# Modeling Crowdsourcing Systems: Design and Analysis of Incentive Mechanism and Rating System

Hong Xie John C.S. Lui
Department of Computer Science & Engineering
The Chinese University of Hong Kong
{hxie,cslui}@cse.cuhk.edu.hk

## 1. INTRODUCTION

Over the past few years, we have seen an increasing popularity of crowdsourcing services [5]. Many companies are now providing such services, e.g., Amazon Mechanical Turk [1], Google Helpouts [3], and Yahoo! Answers [8], etc. Briefly speaking, crowdsourcing is an online, distributed problem solving paradigm and business production platform. It uses the power of today's Internet to solicit the collective intelligence of large number of users. Relying on the *wisdom of the crowd* to solve posted tasks (or problems), crowdsourcing has become a promising paradigm to obtain "solutions" which can have *higher quality* or *lower costs* than the conventional method of solving problems via specialized employees or contractors in a company.

Typically, a crowdsourcing system operates with three basic components: *Users*, *tasks* and *rewards*. Users are classified into *requesters* and *workers*. A user can be a requester or a worker, and in some cases, a user can be a requester/worker at the same time. Requesters outsource tasks to workers and associate each task with certain rewards, which will be granted to the workers who solve the task. Workers, on the other hand, solve the assigned tasks and reply to requesters with solutions, and then take the reward, which can be in form of money [1], entertainment [7] or altruism [8], etc.

To have a successful crowdsourcing website, it is pertinent to attract high volume of participation of users (requesters and workers), and at the same time, solutions by workers have to be of high quality. In this paper we design a rating system and a mechanism to encourage users to participate, and incentivize workers to high quality solutions. First, we develop a game-theoretic model to characterize workers' strategic behavior. We then design a class effective incentive mechanisms which consist of a *task bundling scheme* and a *rating system*, and pay workers according to solution ratings from requesters. We develop a model to characterize the design space of a class of commonly users rating systems–*threshold based rating system*. We quantify the impact of such rating systems, and the bundling scheme on reducing requesters' reward payment in guaranteeing high quality solutions. We find out that a simplest rating system, e.g., two rating points, is an effective system in which requesters only need to provide binary feedbacks to indicate whether they are satisfied or not with a solution.

*MAMA* 2014 Austin, Texas, USA


## 2. MODEL

We present the system model and the design of our incentive mechanism. We then analyze the incentive mechanism via the *game-theoretic technique*, showing its effectiveness.

### 2.1 System Model

Consider a crowdsourcing system which categorizes tasks into $K$ types. This is common, for example, in "*Yahoo! Answers*", it contains 25 types of tasks ranging from "*Health*" to "*Travel*" [8]. Users of a crowdsourcing system are classified into *requesters* and *workers*. Requesters outsource tasks to workers and at the same time, associate each type $k$ task with a reward of $r_k$, $k \in \{1, \dots, K\}$. The reward $r_k$ will be granted to the workers who make contributions to the corresponding task. For a type $k$ task, a requester also pays $T_k$ to the crowdsourcing system as service charge. We focus on one task type in our analysis, for it can be generally applied to all task types. We thus drop the subscript.

A task is assigned to *only one* worker. We capture the scenario that a task requires many workers as follows. A task can be divided into many copies and each copy requires one worker. Note that some service allows requesters to pick workers, such as Google Helpouts, while others practice the other way, such as [4]. We emphasize that our model support both cases. A worker can exert $L \geq 2$ levels of effort $\mathcal{L} = \{1, \dots, L\}$ in solving a task, which results in $L$ levels of contribution $\mathcal{C}_L = \{C_1, \dots, C_L\}$. We assume that $C_L \succ C_{L-1} \succ \dots \succ C_1$, where $C_i \succ C_j$ represents that contribution $C_i$ is higher than $C_j$. For the ease of presentation, we use $\{C_1, \dots, C_L\}$ to denote the action set for workers. When a worker acts with $C_i$, it means the worker exerts the $i$-th level of effort to solve the task. The cost in making a $C_j$ contribution to a task is denoted as $c_j$, where $c_L > c_{L-1} > \dots > c_1 = 0$. Here, we use $c_1 = 0$ to model the the "*free-riding*" scenario from workers. For a task, if a worker exerts $C_j$ to provide a solution, then it brings a benefit of $V_j$ to a requester, where $V_L > V_{L-1} > \dots > V_1 = 0$. Again, $V_1 = 0$ models *free-riding* because $c_1 = 0$. We require $V_L > r + T$, which induces incentives for requesters to participate. And $V_\kappa < r + T, \forall \kappa < L$, means that level $\kappa$ contribution is not incentive-compatible. The objective of this work is show how to incentivize workers to exert $C_L$, the highest possible contribution to solve the assigned task.

