Incentive Mechanism and Rating System Design for Crowdsourcing Systems: Analysis, Tradeoffs and Inference

Hong Xie, *Member, IEEE* and John C. S. Lui, *Fellow, IEEE*

Abstract—Macrotasking crowdsourcing systems like Elance and Fiverr serve as efficient platforms for requesters to outsource challenging and innovative tasks that require special skills to workers. It is widely practiced in such systems that requesters reward workers based on requesters' assessment on solution quality. The challenge is that requesters' assessment may not be accurate to reflect the intrinsic quality of a solution due to human factors like personal preferences or biases. In this work, we consider answering the following questions: *How to design a mechanism to incentivize workers provide high quality solutions in the presence of such human factors? How to formally study the impact of human factors on workers' financial incentive to participate?* We design a mechanism to incentivize workers to provide high-quality contributions, which is robust to human factors. Our incentive mechanism consists of a "*task bundling scheme*" and a "*rating system*", which reward workers based on requesters' rating on the solution quality. We propose a probabilistic model to capture human factors, and quantify their impact on the incentive mechanism. We formulate an optimization framework to select appropriate rating system parameters, which can be viewed as a tradeoff between "*system efficiency*", i.e., the total number of tasks can be solved given a fixed reward budget, and the "*rating system complexity*", which determines the human cognitive cost and time in expressing ratings. We also formulate an optimization framework to select appropriate bundling size, which can tradeoff system efficiency against service delay (i.e., the waiting time to form a task bundle). Finally, we conduct experiments on a dataset from Elance. Experimental results show that our incentive mechanism achieves at least 99.95 percent of the theoretical maximum system efficiency with a service delay of at most 2.3639 hours. Furthermore, we discover that the rating system in Elance is too complex, and it should be simplified to a binary rating system (i.e., two rating points).

Index Terms—Crowdsourcing, incentive mechanism, rating system, tradeoffs

1 Introduction

Over the past decade, we have witnessed the rise and success of online crowdsourcing services [2], [3]. Many well-known Internet companies provide crowdsourcing services, e.g., Amazon Mechanical Turk [4], Elance [5] and Yahoo!Answers [6]. On the business production side, crowdsourcing has been used as an efficient and cost-effective paradigm to produce products [7], design products [8], etc., by eliciting collective intelligence of a crowd. Besides, crowdsourcing systems like Amazon Mechanical Turk, have emerged as a novel platform to conduct experimentation for behavior research [9], social science [10], etc., where a large amount of human behavioral data can be collected efficiently, which in turn advances social science, behavioral economics, etc. Furthermore, crowdsourcing has evolved as an efficient paradigm to shed new light on many challenging problems like anomaly detection [11], reputation management [12] and network monitoring [13].

Based on the types of tasks, crowdsourcing systems can be broadly classified into two types: *microtasking* [14] and *macrotasking* [15]. Microtasking crowdsourcing systems focus on small and repetitive tasks that are simple for individuals to complete, e.g., image labeling, transcription, etc. Usually tasks do not require special skills and the reward for each task is usually small. Real-world microtasking crowdsourcing systems include Amazon Mechanical Turk [4], and Microtask [16]. Different from microtasking, macrotasking crowdsourcing systems are mainly used to solve challenging and innovative tasks, e.g., develop a computer program, which require special skills. Tasks usually are large in the sense that they require a large amount of time to complete and the reward is large, e.g., hundreds or thousands US dollars. Elance [5] and Fiverr [17] are two real-world macrotasking crowdsourcing systems.

We focus on macrotasking crowdsourcing in this paper. In particular, we conduct the first unified study on incentive and rating system design for such crowdsourcing systems. The underlying connection between incentive mechanism design and rating system design is the human factors like personal preferences or biases in assessing product quality. It is widely deployed in macrotasking crowdsourcing systems (e.g., Elance [5] and Fiverr [17]) that requesters reward workers based on the assessed solution quality. For example, requester will only give a large reward to a worker if the solution quality is high while give a small reward (or even no reward) if the solution is of low quality. However, requesters' assessment may not be accurate to judge the intrinsic quality of a solution due to human factors like

- *H. Xie is with the School of Computing, National University of Singapore, Singapore. E-mail: hongx87@gmail.com.*
- *J. C. S. Lui is with the Department of Computer Science and Engineering, The Chinese University of Hong Kong, Hong Kong. E-mail: cslui@cse.cuhk.edu.hk.*

Manuscript received 20 Aug. 2015; revised 22 Feb. 2016; accepted 3 Mar. 2016. Date of publication 9 Mar. 2016; date of current version 2 Feb. 2018.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TSC.2016.2539954

personal preferences or biases. Such human factors may result in that a worker providing a high-quality solution may receive a small or even no reward at all, which impair workers' financial incentive to participate. In this work, we address the following questions: *How to design a mechanism to incentivize workers provide high quality solutions in the presence of such human factors? How to formally study the impact of human factors on workers' financial incentive to participate?* To answer these two questions, we design an incentive mechanism and couple it with a rating system, which captures human factors. Our contributions are:

