative rating -1) with a solution. We emphasize that it is incentive-compatible for requesters to give "genuine" feedbacks. Note that only a high effort solution by an expert worker can fulfill the satisfaction of a requester, because only it brings a benefit higher than $r + T$. Let $\mathcal{R} : \{H, L\} \times \{0, 1\} \to \{-1, 1\}$ denote a map prescribing a rating for each possible quality of a solution. Then for all $i = 1, \ldots, m$ we have

$$\mathcal{R}(a_i, \theta_i) = \begin{cases} 1, & \text{if } a_i = H \text{ and } \theta_i = 1 \\ -1, & \text{otherwise} \end{cases}.$$

Note that a requester does not have any prior knowledge on the skill type or effort level of a worker, and he expresses ratings only relying on the solution quality. When the administrator collects all m ratings for a given task, he divides the reward r evenly among the workers who receive positive rating 1. If none of them receives a positive rating 1, all m workers equally share the reward r. One important reason for this reward dividing scheme is to avoid *denial of payment* by requesters, because the reward will not return to requesters. And this reward dividing scheme guarantees that a requester does not benefit by providing a false rating (e.g., receives a H solution by an expert but reports a negative rating -1).

It is possible that a worker can earn some rewards, even by providing a low effort L solution, e.g., all m workers providing low effort L solutions results in each of them receiving r/m. To overcome this problem, our incentive mechanism needs to ensure that at least one expert worker will provide high effort solutions. We next provide a game-theoretic analysis of our incentive mechanism and show how to incentivize experts to provide high quality solutions.

C. Formulating the m-player Game

We formulate an m-player game to characterize workers' strategic behavior in solving a task under our proposed incentive mechanism. In this m-player game, the players are m workers who participate in the same task, i.e., $w_1, \ldots, w_m$. Each player has the same strategy set $\{H, L\}$. Let $s_i \in \{H, L\}$ denote the strategy for w_i, and let $s_{-i} = [s_j]_{j \neq i}$ denote a vector of strategies for all players except w_i. Let $\theta_{-i} = [\theta_j]_{j \neq i}$ denote a vector of skill types for all players except w_i. We define the utility for a player as the reward minus the cost. Let $R_i(s_i, s_{-i}, \theta_i, \theta_{-i})$ denote the reward for player w_i. Based on our reward dividing scheme, we have:

$$\begin{aligned} &R_i(s_i, s_{-i}, \theta_i, \theta_{-i}) \\ &= \begin{cases} \dfrac{r}{\sum_{j=1}^{m} \mathbf{I}_{\{\mathcal{R}(s_j, \theta_j) = 1\}}}, & \text{if } \mathcal{R}(s_i, \theta_i) = 1, \\ \dfrac{r}{m} \mathbf{I}_{\{\{j | \mathcal{R}(s_j, \theta_j) = 1, \forall j \neq i\} = \emptyset\}}, & \text{otherwise.} \end{cases} \end{aligned}$$

To illustrate, consider $m = 2$, say two workers work on a task. Suppose both of them are *experts*. We then have $R_i(H, H, 1, 1) = R_i(L, L, 1, 1) = r/2$, $R_i(H, L, 1, 1) = r$. Let $u_i(s_i, s_{-i}, \theta_i, \theta_{-i})$ denote the utility for w_i:

$$\begin{aligned} &u_i(s_i, s_{-i}, \theta_i, \theta_{-i}) \\ &= \begin{cases} R_i(s_i, s_{-i}, \theta_i, \theta_{-i}) - c, & \text{if } s_i = H, \theta_i = 1 \\ R_i(s_i, s_{-i}, \theta_i, \theta_{-i}) - c', & \text{if } s_i = H, \theta_i = 0 \\ R_i(s_i, s_{-i}, \theta_i, \theta_{-i}), & \text{if } s_i = L \end{cases}. \end{aligned} \quad (1)$$

We emphasize that each player only knows its own exact type, i.e., an expert or a novice, but does not have any prior knowledge on other players' type.

We now introduce the notation of *strictly dominant strategy* to facilitate the analysis of the above m-player game. A *strictly dominant strategy* brings a higher utility than any other potential strategies.

Definition *2.1* **(Strictly Dominant Strategy):** Given that w_i is of skill type θ_i, then a strategy $s_{i,\theta_i} \in \{H, L\}$ is a strictly dominant strategy for w_i, if and only if for all $s_{-i} \in \{H, L\}^{m-1}$, $\theta_{-i} \in \{0, 1\}^{m-1}$, we have $u_i(s_{i,\theta_i}, s_{-i}, \theta_i, \theta_{-i}) > u_i(s'_i, s_{-i}, \theta_i, \theta_{-i}), \forall s'_i \in \{H, L\} \setminus \{s_{i,\theta_i}\}$.

That is, if w_i is of skill type θ_i, then a strictly dominant strategy is always the unique best response for w_i, no matter what strategies the other players play, and what types the other players are of. For example, if $\theta_i = 0$, then $s_{i,0}$ is the unique best response for w_i, and if $\theta_i = 1$, the $s_{i,1}$ is the unique best response for w_i. It is important to note that $s_{i,1}$ and $s_{i,0}$ can be different strategies.