### 2.2 Incentive Mechanism Design

We consider a class of incentive mechanisms which consist of two key components: a *bundling scheme*, and a *rating system*. Tasks are completed via transactions under a *task* *bundling scheme*, which can be precisely described as follows. When posting a task, a requester submits its reward $r$ and service charge $T$ to the *administrator*. The administrator bundles $n \geq 1$ tasks of similar type. Once a task is solved, a worker submits its solution to the administrator. After all tasks within a bundle are solved, the administrator delivers them to the corresponding requesters. Requesters provide feedbacks on these solutions to the administrator in the form of solution rating. In particular, requesters rate solutions such that a rating $i$ indicates that the solution was solved with the $C_i$ level of contribution. Note that solutions are independent, and a requester can only express ratings of solution to her task. Finally, when all feedback ratings for a bundle are collected, the crowdsourcing administrator divides the total reward, which is $nr$, to all workers engaged in that bundle. Specifically, the worker who receives the highest rating takes all the reward. When there is a tie, the administrator divides the total reward $nr$ evenly among the tie. We call this reward scheme as "*winner takes all scheme*".

**Remark:** We call the above bundling scheme the *n-bundling scheme*. A requester will not benefit by intentionally providing false ratings, since the reward will not be returned. We next show how our incentive mechanism guarantees workers exerting $C_L$.

## 2.3 Induce Incentive via Pricing

With the above incentive mechanism, we apply the *game-theoretic technique* to derive the desired amount of rewards such that workers are guaranteed to exert $C_L$.

Consider our proposed incentive mechanism, we formulate an $n$-player game to capture the strategic behavior of workers in solving tasks. Players of this game are $n$ workers engaged in a bundle and we denote them as $w_1, \ldots, w_n$. The action set for a player is $\{C_1, \ldots, C_L\}$. Let $s_j$ denote the strategic action of worker $w_j$. Let $s_{-j} = [s_\kappa]_{\kappa \neq j}$ be a vector of strategic actions for all players except $w_j$. We use the notation $u_j(s_j, s_{-j}|r)$ to denote the utility for player $w_j$ under the strategy profile $(s_j, s_{-j})$, which is defined as the reward minus cost. The utility of player $w_j$ is:

$$u_j(s_j, s_{-j}|r) = R_j(s_j, s_{-j}|r) - c_\kappa, \quad \text{if } s_j = C_\kappa, \qquad (1)$$

where $R_j(s_j, s_{-j}|r)$ denotes the reward for worker $w_j$ under the strategy profile $(s_j, s_{-j})$. We express $R_j(s_j, s_{-j}|r)$ under the *winner takes all scheme* as

$$R_j(s_j, s_{-j}|r) = \begin{cases} \dfrac{nr}{\sum_{\kappa=1}^{n} \mathbf{I}_{\{s_\kappa = s_j\}}}, & \text{if } s_j = \max_\kappa s_\kappa \\ 0, & \text{otherwise.} \end{cases} \qquad (2)$$

Our objective is to guarantee each player in the above game plays $C_L$. One sufficient condition to achieve this: the strategy profile $(C_L, \ldots, C_L)$ is a unique "Nash Equilibrium".

**Definition** 2.1. *We define $(C_L, ..., C_L)$ as our desirable Nash equilibrium that all workers solve the task at $C_L$ level.*

We present a formal approach to show the uniqueness of the desirable Nash equilibrium, which relies on elimination of strictly dominated strategies, in the following lemma.

**Definition** 2.2. *A strategy $s_i \in \mathcal{C}_L$ is a strictly dominated strategy for player i if there exists some $s'_i \in \mathcal{C}_L$ such that $u_i(s'_i, s_{-i}|r) > u_i(s_i, s_{-i}|r)$, for all $s_{-i} \in \mathcal{C}_L^{n-1}$.*

**Lemma** 2.1 ([2]). *Suppose $(s_1^*, \ldots, s_n^*)$ is a pure Nash equilibrium for an $n$–player game. If iterated elimination of strictly dominated strategies eliminates all but the strategies $(s_1^*, \ldots, s_n^*)$, then it is a unique Nash equilibrium.*

We next derive the minimum amount of rewards to guarantee $(C_L, ..., C_L)$ being a unique Nash equilibrium. Specifically, for the 1-*bundling scheme*, it is impossible to achieve this (Lemma 2.2). However, this can be achieved if the bundle size is $n \geq 2$ without increasing the amount of desired rewards (Theorem 2.1).