- We conduct the first unified study on incentive and rating system design for crowdsourcing systems. We design a simple but effective incentive mechanism, which consists of a *task bundling scheme*and a *rating system*. A task bundle means that several tasks operate as a group in reward splitting. We propose a probabilistic model to capture human factors such as biases in rating and quantifying their impact on the incentive mechanism.
- We show that increasing the rating system complexity (i.e., the number of rating points) can decrease the reward that a requester must pay to incentivize high-quality contributions from workers. However, it increases the human cognitive cost and time needed in expressing ratings. We formulate an optimization framework to select appropriate number of rating points.
- We show that selecting the bundling size for our incentive mechanism is a tradeoff between "*system efficiency*" (i.e., the total number of tasks can be solved given a fixed reward budget) and "*service delay*" (i.e., the waiting time to form a bundle). We formulate an optimization framework to select the appropriate bundle size.
- We conduct experiments on a real-life dataset from Elance. We formulate an optimization framework to infer model parameters from the data. We show that our incentive mechanism achieves at least 99.95 percent of the theoretical maximum system efficiency with service delay of at most 2.3639 hours. We discover that the rating system of Elance is in fact too complex, and it can be simplified to a binary rating system (i.e., two rating points).

This paper organizes as follows. Section 2 presents the system model. Section 3 presents our incentive mechanism. Section 4 studies the impact of human factors. Section 5 characterizes the rating systems design space. Section 6 studies tradeoffs in incentive and rating system design. Section 7 presents experimental results on a dataset from Elance. Section 8 presents the proof for lemmas and theorems. Related work is given in Section 9 and Section 10 concludes.

2 System Model

Consider a crowdsourcing system which categorizes tasks into K types. For example, "*Yahoo! Answers*" contains 25 types of tasks ranging from "*Health*" to "*Travel*" [18]. Users of a crowdsourcing system are classified into *requesters* and *workers*. A user can be a requester, or a worker, or in some cases, a requester/worker at the same time. Requesters outsource tasks to a crowdsourcing system 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, and it can be generally applied to all task types. We thus drop the subscript.

A task requires *only one* solution. We capture the scenario that a task requires multiple solutions by constructing replicas for a task such that each replica requires one solution. We assume that tasks are solved by workers having sufficient skills. In fact this can be achieved by deploying some task assigning algorithms [19], or deploying some reputation mechanisms [20].

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 greater 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 implies the worker exerts the ith 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. If $V_\kappa < r + T, \forall \kappa < L$, means that level κ contribution is not incentive-compatible. This paper aims to incentivize workers to provide their greatest contribution (C_L) and study how a rating system can influence workers' financial incentive to participate.

3 Incentive Mechanism

We present the design of our incentive mechanism, and we apply game theoretical technique to derive the minimum reward to incentivize the greatest contribution.

3.1 Incentive Mechanism Design

Our incentive mechanism consists of a *bundling scheme*, and a *rating system*. Tasks are completed via transactions under a *task bundling scheme*. When posting a task, a requester submits its reward r and service charge T to the *administrator*. The administrator bundles $n \geq 1$ tasks. 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 corresponding requesters. Requesters provide ratings on solution quality to the administrator. In particular, a rating i indicates that a worker provides level i contribution. Note that solutions are independent, and a requester can only express ratings to her own 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 crowdsourcing administrator divides the total 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*. One can observe that the administrator will not withhold the reward. A requester under our bundling scheme will not benefit by intentionally providing false ratings (since the reward was given to the administrator when she submits the task, and the reward will not be returned to the requester).

One benefit for requesters to provide feedback ratings is that workers will be incentivized to provide their maximum contribution. Another benefit is that requesters need to pay a smaller reward, if they can provide accurate feedback ratings. It is reasonable for requesters to provide feedback ratings. In fact, it is very common in real-world macrotasking crowdsourcing systems that requester proving feedback ratings on solution quality, e.g., Elance [5] Fiverr [17]. Such systems usually have the practice that requesters give rewards to workers based on the assessed solution quality. After assessing the product quality, it is natural to require a requester to provide feedback ratings, since it only incurs a quite small cost.

The physical meaning of a correct value for requesters' rating is to model the scenario that the requester to identify the true quality of a solution. However, in real-world crowdsorucing systems, requesters may not be able to identify the true quality due to reasons such as personal biases or preferences. For example, a critical requester may assign smaller ratings while a lenient requester may assign higher ratings [21], [22]. We incorporate such human factors in Section 4.

We like to emphasize that we have two justifications in using a competition-based approach for this work. The first one is that the competition-based approach is a natural choice to incentivize workers to provide their maximum contribution, which is crucial for macrotasking crowdsourcing systems. The second one is that with the competition-based approach, we can draw a clear and explicit connection between the incentive and rating system. The rating system enables us to explore the impact of human factors (like personal biases in evaluating solution quality) on workers' financial incentive to participate (Section 4). The challenge is how to incentivize participating workers provide their greatest contribution (C_L).

3.2 Formulating the n-Player Game

Consider an n-bundling scheme, we formulate an n-player game to capture the strategic behavior of workers who participate in the same task bundle. Specifically, 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 $\{C_1, \ldots, C_L\}$. We use the notation s_j to represent the strategic action of worker w_j. Let $s_{-j} = [s_\kappa]_{\kappa \neq j}$ denote 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 strategy profile (s_j, s_{-j}), which is defined as the reward minus cost. Formally, the utility of player w_j can be expressed as:

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

where $R_j(s_j, s_{-j}|r)$ is the reward under strategy profile (s_j, s_{-j}). We express $R_j(s_j, s_{-j}|r)$ as

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

To illustrate, consider an example of three levels of contribution $L = 3$ and a two-bundling scheme, say bundles two tasks. We then have $R_1(C_3, C_3|r) = R_1(C_2, C_2|r) = r$, $R_1(C_3, C_2|r) = 2r$, and $R_1(C_2, C_3|r) = 0$. Furthermore, $u_1(C_3, C_3|r) = r - c_3$, and $u_1(C_2, C_3|r) = -c_2$. Suppose $c_2 < r < c_3$. If worker w_1 provides C_1 contribution, then worker w_2 can maximize his utility by providing C_2 contribution. If worker w_1 provides C_3 contribution, then worker w_2 can maximize his utility by providing C_1 contribution.