The above m-player game does not capture the case that a worker refuses a task. In case of refusing a task, a worker does not participate in the above m-player game. We incorporate this case by defining the utility of refusing a task to be zero. We next show that our incentive mechanism can guarantee that *expert* workers exert high effort.

III. Induce Incentive via Appropriate Reward

We show that when only one worker $m = 1$ works on a task, it is impossible to incentivize high effort H solutions, no matter what reward we set. By increasing m to $m \geq 2$, our incentive mechanism can guarantee that *expert* workers exert high effort. We derive the minimum reward to achieve this property as well. Moreover, we show that *novice* workers will always free ride, no matter what reward we set. Finally, we derive the minimum number of workers needed, and the corresponding reward, to guarantee that at least one high effort solution by experts is collected with high probability.

A. Analysis for Single Worker $m = 1$

We show that when only one worker works on a task, workers will exert low effort L, no matter what reward we set. Each participating worker plays a single player game. We outline its utility matrix in Table I, which depicts the utility for the worker, the requester and the administrator. It is straightforward that L is a *strictly dominant strategy* for the worker, no matter what reward we set. Since $r > 0$, thus L brings a utility r higher than refusing a task. We summarize these results in the following lemma.

TABLE I
UTILITY MATRIX WHEN $m = 1$.

		Worker	**Requester**	**Admin.**
(Expert,	H	$r - c\ (r - c')$	$V - r - T\ (-r - T)$	$T\ (T)$
Novice)	L	$r\ (r)$	$-r - T\ (-r - T)$	$T\ (T)$

Lemma *3.1:* Consider our incentive mechanism, and a task is solved by only one worker. Then L is a strictly dominant strategy and gives a utility higher than refusing a task, no matter what reward we set.

Remark: It implies that workers will be attracted and they provide low effort L solutions. Hence, requesters only collect low quality solutions. We next show how to eliminate this undesirable result by allowing *two workers* to work on a task.

B. Analysis for Two Workers $m = 2$

When two workers work on a task, our proposed incentive mechanism guarantees *expert* workers to exert high effort by setting a reward $r > 2c$. Novice workers exert low effort L no matter what reward we set.

We first show that we can make H being a *strictly dominant strategy* and giving a utility higher than refusing a task for *expert* workers, if and only if $r > 2c$. When two workers work on a task $m = 2$, participating workers play a two player game. Without loss of generality, assume that w_1 is an *expert*. The player w_2 can be either an expert or a novice. Consider w_2 is a novice. We outline the utility matrix in Table II. Then it is straightforward that H gives a higher utility than L to w_1, no matter what strategies w_2 plays, if and only if $r > 2c$. And $r > 2c$ implies that exerting high effort H gives a utility $r - c$ higher than refusing a task. Similarly, we can show that these statements also hold when w_2 is an expert. We summarize them in the following lemma.

TABLE II
UTILITY MATRIX FOR THE M-PLAYER GAME($m = 2$).

		w_2**(novice)**	
		H	L
w_1**(expert)**	H	$r - c, -c'$	$r - c, 0$
	L	$r/2, r/2 - c'$	$r/2, r/2$

Lemma *3.2:* Consider our incentive mechanism, and a task is solved by two workers. Then for expert workers, H is a strictly dominant strategy and gives a utility higher than refusing a task, if and only if $r > 2c$.

Remark: It implies that by setting a reward $r > 2c$, our proposed mechanism guarantees that expert workers will be attracted to provide high effort solutions. Thus high quality solutions will be collected, if at least one *expert* gets in.

We now show that for *novice* workers, L is a *strictly dominant strategy*, and it gives a utility no less than refusing a task. Without any loss of generality, we assume that w_2 is a *novice*. The player w_1 can be either an expert or a novice. Consider w_1 is an expert. We outline the utility matrix in Table II. We observe that for w_2, the L brings a higher utility than H, and it gives a utility no less than refusing a task, no matter what reward we set. These statements also hold for the case that w_1 is a novice worker. We summarize these results in the following lemma.

Lemma *3.3:* Consider our incentive mechanism, and a task is solved by two workers. For novice workers, L is a strictly dominant strategy, and gives a utility no less than refusing a task, no matter what reward we set.

Remark: It implies that *novice* workers do not refuse tasks, but instead provide low effort (L) solutions for assigned tasks. Thus with probability β^2, two novice workers get in. In this case, a requester collects two low quality solutions, while costing $r + T$. This is the *novice free-riding risk.*

Definition *3.1* (**Novice Free-riding Risk**): We define novice free-riding risk as the scenario that all m workers are novices and they provide low effort solutions.

Combining Lemma 3.2 and Lemma 3.3, we obtain that such risk happens with probability β^2, when $m = 2$. And by setting $r > 2c$, a requester collects at least one high quality solution with probability $1 - \beta^2$. We next reduce this risk by allowing more workers to solve a task so that at least one *expert* worker will be involved to provide a high effort solution.