**1-*bundling scheme.*** We express the utility for the worker, requester and the administrator in Table 1. From Table 1, one can observe that the *dominant strategy* for the worker is $C_1$. Namely, a worker will simply *free-ride* without making any contribution to solve the task. Another way to look at this result is that the reward $r$ is surely given to this worker independent of her effort or contribution (because the bundle size is one). Hence, there is no incentive for the worker to exert a higher effort.

| | | Worker | Requester | Admin. |
|---|---|---|---|---|
| Worker | $C_L$ | $r - c_L$ | $V_L - r - T$ | $T$ |
| | $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ |
| | $C_1$ | $r - c_1$ | $V_1 - r - T$ | $T$ |

**Table 1: Utility matrix under 1-*bundling scheme.***

**Lemma** 2.2. *Consider our proposed incentive mechanism, and a 1-*bundling scheme, *workers will always exert $C_1$ contribution, no matter what reward we set.*

**Remark.** It states that there is no way to incentivize the single worker to contribute $C_L$. We next show how to eliminate this undesirable result by bundling more than one task.

| | | $w_2$: $C_3$ | $w_2$: $C_2$ | $w_2$: $C_1$ |
|---|---|---|---|---|
| $w_1$ | $C_3$ | $r-c_3, r-c_3$ | $2r - c_3, -c_2$ | $2r - c_3, -c_1$ |
| | $C_2$ | $-c_2, 2r - c_3$ | $r - c_2, r - c_2$ | $2r - c_2, -c_1$ |
| | $C_1$ | $-c_1, 2r - c_3$ | $-c_1, 2r - c_2$ | $r-c_1, r-c_1$ |

**Table 2: Utility matrix under the 2-*bundling scheme.***

**2-*bundling scheme.*** We show that under this bundling scheme, we can incentivize high quality contributions via setting a proper reward. To illustrate, consider an example with three levels of contribution $L = 3$. We express the utility matrix for this example in Table 2, in which one can observe that the strategy profile $(C_3, C_3)$ is a unique Nash Equilibrium if and only if $r > c_3 - c_1 = c_3$. We generalize this positive result to $L$ levels of contribution in the following lemma. We define critical value to present the lemma.

**Definition** 2.3. *We define the "critical value" $\underline{r}$ as the minimum amount of reward to incentivize $C_L$ contribution.*

**Lemma** 2.3. *Consider our proposed incentive mechanism, and a 2–bundling scheme. The strategy profile $(C_L, C_L)$ is a unique Nash equilibrium if and only if $r > \underline{r} = c_L$.*

**Proof:** It can be easily proved by applying Lemma 2.1. ■

**Remark.** It implies that workers will contribute $C_L$, if they *do not collude*. If collusion is allowed, the best strategy for them is $(C_1, C_1)$. One way to eliminate this risk is by bundling *more* tasks so to guarantee that at least one worker will not collude. We next prove that increasing the bundle size does not increase the reward payment for requesters.

**Theorem** 2.1. *Consider our proposed incentive mechanism, and an $n$–bundling scheme ($n \geq 2$). The strategy profile $(C_L, \ldots, C_L)$ is a unique Nash equilibrium iff $r > \underline{r} = c_L$.*

**Proof:** This proof is similar to that of Lemma 2.3. ■

**Summary.** Our model thus far considers a rating system with a small number of contribution level. When $L$ is large, it may be difficult for a requester to express a rating accurately, i.e., the time or cognitive cost will be high [6]. Let us consider a rating system which can address this challenge.

# 3. MODELING RATING SYSTEMS

We present a model to characterize the design space of a class of commonly used rating system – *threshold based rating system*. We also quantify its impact on requests' reward payment. We determine a binary rating system, i.e., two rating points indicating satisfied or not, is sufficient to guarantee highest contributions $C_L$.

## 3.1 Threshold Based Rating Systems

Many crowdsourcing services adopt *threshold based rating systems* (TBR), where the quality of a solution below a "threshold" receives the lowest rating, which may incur some warnings or punishments, etc, to a worker. We develop a model to characterize the design space of such rating systems, and quantify its impact on the critical value.

A *threshold based rating system* is a triplet $\langle \mathcal{L}', \mathcal{C}_L, \mathcal{R}(\cdot) \rangle$, where $\mathcal{L}' = \{1, \ldots, L'\}$ represents an $L'$-level cardinal rating metric such that $2 \leq L' \leq L$. And $\mathcal{C}_L = \{C_1, \ldots, C_L\}$ denotes a set of potential contribution levels. The notation $\mathcal{R}(\cdot)$ represents a rating function which maps any given contribution $C_i \in \mathcal{C}_L$ to a specific rating $j \in \mathcal{L}'$, or mathematically $\mathcal{R}(\cdot) : \mathcal{C}_L \to \mathcal{L}'$. The rating function $\mathcal{R}(\cdot)$ maps the highest contribution $C_L$ to the highest rating $L'$, and maps the second highest contribution $C_{L-1}$ to the second highest rating $L' - 1$. This process continues until the threshold contribution level $L - L' + 1$ is reached, which is mapped to the lowest rating 1, and all the remaining levels of contribution are mapped to rating 1. We formally express $\mathcal{R}(\cdot)$ as

$$\mathcal{R}(C_k) = \begin{cases} k - L + L', & k > L - L' + 1 \\ 1, & k \leq L - L' + 1 \end{cases},$$

where $C_{L-L'+1}$ is the threshold contribution.