Remark. 1) The above game is a complete information game. More concretely, each participating worker has sufficient skills to complete a task. In practice, one can achieve this by deploying a reputation mechanism to guarantee that only high skilled workers will commit to solve a task and low skilled workers will refuse a task [20]. 2) We assume that workers have identical marginal cost mainly for the ease of presentation. One can easily see that our framework applies to the case that workers have different cost. 3) We focus on the case that the tasks within a bundle are of similar type and degree of difficulty. In practice, this condition can be easily achieved by carefully selecting tasks of similar types and similar difficulty to form a bundle, since there are usually a large quantity of tasks in the pool.

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

Definition 3.1. *The desired Nash equilibrium $(C_L, \ldots, C_L)$ is a strategy profile that all workers provide C_L contribution.*

We present a formal way to show the uniqueness of a Nash equilibrium. We present strictly dominated strategy, which a player never plays.

Definition 3.2. *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*

$$u_i(s'_i, s_{-i}|r) > u_i(s_i, s_{-i}|r), \text{for all} s_{-i} \in \mathcal{C}_L^{n-1}.$$

Lemma 3.1 (Uniqueness [23]). *Consider a pure Nash equilibrium $(s_1^*, \ldots, s_n^*)$ for the 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.*

3.3 Deriving the Critical Value

We derive the minimum reward to guarantee workers provide their greatest contribution. We show that for the one-*bundling scheme*, it is impossible to achieve this. However, this can be achieved if we increase the bundle size to $n \geq 2$, without increasing the amount of desired reward (Theorem 3.1).

One-bundling scheme. Consider the *one-bundling scheme*, one can observe that the *dominant strategy* for the worker is C_1. In other words, 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.

TABLE 1
Utility Matrix Under the *Two-Bundling Scheme*

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

Two-bundling scheme. We show that under this bundling scheme, we can incentivize the greatest contribution 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 1, 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 for L levels of contribution in the following lemma.

Definition 3.3. *We define the "critical value" $\underline{r}$ as the minimum reward to incentivize the greatest contribution C_L.*

Lemma 3.2 (Two-bundling). *Consider a two-*bundling scheme *and L levels of contribution. The strategy profile (C_L, C_L) is a unique Nash Equilibrium if and only if $r > \underline{r} = c_L$.*

Proof. Please refer to Section 8 for derivation. □

Remark. Since (C_L, C_L) is a unique Nash equilibrium, workers will provide the greatest contribution C_L provided that they *do not collude*. If collusion is allowed, the best strategy for them is (C_1, C_1) so that everyone freerides. One way to eliminate this undesirable result is by bundling *more* tasks so to guarantee that at least one worker will not collude. However, we prove that increasing the bundle size does not increase the cost for requesters, in the following theorem.

Theorem 3.1. *Consider a n-*bundling scheme *with $n \geq 2$ and L levels of contribution. 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 3.2. □

Summary. All results thus far assume that requesters can perfectly express correct ratings to indicate workers' contribution level. In the next section, we analyze how human factors like preferences or biases may influence the design of the incentive system.

4 Human Factors in Solution Rating

We present a model to capture human factors in solution rating, and quantify their impact on the critical value. The complexity in computing the critical value is $\Theta(nL^2)$.

4.1 Model for Human Factors

We now extend the above model to a realistic scenario by accommodating various important human factors in rating such as *bias*, *preference*, *leniency*, etc [21], [22]. Specifically, a critical requester may assign smaller ratings while a lenient requester may assign higher ratings. They result in that a low rating on a high quality contribution, or a high rating on a low quality contribution. We call such ratings *"erroneous ratings"*.

We present a probabilistic model to capture the above human factors in solution ratings. To illustrate, consider a worker who exerted C_L, or a high quality contribution. Due to human factors, a worker may receive a rating ranging from 1 to L. We use the notation $\alpha_{L,j} \in [0, 1]$ to represent the probability that this worker receives a rating j, i.e., the requester evaluates his solution as a C_j contribution. Mathematically, we have $\alpha_{L,j} = \Pr[\text{evaluated as } C_j \text{ contribution} | C_L \text{ contribution}]$, where $\sum_{j=1}^{L} \alpha_{L,j} = 1$. Similarly, we define $\alpha_{i,j}$ as $\alpha_{i,j} = \Pr[\text{evaluated as } C_j \text{ contribution} \mid C_i \text{ contribution}]$. When $\alpha_{i,i} = 1$ for $i \in \{1, ..., L\}$, this implies requesters have perfect evaluation on the contribution by workers. Otherwise, $\alpha_{i,j}$ can represent different degrees of variability in evaluation. Note that all $\alpha_{i,j}, \forall i, j$ composes a matrix, which presents the probability of all possible rating outcome. We use $\boldsymbol{\alpha}$ to represent this matrix. We call this the *confusion matrix*.