C. More than Two Workers $m \geq 2$

We first reduce the *novice free-riding risk* by allowing more workers to work on a task. Specifically, this is done by generalizing the results of Lemma 3.2, and Lemma 3.3 to $m \geq 2$ in the following lemma.

Lemma *3.4:* Consider our incentive mechanism, and a task is solved by $m \geq 2$ workers. For expert workers, H is a strictly dominant strategy and gives a utility higher than refusing a task, if and only if $r > mc$. For novice workers, L is a strictly dominant strategy and gives a utility higher than refusing a task, no matter what reward we set.

Proof: We first consider *expert* workers. Without loss of generality, suppose player w_i is an expert, and besides w_i there are $\ell \leq m - 1$ players that are experts. Suppose $\ell' \leq \ell$ of these ℓ players exert H. Then we have $u_i(H, s_{-i}, 1, \theta_{-i}) = r/(\ell' + 1) - c$, and $u_i(L, s_{-i}, 1, \theta_{-i}) = 0$ if $\ell' \geq 1$, and $u_i(L, s_{-i}, 1, \theta_{-i}) = r/m$ if $\ell' = 0$. Then if follows that $u_i(H, s_{-i}, 1, \theta_{-i}) - u_i(L, s_{-i}, 1, \theta_{-i})$ attains the minimum value $r/m - c$ when $\ell' = \ell = m - 1$. We therefore conclude that H is a strictly dominant strategy for w_i if and only if $r > mc$. The utility for an expert worker exerting high effort is at least $r/m - c$, which is larger than 0 (i.e., utility of refusing a task), if $r > mc$. Thus the proof for expert workers part is completed. Now consider *novice* workers. One can easily check that L is a strictly dominant strategy and gives a utility higher than refusing a task. ■

Remark: It states that when we allow more workers to work on a task, we need to increase the reward linearly with respect to m, so that expert workers will be attracted to provide high effort H solutions. The probability that all participating workers are novices is β^m. Thus, we can tradeoff reward against the probability that novice freeriding risk happens.

Let us explore the optimal tradeoff between minimum reward and the probability that novice freeriding risk happens. More precisely, we derive the minimum number of workers needed, and the corresponding reward, to guarantee that novice freeriding risk is avoided (or at least one high quality solution is collected) with a given probability.

Theorem *3.1:* Consider our incentive mechanism, at least one high effort solution by experts will be collected with probability at least $1 - \epsilon$, iff a task is solved by $\max\{\lceil \log_\beta \epsilon \rceil, 2\} \triangleq m'$ workers, and $r > m'c$.

Proof: This proof is similar to that of Lemma 3.4. ■

Table III presents some numerical results, where $\epsilon = 0.01$ (i.e., with probability 0.99). It depicts β, m', and r. One can observe that, as we increase β from 0.1 to 0.8, we increase the minimum number of workers m' from 2 to 21, and increase the corresponding reward from $2c$ to $21c$. In other words, we increase the reward significantly. This result suggests that a crowdsourcing system needs to increase the task assignment accuracy (or decrease β) so as to decrease the minimum reward, leading to more requesters being attracted.

TABLE III
REWARD NEEDED TO GUARANTEE AT LEAST ONE HIGH EFFORT SOLUTION BY EXPERTS WITH PROBABILITY 0.99.

β	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8
m'	2	3	4	6	7	10	13	21
r	$2c$	$3c$	$4c$	$6c$	$7c$	$10c$	$13c$	$21c$

Summary. One important property of our incentive mechanism is that *expert* workers can be guaranteed to exert high effort. However, novice workers exert low effort L instead of voluntarily refusing the task. This puts requester under the *novice free-riding risk*, i.e., collecting low quality solutions while costing $r + T$. Assigning a task to more workers may reduce this risk, however this also increases the minimum reward significantly. We next design a reputation system to guarantee novice workers refuse tasks voluntarily.

IV. Reputation System

We design a reputation system to eliminate the *novice free-riding risk*. Our reputation system tracks each worker's contribution history and penalizes him when his reputation is poor. We use a *repeated game framework* to characterize long-lived workers' strategic behavior. We apply the *one-shot deviation principle* to establish a subgame perfect equilibrium that *expert* workers exert high effort H, and *novice* workers refuse tasks voluntarily. We derive the minimum reward to guarantee that our established subgame perfect equilibrium is unique. Finally, we show that our proposed reputation system not only eliminates the *novice free-riding risk* but also reduces the reward payment for requesters.

A. Modeling Reputation System

Our objective is to design a reputation system so that expert workers exert high effort (H), and novice workers refuse tasks (D). We define it as our desirable strategy.

Definition *4.1 (***Desirable Strategy):** Let s^* denote our desirable strategy for workers that expert workers exert high effort (H), and novice workers refuse tasks (D).

If workers play the desirable strategy s^*, then all the solutions will be submitted by experts at high effort, i.e., high-quality solutions. Hence, a low quality solution indicates that a worker does not play the desirable strategy s^*.