One can observe that the rating system introduced in Section 2 is a special case of $L' = L$. One can vary the value of $L'$ to obtain a rating system with different complexity, i.e., the number of rating points. We next quantify the impact of *threshold based rating systems* on the incentive mechanism.

## 3.2 Derivation of the Critical Value

We seek to quantify the impact of *threshold based rating systems* on the critical value. We extend the $n$-player game in Section 2 to accommodates the *threshold based rating system*, i.e., rewrite the reward function derived in Eq. (2), as

$$R_j(s_j, s_{-j}|r) = \begin{cases} \dfrac{nr}{\sum_\kappa \mathbf{I}_{\{\mathcal{R}(s_\kappa) = \mathcal{R}(s_j)\}}}, \text{if } \mathcal{R}(s_j) = \max\limits_\kappa \mathcal{R}(s_\kappa) \\ 0, \quad \text{otherwise} \end{cases}.$$

To illustrate, consider an example of 2-*bundling scheme*, three levels of contribution $L = 3$, and two rating points $L' = 2$. We show the corresponding utility in Table 3. One can observe that the strategy profile $(C_3, C_3)$ is a unique *Nash equilibrium* if and only if $r > c_3 - c_1 = c_3$. We generalize this result in the following theorem.

| | | $\boldsymbol{w}_2$: $C_3$ | $C_2$ | $C_1$ |
|---|---|---|---|---|
| $\boldsymbol{w}_1$ | $C_3$ | $r - c_3, r - c_3$ | $2r - c_3, -c_2$ | $2r - c_3, -c_1$ |
| | $C_2$ | $-c_2, 2r - c_3$ | $r - c_2, r - c_2$ | $r - c_2, r - c_1$ |
| | $C_1$ | $-c_1, 2r - c_3$ | $r - c_1, r - c_2$ | $r - c_1, r - c_1$ |

**Table 3: Utility matrix under TBR.**

**Theorem** 3.1. *Consider our proposed incentive mechanism, an $n$–bundling scheme and a threshold based rating system $\langle \mathcal{L}', \mathcal{C}_L, \mathcal{R}(\cdot) \rangle$. The strategy profile $(C_L, \ldots, C_L)$ is a unique Nash Equilibrium if and only if $r > \underline{r} = c_L$.*

**Proof:** This proof is similar to that of Lemma 2.1. ■

**Remark.** It states that the critical value is invariant of the number of rating points $L'$. This implies that the simplest rating system, i.e., $L' = 2$, is also an optimal system, where requesters only need to provide binary feedbacks to indicate whether they are satisfied or not with a solution.

# 4. CONCLUSION AND DISCUSSION

This paper studies incentive mechanism and rating system design for crowdsourcing applications. We develop a game-theoretic model to characterize workers' strategic behavior, which allows $L \geq 2$ levels of contribution. We design a class of simple but effective incentive mechanisms, which consist of a *task bundling scheme* and a *rating system*, and pay workers according to solution ratings from requesters. We develop a model to characterize the design space of a class of commonly users rating systems–*threshold based rating system*. We quantify the impact of such rating systems, and the bundling scheme on reducing requesters' reward payment. We find out that the simplest rating system, e.g., two rating points, is also an optimal system, where requesters only need to provide binary feedbacks to indicate whether they are satisfied or not with a solution.

We believe that our work has number of future directions, e.g., explore how human factors like preferences or biases in solution rating may influence the design of the incentive and rating system, or explore workers with different skill sets and how it may influence the crowdsourcing systems.

# 5. REFERENCES


[1] Amazon Mechanical Turk. `https://www.mturk.com`.
[2] R. Gibbons. *Game theory for applied economists*. Princeton University Press, 1992.
[3] Helpouts. `https://helpouts.google.com/home`.
[4] C.-J. Ho and J. W. Vaughan. Online task assignment in crowdsourcing markets. In *Proc. of AAAI*, 2012.
[5] J. Howe. Crowdsourcing: Why the power of the crowd is driving the future of business. *Crown Business*, 2008.
[6] E. I. Sparling and S. Sen. Rating: How difficult is it? In *Proc. of ACM RecSys*, 2011.
[7] L. von Ahn, R. Liu, and M. Blum. Peekaboom: A game for locating objects in images. In *Proc. of ACM CHI*, 2006.
[8] Yahoo! Answers. `http://answers.yahoo.com`.