We state two natural properties that a confusion matrix $\boldsymbol{\alpha}$ should satisfy. First, consider a C_L solution. Intuitively, when contribution C_i is higher than C_j (or $C_i \succ C_j$), the probability that a requester misjudges this solution as C_i should be larger than misjudging it as a C_j one, or formally $\alpha_{L,i} > \alpha_{L,j}$. Generalizing this statement, we obtain the first natural property, namely, row diagonally dominated and row singly peaked.

Definition 4.1 (Row/Column Diagonally Dominated). *We say $\boldsymbol{\alpha}$ is a row/column diagonally dominated matrix, if each row/column has a maximum entry at its diagonal entry.*

Definition 4.2 (Row/Column Singly Peaked). *We say $\boldsymbol{\alpha}$ is row/column singly peaked, if the entry of each row/column strictly increasing/decreasing prior/after the diagonal entry.*

Second, the probability that a requester misjudges a C_i solution as a C_L one should be larger than misjudging a C_j solution as a C_L one. Generalizing this statement, we obtain another property: column diagonally dominated and column singly peaked.

Proposition 4.1. *The confusion matrix $\boldsymbol{\alpha}$ has the properties: row/column diagonally dominated, and singly peaked.*

4.2 Deriving the Critical Value

Let us now derive the critical value under erroneous ratings due to human factors. In the presence of erroneous ratings (or the *confusion matrix $\boldsymbol{\alpha}$*), we have the question: is it possible to sustain a Nash equilibrium where worker provide their greatest contribution? Observe that with the *confusion matrix $\boldsymbol{\alpha}$* the utility becomes a random variable, i.e., the same strategy profile may result in different utilities with certain probabilities. Our approach is to characterize workers' strategic behavior via the n-player game under the *expected utility*, i.e., $E[u_j(C_\kappa, s_{-j}|r)]$. In particular, we express the necessary and sufficient condition under which $(C_L, \ldots, C_L)$ remains a Nash equilibrium as:

$$E[u_j(C_L, s_{-j}|r)] \geq E[u_j(C_\kappa, s_{-j}|r)], \forall \kappa, \tag{3}$$

holds for all j, where $s_{-j} = (C_L, \ldots, C_L)$. We prove the existence and and uniqueness of the desired Nash equilibrium $(C_L, \ldots, C_L)$ in the following theorem.

TABLE 2
Numerical Examples for the Critical Value $\underline{r}$

n	2	3	4
$\underline{r}$ (no variability)	c_3	c_3	c_3
$\underline{r}$ (variability)	$1.256c_3$	$1.128c_3$	$1.106c_3$

Theorem 4.1. *Consider a n-bundling scheme with $n \geq 2$, L levels of contribution and a confusion matrix $\boldsymbol{\alpha}$. The strategy profile $(C_L, \ldots, C_L)$ is a unique Nash equilibrium iff*

$$r > \underline{r} = \max_{\kappa}\left\{\frac{c_L - c_\kappa}{1 - \sum_{l=1}^{L}\sum_{\ell=0}^{n-1}\binom{n}{\ell+1}\left(\sum_{k=1}^{l-1}\alpha_{L,k}\right)^{n-\ell-1}\alpha_{\kappa,l}\alpha_{L,l}^{\ell}}\right\}.$$

Remark. It implies that one can incentivize workers to provide the greatest contribution by setting a proper reward even under the erroneous rating setting. Besides, the complexity in computing the desired reward is $\Theta(nL^2)$. One potential attack is that a requester intentionally provides some adversarial ratings regardless of the solution quality. For example, he can always provide the lowest rating 1. However, requesters do not benefit by providing adversarial ratings, since a requester cannot get the reward back no matter what ratings he gives.

We show some illustrating numerical examples for the critical value in Table 2, where we consider three levels of contribution $L = 3$ and a *confusion matrix*

$$\boldsymbol{\alpha} = \begin{bmatrix} 0.719 & 0.216 & 0.065 \\ 0.188 & 0.625 & 0.187 \\ 0.065 & 0.216 & 0.719 \end{bmatrix}.$$

One can observe that when there is no erroneous ratings (or no variability), as we vary the bundle size from two to four, the critical value remains at c_3. While in the presence of erroneous ratings (or variability), the critical value decreases from $1.256c_3$ to $1.106c_3$. This implies that requesters need to pay more, due to variability. Also, as we increase bundle size, we decrease the critical value.

Summary. Our model thus far considers a rating system with a small number of contribution level L. 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 [24]. How to design a proper rating system to address this challenge? How different designs of rating system may influence the incentive mechanism?

5 Modeling Rating Systems

We present a model to characterize the design space rating systems, and we quantify their impact on the critical value. We show that when there is no erroneous ratings, a binary rating system, i.e., two rating points indicating satisfied or not, is optimal in terms of critical value. While in the presence of erroneous ratings, an increase in the number of rating points leads to a drop on the critical value.

5.1 Threshold Based Rating Systems

Many crowdsourcing services adopt *threshold based rating systems*, 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. Our objective is to quantify its impact on the requesters' overhead, as well as illustrate how to model and analyze a rating system.