Definition *4.2 (***Mistake):** A "mistake" of a worker w_i is a low quality solution, i.e., $\mathcal{R}(a_i, \theta_i) = -1$.

The above defined mistake does not capture that an expert worker refuses a task. Our incentive mechanism guarantees that expert workers are incentivized to exert high effort H.

We now describe the design of our reputation system. Workers in real-world crowdsourcing systems are long-lived and attempt to solve many tasks. Hence, one can induce extra incentives for workers to comply with the desirable strategy s^* by tracking the "*reputation levels*" of workers, and penalizing a worker if his reputation is poor. More precisely, we propose a reputation system to track the mistake history of each worker and it penalizes a worker if his number of mistakes exceeds a given threshold. Our reputation system tags each worker with a reputation index indicating the number of mistakes that a worker has made since the last reset. Each worker is initialized with zero mistakes. A mistake of a worker results in an increase of his reputation index by one. We emphasize that our reputation system does *not* assume any prior knowledge on workers' skill sets or effort level. When a worker's reputation index reaches a given threshold $W_g \geq 1$, the reputation system triggers a punishment that *disables* this worker's account (i.e., not allowed to participate in any crowdsourcing activity) for a given number of $W_p \geq 0$ time slots. We call W_g, W_p the *graceful window size* and the *punishing window size* respectively. After a worker is disabled for W_p time slots, the reputation system reset his reputation index (i.e., number of mistakes) to zero and activates his account. We denote this reputation protocol as the (W_g, W_p)*-mechanism.*

Formally, a worker's reputation is indexed by a 2-tuple $(\mathbf{I}_a, k)$, where $\mathbf{I}_a \in \{0, 1\}$ is an indicator variable indicating whether a worker is active ($\mathbf{I}_a = 1$) or disabled ($\mathbf{I}_a = 0$). For an active worker, say $\mathbf{I}_a = 1$, the value of k denotes his reputation index (or number of mistakes made since last reset). For a blocked worker, say $\mathbf{I}_a = 0$, the value of k denotes the number of time slots that he has been disabled. This transaction of our reputation mechanism can be described as follows:

$$(\mathbf{I}_a, k) \xrightarrow{(s_i,\theta_i)} \begin{cases} (1, k+1), & \text{if } \mathbf{I}_a = 1, k < W_g - 1, \\ & \text{and } \mathcal{R}(s_i, \theta_i) = -1 \\ (0, 0), & \text{if } \mathbf{I}_a = 1, k = W_g - 1, \\ & \text{and } \mathcal{R}(s_i, \theta_i) = -1 \\ (1, k), & \text{if } \mathbf{I}_a = 1, \\ & \text{and } (\mathcal{R}(s_i, \theta_i) = 1 \vee s_i = D) \\ (0, k+1), & \text{if } \mathbf{I}_a = 0, \text{ and } k < W_p - 1 \\ (1, 0), & \text{if } \mathbf{I}_a = 0, \text{ and } k = W_p - 1 \end{cases}$$

We emphasize that the system does *not* need to know the skill type θ_i or the strategy s_i of a worker in reputation updating, but rather the solution rating $\mathcal{R}(s_i, \theta_i)$. Figure 1 illustrates a transition diagram for a reputation index using a $(3, 5)$–mechanism, where the solid line represents the transition for *expert* workers, while the dashed line represents the transition for *novice* workers.

We next apply the *repeated game* framework to characterize the long-lived workers' strategic behavior under reputation system, so as to show how our reputation system eliminates the *novice free-riding risk*.

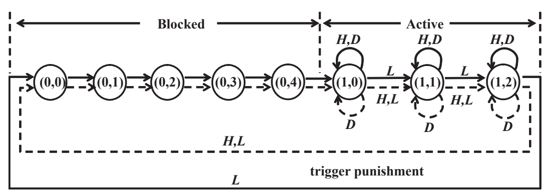

Fig. 1. Transition of a reputation index with $W_g = 3, W_p = 5$.

B. Repeated Game Formulation

We describe a repeated game formulation to characterize long-lived workers' strategic behavior under our reputation system. Recall that in Section II, we formulated an m–player *single-shot* (or single time slot) game to characterize workers' strategic behavior in solving one task. Now, consider a long-lived worker who attempts to solve many tasks, he repeatedly plays the m–player *single-shot* game, and we characterize his strategic behavior via the discounted long term utility with discounting factor $0<\delta<1$. Time is divided into slots, and we assume that at each time slot only one task is assigned to an active worker. If an active worker refuses a task assignment, he remains idle at that time slot. A worker remains idle if his account is disabled. Suppose that at time t a worker registers to solve a task. Let $s_i^t \in \{H, L\}$ denote his strategy at time t, and let $s_{-i}^t \in \{H, L\}^{m-1}$ denote a vector of strategies for other $m-1$ players. And let θ_i^t denote the type of w_i at time t, and let θ_{-i}^t denote a vector of types for other $m-1$ players. Then his single shot utility at time t is $u_i(s_i^t, s_{-i}^t, \theta_i^t, \theta_{-i}^t)$ derived in Equation (1). If a worker remains idle at time t (either refuses a task or his account is disabled by the reputation system), his utility is 0. Let $\mathbf{s}_i = \{s_i^t\}_{t=0}^{\infty}$ denote an infinite sequence of strategy profiles for a worker. The long term discounted utility for this worker is: $u_i^{\infty}(\mathbf{s}_i) = \sum_{t=0}^{\infty} \delta^t \left(1 - \mathbf{I}_{\{\text{idle}\}}^t\right) u_i(s_i^t, s_{-i}^t, \theta_i^t, \theta_{-i}^t)$, where $\mathbf{I}_{\{\text{idle}\}}^t = 1$ if the status of a worker is idle at time t, otherwise 0. We assume that workers' strategies at different time slots are independent. We next show establish a subgame perfect equilibrium, where workers comply with s^*.