TABLE 3
Examples of the Threshold Based Rating System

	$\widetilde{L}$	C_1	C_2	C_3	C_4
$\mathcal{R}(\cdot)$	4	1	2	3	4
$\mathcal{R}(\cdot)$	3	1	1	2	3
$\mathcal{R}(\cdot)$	2	1	1	1	2

A *threshold based rating system* is a triplet $\langle \widetilde{\mathcal{L}}, \mathcal{C}_L, \mathcal{R}(\cdot)\rangle$, where $\widetilde{\mathcal{L}} = \{1, \ldots, \widetilde{L}\}$ represents an $\widetilde{L}$-level cardinal rating metric such that $2 \leq \widetilde{L} \leq L$. 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 $j \in \widetilde{\mathcal{L}}$, or mathematically $\mathcal{R}(\cdot) : \mathcal{C}_L \rightarrow \widetilde{\mathcal{L}}$. The rating function $\mathcal{R}(\cdot)$ maps the greatest contribution C_L to the highest rating $\widetilde{L}$, and maps the second greatest contribution C_{L-1} to the second highest rating $\widetilde{L} - 1$. This process continues until the threshold contribution level $L - \widetilde{L} + 1$ 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 this rating function $\mathcal{R}(\cdot)$ as

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

where $C_{L-\widetilde{L}+1}$ is the threshold contribution, or the minimum requirement on solutions. We show some illustrating examples in Table 3, where we consider four levels of contribution $L = 4$, and we vary the number of rating points $\widetilde{L}$ from 2 to 4. We show the corresponding rating function $\mathcal{R}(\cdot)$. One can see that when $\widetilde{L} = 2$, we have $\mathcal{R}(C_1) = \mathcal{R}(C_2) = \mathcal{R}(C_3) = 1$, and $\mathcal{R}(C_4) = 2$.

One can observe that the rating system introduced in Section 2 is a special case of $\widetilde{L} = L$. When $\widetilde{L} = 2$, we obtain a binary rating system. One can vary the value of $\widetilde{L}$ 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.

5.2 Deriving the Critical Value

We seek to quantify the impact of *threshold based rating systems* on the critical value. We explore the setting without/with erroneous ratings respectively.

We explore the setting without erroneous ratings. We extend the n-player game specified in Section 2 to accommodates the *threshold based rating system*. We rewrite the reward function derived in Equation (2), as

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

We next derive the critical value when there are no erroneous ratings.

Lemma 5.1. *Consider the setting without erroneous ratings, a* $n-$*task bundling scheme and a threshold based rating system* $\langle\widetilde{\mathcal{L}},\mathcal{C}_L,\mathcal{R}(\cdot)\rangle$. *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 3.1. □

Remark. Under the perfect scenario without erroneous ratings, the critical value is invariant of the number of rating points $\widetilde{L}$. This implies that the simplest rating system, e.g., $\widetilde{L}=2$, is also an optimal system, where requesters only need to provide binary feedbacks to indicate whether they are satisfied or not with a solution.

We now explore the scenario with erroneous ratings. In the presence of erroneous ratings, the rating process becomes probabilistic, which is governed by the *confusion matrix* $\boldsymbol{\alpha}$ and the the *threshold based rating system*$\langle\widetilde{\mathcal{L}},\mathcal{C}_L,\mathcal{R}(\cdot)\rangle$. Specifically, let $\Pr[k'|C_k,\langle\widetilde{\mathcal{L}},\mathcal{C}_L,\mathcal{R}(\cdot)\rangle]$ denote the probability that a C_k solution receives a rating k'. We formally describe the rating process as

$$\Pr[k'|C_k,\langle\widetilde{\mathcal{L}},\mathcal{C}_L,\mathcal{R}(\cdot)\rangle] = \begin{cases} \alpha_{k,L-\widetilde{L}+k'}, k'=2,\ldots,\widetilde{L} \\ \sum_{j=1}^{L-\widetilde{L}+1}\alpha_{k,j}, k'=1. \end{cases} \tag{4}$$

To illustrate, consider a rating system $\langle\widetilde{\mathcal{L}},\mathcal{C}_L,\mathcal{R}(\cdot)\rangle$, with $\mathcal{C}_L=\{C_1,C_2,C_3\}$, and $\widetilde{\mathcal{L}}=\{1,2\}$. We have $\Pr[2|C_3,\langle\widetilde{\mathcal{L}},\mathcal{C}_L,\mathcal{R}(\cdot)\rangle]=\alpha_{3,3}$, and $\Pr[1|C_3,\langle\widetilde{\mathcal{L}},\mathcal{C}_L,\mathcal{R}(\cdot)\rangle]=\alpha_{3,1}+\alpha_{3,2}$.

We now explore whether workers will provide the greatest contributions, or whether $(C_L,\ldots,C_L)$ is still a *Nash equilibrium* strategy. Again, we characterize workers' strategic behavior via the n-player under expected utility. In the following theorem we state the key result of this work, which quantifies the impact of the threshold based rating system on the critical value.

Theorem 5.1. *Consider a* n*-bundling scheme, a confusion matrix* $\boldsymbol{\alpha}$, *and a* $\langle\widetilde{\mathcal{L}},\mathcal{C}_L,\mathcal{R}(\cdot)\rangle$ *rating system. The strategy profile* $(C_L,\ldots,C_L)$ *is a unique Nash equilibrium iff*

$$r > \underline{r} = \max_\kappa\left\{(c_L-c_\kappa)\Bigg/\left[1-\sum_{l=1}^{L-\widetilde{L}+1}\alpha_{\kappa,l}\left(\sum_{k=1}^{L-\widetilde{L}+1}\alpha_{L,k}\right)^{n-1}\right.\right.$$
$$\left.\left.-\sum_{l=2}^{\widetilde{L}}\sum_{\ell=0}^{n-1}\binom{n}{\ell+1}\left(\sum_{k=1}^{L-\widetilde{L}+l-1}\alpha_{L,k}\right)^{n-\ell-1}\alpha_{\kappa,L-\widetilde{L}+l}\alpha^{\ell}_{L,L-\widetilde{L}+l}\right]\right\}. \tag{5}$$

Proof. Please refer to Section 8 for derivation. □

Remark. The importance of the above theorem is on the *existence and uniqueness*of the desired Nash Equilibrium $(C_L,\ldots,C_L)$ under different design of rating systems. In addition, it quantifies the impact of the number of rating points $\widetilde{L}$ on the critical value. As we shall see later, this result serves as building blocks to explore rating system design tradeoffs.

TABLE 4
Impact of Number of Rating Points on the Critical Value $\underline{r}$, Where $L=7, n=2$

$\widetilde{L}$	θ	2	3	4	5	6	7
$\underline{r}/c_7$	0.1	1.11	1.01	1.00	1.00	1.00	1.00
$\underline{r}/c_7$	0.2	1.25	1.04	1.01	1.00	1.00	1.00
$\underline{r}/c_7$	0.3	1.43	1.10	1.03	1.01	1.01	1.00

We show some illustrating numerical examples on the critical value in Table 4, where we examine the impact of number of rating points. In Table 4, we specify the cost for each level of contribution as

$$c_j = (j-1)/(L-1), j=1,\ldots,L, \tag{6}$$

and we specify the *confusion matrix* $\boldsymbol{\alpha}$ as

$$\alpha_{j,\kappa} = \theta^{|j-\kappa|}\Big/\sum\nolimits_{\kappa=1}^{L}\theta^{|j-\kappa|}, \forall j,\kappa=1,\ldots,L, \tag{7}$$

where $\theta\in(0,1)$. Note that this choice of the cost function and confusion matrix are only for illustration purpose. Practically, one can infer them from data. The higher the value of θ, the higher the variability in ratings. The number of contribution levels is $L=7$. And we vary the value of the number of rating points $\widetilde{L}$ from 2 to 7. One can observe that as we increase the number rating points $\widetilde{L}$, we decrease the critical value. When we increase the value of θ, we increase the critical value. Namely, the higher variability in rating, the higher the reward requesters need to pay. It is interesting to observe that five rating points $\widetilde{L}=5$ is actually good enough, since the critical value is very close to c_7, and further increasing on the number of rating points, decreases the critical very less than 1 percent. This coincides with that five-level rating system is commonly used in crowdsourcing systems.

Summary. By using the *threshold based rating system*, it is possible incentivize the greatest contribution even in the presence of erroneous ratings. We also quantify the impact of number of rating points on the critical value. We will apply these results to explore rating system design tradeoffs in Section 6.

6 Design Tradeoffs

We formulate an optimization framework to select appropriate rating system parameters to tradeoff between *"system efficiency"*, i.e., the total number of tasks can be solved given a fixed reward budget, and the *"rating system complexity"*, which determines the human cognitive cost and time in expressing ratings. We also formulate an optimization framework to select appropriate bundling size, which can tradeoff system efficiency against service delay (i.e., the waiting time to form a task bundle).

6.1 Metrics

We first provide a metric to quantify the crowdsourcing system efficiency. We say a crowdsourcing system is efficient if a large number of high quality solutions can be solicited using a given reward budget. Lowering the critical value implies that more tasks can be solved with the same reward budget. Hence, it leads to an improvement on system efficiency and an increased level of user participation and engagement. We formally define system efficiency as follows.

Definition 6.1. *Let $\mathcal{E}$ denote crowdsourcing system efficiency, which is defined as the contribution reward ratio, i.e.,*

$$\mathcal{E} = \frac{c_L}{\underline{r}}.$$

The higher the value of $\mathcal{E}$, the higher the system efficiency. Note that $\underline{r} \geq c_L$, so we have $\mathcal{E} \in [0,1]$. As we have shown that increasing the number of rating points can reduce the critical value $\underline{r}$, i.e., improve system efficiency. From crowdsourcing owners' perspective, they seek to maximize the system's efficiency, i.e., maximizing $\mathcal{E}$, under the constraint that workers provide the greatest contributions, because low quality contributions may discourage requesters, which may finally result in revenue loss. Let $\overline{\mathcal{E}}$ denote the theoretical maximum system efficiency. We have $\overline{\mathcal{E}} = 1$.

Increasing the number of rating points can improve system efficiency. However, it comes at the price of increasing the cost in expressing ratings. For example, increasing the number of rating points, may lead to more erroneous rating, e.g., a higher degree of variability, or increase the cognitive load in expressing ratings [24].