C. Establishing Subgame Perfect Equilibrium

We apply one-shot deviation principle to establish a subgame perfect equilibrium that workers comply with the desirable strategy s^*.

We first apply the *one-shot deviation principle* to derive the necessary and sufficient condition that a worker would not unilaterally deviate from s^* if everyone else complies with s^*. We say a worker is *compliant* if he always complies with s^*. Now we restrict our attention to the scenario that all workers are compliant. Let $u^{\infty}(\mathbf{I}_a, k)$ denote the long term utility of a compliant worker with starting reputation index $(\mathbf{I}_a, k)$. In the following lemma, we derive the necessary and sufficient condition that $u^{\infty}(\mathbf{I}_a, k)$ must satisfy so that a worker would not unilaterally deviate.

Lemma *4.1:* Consider our incentive mechanism that allows m workers to solve a task, and a (W_g, W_p)–reputation system. A worker with reputation index $(1, k), \forall k = 0, 1, \ldots, W_g - 1$, will not unilaterally deviate (if everyone else are compliant) if and only if $\Delta u^{\infty}(1, k) > 0$, and $r > mc$, where $\Delta u(1, k) = u^{\infty}(1, k) - u^{\infty}(1, k+1)$, if $k < W_g - 1$, and $\Delta u(1, W_g - 1) = u^{\infty}(1, W_g - 1) - u^{\infty}(0, 0)$.

Proof: Consider an expert worker with reputation $(1, k)$ and is assigned a task. His long term utility starting from this reputation index is: $\delta u^{\infty}(1, k)$ if he refuses this task, $\delta u^{\infty}(1, k+1)$ if he exerts low effort. and $\delta u^{\infty}(1, k) + \frac{r}{m} - c$ if he exerts high effort. Hence he will exert high effort if and only if $\delta u^{\infty}(1, k) + \frac{r}{m} - c > \max\{\delta u^{\infty}(1, k), \delta u^{\infty}(1, k+1)\}$. Namely we need $r > mc$ and $\Delta u(1, k) > c - \frac{r}{m}$. Similarly, if this worker is a novice, he will not not unilaterally deviate from s^*, if and only if $\Delta u(1, k) > 0$. ■

Given the above lemma, we next show that if the graceful window size is $W_g \geq 2$, then novice workers will deviate from s^* when they are with reputation index $(1, k), k = 0, 1, \ldots, W_g - 2$, no matter what reward we set.

Theorem *4.1:* Consider our proposed incentive mechanism that allows m workers to solve a task, and a (W_g, W_p)–reputation system. If $W_g \geq 2$, novices will deviate from s^* when they are with reputation index $(1, k), k = 0, 1, \ldots, W_g - 2$, no matter what reward we set.

Proof: Suppose that all workers comply with s^*. Consider an active worker with reputation index $(1, k)$, his expected single shot utility is $E[u_i(s_i, s_{-i}, \theta_i, \theta_{-i})] = (1-\beta)E[u_i(s_i, s_{-i}, \theta_i, \theta_{-i})|\theta_i = 1] + \beta E[u_i(s_i, s_{-i}, \theta_i, \theta_{-i})|\theta_i = 0] = (1-\beta)\,(r/m - c)$. We can then express his long term utility starting from reputation index $(1, k)$ as $u^{\infty}(1, k) = \sum_{t=0}^{\infty} E[u_j(s^*, s_{-j}|r_i)] = \frac{1-\beta}{1-\delta}\,(r/m - c)$, for all $k = 0, \ldots, W_g - 1$. Then by applying Lemma 4.1, we complete this proof. ■

Remark: The reason is that a novice worker will not be penalized by making a mistake (i.e., providing a low quality solution), when he is with reputation index $(1, k), k = 0, 1, \ldots, W_g - 2$. And for the single shot game, L is a strictly dominant strategy and gives a utility no less than refusing a task for novice workers.

The above theorem implies that it is impossible to guarantee that all workers comply with s^*, if the graceful window size is no smaller than 2 (i.e., $W_g \geq 2$). In the following theorem we show that this undesirable outcome can be eliminated by setting the graceful widow size to be one (i.e., $W_g = 1$). We also derive the minimum reward and the appropriate punishing window size to achieve this encouraging outcome.