Definition 6.2. *Let $\mathcal{C}_\gamma(\cdot) : \{1,\ldots,L\} \to [0,1]$ denote a decreasing function which prescribes a cognitive cost $\mathcal{C}_\gamma(i)$ in expressing a rating when the rating system has i rating points, where $i = 1,\ldots,L$.*

Note that $\mathcal{C}_\gamma(1) = 0$. This models the scenario that when the number of rating point is one, requester does not need to spend any cognitive effort in expressing ratings. Note that learning or determining the specific form of $\mathcal{C}_\gamma(\cdot)$ is not the focus of this paper. Our objective is to design an efficient framework which is general enough to work on any instances of $\mathcal{C}_\gamma(\cdot)$. Once a system designer determines $\mathcal{C}_\gamma(\cdot)$, our framework can be applied to explore different design tradeoffs.

Increasing the bundle size can improve system efficiency, i.e., decrease critical value. However, the side effect is in increasing the delay of having a task completed. This is because that as we increase bundle size, it will take a longer time to collect all n tasks to form a bundle. Formally, we consider that tasks arrive with an average rate λ. We emphasize that our framework is quite general in the sense that it does not assume any task arrival pattern (i.e., specific arrival processes like Poisson).

Definition 6.3. *Let $\mathcal{D}$ denote service delay, which is the expected waiting time to form a bundle. We have $\mathcal{D} = n/\lambda$.*

The smaller the bundle size, the shorter the service delay. For example, consider $\lambda = 24$ tasks/day. If $n = 2$, then the expected delay is $\mathcal{D} = 1/12$ day. If $n = 24$, then the expected delay is $\mathcal{D} = 1$ day. Crowdsourcing system operators are interested to minimize the delay under the constraint that workers provide the greatest contribution.

6.2 Optimal Threshold Based Rating System Design

We formulate our design objective as jointly minimizing the rating cost $\mathcal{C}_\gamma(\widetilde{L})$ and the efficiency loss $1 - \mathcal{E}$, i.e.,

$$(1-\mu)\mathcal{C}_\gamma(\widetilde{L}) + \mu(1-\mathcal{E}) = (1-\mu)\mathcal{C}_\gamma(\widetilde{L}) + \mu\left(1 - \frac{c_L}{\underline{r}}\right), \quad (8)$$

where $\mu \in [0,1]$, subject to the constraint that workers provide the greatest contribution. One can vary the weight μ to attain different design tradeoffs. For example, $\mu = 1$ means that a designer only wants to minimize the efficiency loss. While $\mu = 0$ means that the designer only wants to minimize the rating cost.

Consider the *threshold based rating system*, we seek to find the optimal number of rating points $\widetilde{L}^*$ that minimize Equation (8). We can formulate the following optimization problem:

$$\begin{aligned} &\min_{\widetilde{L}} (1-\mu)\mathcal{C}_\gamma(\widetilde{L}) + \mu\left(1 - \frac{c_L}{\underline{r}}\right) \\ s.t. \quad & 2 \leq \widetilde{L} \leq L, \\ & \underline{r} \text{ satisfies Equation } (5). \end{aligned}$$

The second constraint derived in Theorem 5.1 is a necessary condition such that workers provide the greatest contribution. We can locate the optimal solution via exhaustive search, i.e., enumerating the number of rating points from 2 to L. The complexity is simply $\Theta(L^3)$. In practice, the value of L is at most hundreds. Hence, it will not be computationally expensive.

6.3 Optimal Bundling Size

We formulate our design objective as jointly minimizing system efficiency loss $1 - \mathcal{E}$ and service delay $\mathcal{D}$, i.e.,

$$\widetilde{\mu}(1-\mathcal{E}) + (1-\widetilde{\mu})\mathcal{D} = \widetilde{\mu}\left(1 - \frac{c_L}{\underline{r}}\right) + (1-\widetilde{\mu})\frac{n}{\lambda},$$

where $\widetilde{\mu} \in [0,1]$, subject to that workers provide the greatest contribution. The optimization formulation is:

$$\begin{aligned} &\min_{n} \widetilde{\mu}\left(1 - \frac{c_L}{\underline{r}}\right) + (1-\widetilde{\mu})\frac{n}{\lambda} \\ s.t. \quad & n \geq 2, n \in \mathbb{N}, \\ & \underline{r} \text{ satisfies Equation } (5). \end{aligned}$$

The second constraint derived in Theorem 5.1 is a necessary condition such that workers provide the greatest contribution. One can vary the weight $\widetilde{\mu}$ to attain different design tradeoffs. For example, $\widetilde{\mu} = 1$ means that a crowdsourcing system operator only wants to minimize system efficiency loss. While $\widetilde{\mu} = 0$ means that a crowdsourcing system operator only wants to minimize service delay.

The remaining question is how to locate the optimal bundling size for the above optimization problem. Let n^* denote the optimal bundling size. We state an upper bound for n^* in the following lemma.

Lemma 6.1. *The optimal bundling size n^* satisfies*

$$n^* \leq \frac{\lambda\widetilde{\mu}}{1-\widetilde{\mu}} + 2. \quad (9)$$

Remark. This lemma implies that to locate the optimal bundling size, we only need to search all the values from 2 to $\lfloor \lambda\widetilde{\mu}/(1-\widetilde{\mu}) + 2 \rfloor = \lfloor \lambda/(1-\widetilde{\mu}) - \lambda + 2 \rfloor$. This means that the computational complexity is linear in λ and $1/(1-\widetilde{\mu})$.

7 Experiments on Real-World Data

We present experimental results on a real-life dataset from Elance. We show that our incentive mechanism can achieve at least 99.95 percent of the theoretical maximum efficiency with a service delay of at most 2.3639 hours. The rating system in Elance is too complex, and it should be simplified to a binary rating system.