Theorem *4.2:* Consider our proposed incentive mechanism that allows $m \geq 2$ workers to work on a task, and a (W_g, W_p)–reputation system. If $W_g = 1$ and $W_p \geq 1$, each player will not unilaterally deviate from s^*, iff $r > mc$.

Proof: Applying lemma 4.1, we only need $\Delta u^{\infty}(1, 0) > 0$, and $r > mc$. As derived in the proof of Theorem 4.1, the long term utility for an active with reputation index $(1, k)$ is $u^{\infty}(1, k) = \frac{1-\beta}{1-\delta}\,(r/m - c)$. Then it follows that the long term utility for a disabled worker with reputation index $(0, k)$ is $u^{\infty}(0, k) = \delta^{W_p - k} u^{\infty}(1, 0) = \delta^{W_p - k} \frac{1-\beta}{1-\delta}\,(r/m - c)$. Hence, $\Delta u^{\infty}(1, 0) = (1 - \delta^{W_p}) \frac{1-\beta}{1-\delta}\,(r/m - c)$. Then it follows that $\Delta u^{\infty}(1,0)>0$, iff $W_p \geq 1$, and $r > mc$. ■

Remark: It implies that it is possible to sustain compliance by workers. To guarantee that workers always play s^*, we need to show the uniqueness of the above derived subgame perfect equilibrium. We next show how to make it unique.

D. Uniqueness of the Subgame Perfect Equilibrium

We now show the uniqueness of the subgame perfect equilibrium derived in Theorem 4.2. More precisely, this is achieved by an appropriate increase on the reward payment. In the following theorem we derive the minimum reward, the minimum punishing window size, so that the subgame perfect equilibrium derived in Theorem 4.2 is unique.

Theorem *4.3:* Consider our proposed incentive mechanism that allows $m \geq 2$ workers work on a task, and a (W_g, W_p)–reputation system. Suppose $\delta > 1/\left(1+\frac{1-\beta}{\beta^{m-1}}\right)$, $W_g = 1$, and $W_p \geq \left\lceil \ln\left(1-\frac{\beta^{m-1}}{1-\beta}\frac{1-\delta}{\delta}\right)/\ln\delta \right\rceil$. If reward r satisfies

$$r > mc\left(1-\frac{1-\delta}{(1-\delta^{W_p})\delta}\frac{\beta^{m-1}}{1-\beta}\right)^{-1}, \tag{2}$$

then the subgame perfect equilibrium established in Theorem 4.2 is unique.

Proof: Please refer to the appendix for derivation. ■

Remark: That is if workers are patient enough (i.e., $\delta > (1+(1-\beta)/\beta^{m-1})^{-1}$), workers can be guaranteed to play s^* by selecting a proper punishing window size and reward. The reward is decreasing with respect to W_p.

Table IV presents some numerical results on the minimum reward derived in Equation (2). Let us fix $m=2, \delta=0.9$ and examine the impact of punishing window size and task assignment accuracy (or β) on the minimum reward. When $\beta=0.2$, say with probability 0.2, a task is assigned to a novice, as we increase the punishing window size from 10 to 40, we decrease the minimum reward from $2.089c$ to $2.058c$, a slight decrease on the minimum reward. Similar results are shown when $\beta = 0.4, 0.6$ respectively. When $W_p = 10$, as we increase β from 0.2 to 0.6, we increase the minimum reward from $2.089c$ to $2.688c$, an around 30% increase. Similar results are shown when $W_p = 20, 30, 40$ respectively. These results suggest a crowdsourcing system to increase the task assignment accuracy or increase the punishing window size. We now fix $\beta = 0.4, W_p = 40$ and examine the impact of m and δ on the minimum reward. When $\delta = 0.7$, as we increase m from 2 to 5, we increase the minimum reward from $2.800c$ to $5.093c$, a significant increase. When $m=2$, as we increase δ from 0.7 to 0.9, we decrease the reward from $2.8c$ to $2.163c$. Namely, the more patient the worker, the smaller the reward. Similar results are shown when $m = 3, 4, 5$ respectively. Finally the reward in Table IV are smaller than that in Table III. Namely, our reputation system not only eliminates the novice free-riding risk but also reduces the reward payment in guaranteeing high quality solutions.

TABLE IV
IMPACT OF β, W_p, δ, m ON THE MINIMUM REWARD TO GUARANTEE WORKERS PLAY s^*.

$m=2, \delta=0.9$				
W_p	10	20	30	40
$r(\beta=0.2)$	$2.089c$	$2.065c$	$2.060c$	$2.058c$
$r(\beta=0.4)$	$2.257c$	$2.184c$	$2.168c$	$2.163c$
$r(\beta=0.6)$	$2.688c$	$2.468c$	$2.421c$	$2.407c$

$\beta=0.4, W_p=40$				
m	2	3	4	5
$r(\delta=0.7)$	$2.800c$	$3.387c$	$4.192c$	$5.093c$
$r(\delta=0.8)$	$2.400c$	$3.214c$	$4.110c$	$5.054c$
$r(\delta=0.9)$	$2.163c$	$3.093c$	$4.049c$	$5.024c$

V. Related work

The research on *crowdsourcing* has attracted extensive attention. Researchers explored many aspects of crowdsourcing, for example solution quality management [16], [19], cheat detection [9], new applications design based on crowdsourcing [8], and incentive design [20], etc. A recent survey on crowdsourcing systems is in [27]. Our paper focuses on incentive design for crowdsourcing applications.

Our work is related to the works on task pricing [2], [3], [6], [12], [21], [22], [23]. The objective of these works is to determine the proper reward. Authors in [2], [12] developed approaches to price tasks from the data mining perspective, where they developed models to infer workers' wage from data. A variety of task pricing approaches from economic perspective were proposed in [3], [6], [21], [22], [23]. They apply techniques from microeconomics to designed mechanisms to determine the reward automatically or dynamically. For example, authors in [23] proposed a multi-armed bandit based mechanism to price tasks, and authors in [3], [6], [22] applied auction theory to design pricing mechanisms. However, their works did not address the social dilemma challenge: If requesters pay before tasks begin, free-riding by workers may happen, and if requesters pay after tasks complete, denial of payment by requesters may happen. Our work addresses this challenge. Furthermore, they used a single shot game to model workers' strategic behavior in solving tasks, while we use a recreated game model to characterize long-lived workers' strategic behavior and study the impact of reputation systems.

Our work is also related to works on incentive mechanism (or protocol) design. Their objective is to incentivize high-quality contributions. Authors in [17] proposed incentive protocols for question-and-answer websites. They used a one-shot game to model workers' strategic behavior in answering questions. Our work designs incentive mechanism and reputation protocols for general crowdsourcing applications. Motivated by the seminal work [18] on social norm and reputation system, authors in [10], [25], [28] developed several incentive protocols based on reputation systems. Our work is different form theirs in the following four technical aspects. First, our work avoids free-riding or denial of payment by a payment mechanism via administrator, while their models [10], [25], [28] paid workers either after a task completes or before a task begins. Second, they assumed a perfect crowdsourcing system scenario that all workers are *experts*. Our model considers a practical scenario that some workers are *experts* while others are *novices*, and our proposed incentive mechanism allows m workers to solve a task, so as to increase the chance that at least one high quality solution would be collected. Third, they did not consider the impact of task assignment algorithm on incentive mechanism design. Our work quantifies the impact of task assignment algorithm on the design of incentive mechanisms and reputation protocols.

VI. Conclusion and Future Work

This paper designed an effective incentive mechanism and efficient reputation protocol for crowdsourcing systems. Our design incorporates various important elements of a crowdsourcing system such as workers having heterogeneous skill sets (i.e., some are "experts" while others are "novices"), and task assignment process, rating system, etc. And we quantified the impact of these elements on our incentive mechanism and reputation protocol. Our incentive mechanism is composed of a rating system and a reward dividing scheme, and requires the system administrator to divide the reward based on requesters' rating on the solution quality. We derived the minimum reward needed in order to guarantee that "expert" workers provide high-quality solutions. We showed that "novice" workers will provide low-quality solutions, and our *reputation protocol* eliminates this undesirable behavior, by tracking a worker's solution history and penalizing him when his reputation is poor. We applied repeated game-theoretic frameworks to quantify the impact of reputation protocol on requesters' overhead (i.e., cost in guaranteeing high quality solutions).

In our future work, we plan to explore the impact of human factors such as biases in solution rating, on our incentive mechanism and reputation protocol.

References

[1] Amazon Mechanical Turk. `https://www.mturk.com`.
[2] D. F. Bacon, D. C. Parkes, Y. Chen, M. Rao, I. Kash, and M. Sridharan. Predicting your own effort. In *Proc. of AAMAS*, 2012.
[3] S. Chawla, J. D. Hartline, and B. Sivan. Optimal crowdsourcing contests. In *Proc. of SODA*, 2012.
[4] Click Worker. `http://www.clickworker.com/`.
[5] Cloudcrowd. `http://www.cloudcrowd.com`.
[6] D. DiPalantino and M. Vojnovic. Crowdsourcing and all-pay auctions. In *Proc. of ACM EC*, 2009.
[7] Google Helpouts. `https://helpouts.google.com/home`.
[8] J. Heer and M. Bostock. Crowdsourcing graphical perception: Using mechanical turk to assess visualization design. In *Proc. of ACM CHI*, 2010.
[9] M. Hirth, T. Hossfeld, and P. Tran-Gia. Cost-optimal validation mechanisms and cheat-detection for crowdsourcing platforms. In *Proc. of IMIS*, 2011.
[10] C.-J. Ho, Y. Zhang, J. Vaughan, and M. van der Schaar. Towards social norm design for crowdsourcing markets. In *Proc. of HCOMP*, 2012.
[11] J. Horton. Online labor markets. *Internet and Network Economics*, pages 515–522, 2010.
[12] J. J. Horton and L. B. Chilton. The labor economics of paid crowdsourcing. In *Proc. of ACM EC*, 2010.
[13] J. Howe. The rise of crowdsourcing. *Wired magazine*, 14(6):1–4, 2006.
[14] J. Howe. Crowdsourcing: Why the power of the crowd is driving the future of business. *Crown Business*, 2008.
[15] P. G. Ipeirotis. Analyzing the Amazon Mechanical Turk marketplace. *XRDS*, 17(2):16–21, Dec. 2010.
[16] P. G. Ipeirotis, F. Provost, and J. Wang. Quality management on Amazon Mechanical Turk. In *Proc. of HCOMP*, 2010.
[17] S. Jain, Y. Chen, and D. C. Parkes. Designing incentives for online question and answer forums. In *Proc. of ACM EC*, 2009.
[18] M. Kandori. Social norms and community enforcement. *The Review of Economic Studies*, 59(1):63–80, 1992.
[19] D. R. Karger, S. Oh, and D. Shah. Efficient crowdsourcing for multi-class labeling. In *Proc. of ACM SIGMETRICS*, 2013.
[20] W. Mason and D. J. Watts. Financial incentives and the "performance of crowds". *SIGKDD Explor. Newsl.*, 11(2):100–108, May 2010.
[21] Y. Singer and M. Mittal. Pricing mechanisms for crowdsourcing markets. In *Proc. of WWW*, 2013.
[22] A. Singla and A. Krause. Truthful incentives in crowdsourcing tasks using regret minimization mechanisms. In *Proc. of WWW*, 2013.
[23] L. Tran-Thanh, S. Stein, A. Rogers, and N. R. Jennings. Efficient crowdsourcing of unknown experts using multi-armed bandits. In *Proc. of ECAI*, 2012.
[24] L. Von Ahn. Games with a purpose. *Computer*, 39(6):92–94, 2006.
[25] Y. Xiao, Y. Zhang, and M. van der Schaar. Socially-optimal design of crowdsourcing platforms with reputation update errors. In *Proc. of IEEE ICASSP*, 2013.
[26] Yahoo! Answers. `http://answers.yahoo.com`.
[27] M.-C. Yuen, I. King, and K.-S. Leung. A survey of crowdsourcing systems. In *Proc. of IEEE Socialcom*, 2011.
[28] Y. Zhang and M. van der Schaar. Reputation-based incentive protocols in crowdsourcing applications. In *Proc. of IEEE INFOCOM*, 2012.

Appendix

Proof of Theorem 4.2: Applying Lemma 3.4 one obtains that H is a strictly dominant strategy for *expert* workers, because Inequality (2) implies that $r > mc$. Thus expert workers will always comply to s^*, irrespective of whatever strategies the other workers play. Now we consider a reduced repeated game, where expert workers comply to s^*. We show that in this reduced repeated game, novice workers will comply to s^*, irrespective of whatever strategies other novice workers play. Consider a novice worker at state $(1,k)$. Suppose he is assigned a task. If he refuses this task assignment, then his long term utility starting form $(1,k)$ will be $\delta u^\infty(1,k)$. If he exerts a low effort L, his long term utility will be $\delta u^\infty(1,k+1) + E[u_i(L, s_{-i}, 0, \theta_{-i})]$. Note that for the single shot game, H is strictly dominated by L for novice workers. Thus this novice worker will refuse tasks, irrespective of other novice workers strategies, if and only if $\delta u^\infty(1,k) > \delta u^\infty(1,k+1) + E[u_i(L, s_{-i}, 0, \theta_{-i})]$. Note that expert workers exert high effort H. We then have $E[u_i(L, s_{-i}, 0, \theta_{-i})] \leq r\beta^{m-1}/m$. Therefore we only need

$$\delta u^\infty(1,k) > \delta u^\infty(1,k+1) + r\beta^{m-1}/m. \tag{3}$$

One can easily obtain $u^\infty(1,k) = \frac{1-\beta}{1-\delta}(r/m - c)$, and $u^\infty(0,k) = \delta^{W_p - k}\frac{1-\beta}{1-\delta}(r/m - c)$. Substituting them into Inequality (3) we expand Inequality (3) as

$$\left(1 - \frac{1-\delta}{\delta}\frac{\beta^{m-1}}{1-\beta}\frac{1}{1-\delta^{W_p}}\right)\frac{r}{m} > c. \tag{4}$$

To make this inequality hold, we need $1 - \frac{1-\delta}{\delta}\frac{\beta^{m-1}}{1-\beta} > 0$, which implies that $W_p \geq \lceil \ln(1 - \frac{\beta^{m-1}}{1-\beta}\frac{1-\delta}{\delta})/\ln\delta \rceil$ and $\delta > 1/(1 + \frac{1-\beta}{\beta^{m-1}})$. Solving Inequality (4) we obtain $r > mc(1 - \frac{1-\delta}{\delta}\frac{\beta^{m-1}}{1-\beta}\frac{1}{1-\delta^{W_p}})^{-1}$. ■