7.1 Elance Dataset

In early May 2015, we launched a crawler to crawl historical transaction data from Elance [5]. Founded in 1999, Elance provides online crowdsourcing services. In Elance, workers provide solutions to various types of tasks, e.g., programming, translation, etc. Each worker posts the skills that they have and set a price to solve a task, e.g., a worker is skilled in Android programming and sets a price of fifty dollars per hour. Requesters can select workers to solve a task. When a task is completed, requesters can rate the quality of the service using one to five stars. More stars imply higher quality. Our dataset contains 156,982 task transactions conducted from April 2nd 2001 to May 16th 2015. It involves 9,918 workers. The number of ratings across each rating level are: 363 (one star), 620 (two stars), 1,504 (three stars), 13,780 (four stars),140,715 (five stars). We released the dataset in the link [25], which has three columns corresponding to the normalized task ID, the time stamp that a task was completed and the rating respectively.

7.2 Inferring Model Parameters

We first infer the transaction's arrival rate λ. In our dataset, each transaction has a time stamp. We infer the transaction's arrival rate λ as the total number of transactions divided by the total time to accumulate these transactions. From our data, we obtain $\lambda = 30.458 \text{ transactions/day}$.

Elance adopts a five star rating scale. We interpret it as that workers can provide one of five levels of contributions, i.e., $L = 5$. A worker sets a price for providing a service. We interpret this price as the cost of that worker to provide the greatest contribution (c_L). We consider a linear cost function model. Namely, the cost increases linearly with contribution level, or $c_j = \frac{j-1}{L-1} c_L$, where $j = 1, \dots, L$. We choose a linear cost function because in Elance's rating levels, it exhibits a linear property, i.e., {1 ("Terrible"), 2 ("Poor"), 3 ("Average"), 4 ("Good"), 5 ("Excellent") }.

Now we infer the confusion matrix $\boldsymbol{\alpha}$ from historical ratings. We point that it is impossible to infer the exact contribution level that results in a rating. We assume that workers in the past provided their maximum contribution c_L. This assumption is reasonable because in the Elance system, requesters can deny to pay workers if they do not accept any solutions. This means that new workers still need to work. The problem is that personal biases or preferences in assessing solution quality may result in that a requester does not reward a worker even if he provides the maximum contribution. Our work addresses this issue. We can then infer $\alpha_{5,j}, j = 1, \dots, 5$ as the fraction of ratings that equals to j. Apply this rule on our data, we obtain the 5th row of the confusion matrix $\boldsymbol{\alpha}$. Let $\boldsymbol{\alpha}^o$ denote the observed confusion matrix $\boldsymbol{\alpha}$. We have $\alpha^o_{5,1} = 0.0023$, $\alpha^o_{5,2} = 0.0039$, $\alpha^o_{5,3} = 0.0096$, $\alpha^o_{5,4} = 0.0878$, $\alpha^o_{5,5} = 0.8964$ and the other entries of $\boldsymbol{\alpha}^o$ are missing. One can observe that more than 89 percent of rating instances assigning a five star rating. This implies that most requesters are unbiased that they can identify the true quality of a solution. The remaining thing is how to infer the full confusion matrix $\boldsymbol{\alpha}$ from this partial observation $\boldsymbol{\alpha}^o$. We consider the case that $\boldsymbol{\alpha}$ is generated as follows:

$$\alpha_{i,j} = \theta^{|i-j|} \Big/ \sum\nolimits_{j=1}^{L} \theta^{|i-j|}, \forall i, j = 1, \dots, L, \tag{10}$$

where $\theta \in [0, 1]$. We have two reasons to consider this form: (1) this form of $\boldsymbol{\alpha}$ satisfies proposition 4.1; (2) we have only one parameter θ to infer, which enables us to infer $\boldsymbol{\alpha}$ from $\boldsymbol{\alpha}^o$. Our objective is to find θ such that $||\boldsymbol{\alpha} - \boldsymbol{\alpha}^o||_F^2$ is minimized, where $|| \cdot ||_F$ denotes Frobenius norm. Let $\boldsymbol{\alpha}_5$ and $\boldsymbol{\alpha}^o_5$ denote the fifth row of $\boldsymbol{\alpha}$ and $\boldsymbol{\alpha}^o$ respectively. Incorporating the missing entries of $\boldsymbol{\alpha}^o$ we refine our objective as $\min_\theta ||\boldsymbol{\alpha}_5 - \boldsymbol{\alpha}^o_5||^2$, where $|| \cdot ||$ denotes Euclidean norm. Applying Eq. (10) we have

$$\begin{aligned} &\min\nolimits_\theta ||\boldsymbol{\alpha}_5 - \boldsymbol{\alpha}^o_5||^2 = \left(\frac{1-\theta}{1-\theta^5} - 0.8964\right)^2 + \\ &\left(\frac{1-\theta}{1-\theta^5}\theta - 0.0878\right)^2 + \left(\frac{1-\theta}{1-\theta^5}\theta^2 - 0.0096\right)^2 + \\ &\left(\frac{1-\theta}{1-\theta^5}\theta^3 - 0.0039\right)^2 + \left(\frac{1-\theta}{1-\theta^5}\theta^4 - 0.0023\right)^2 \\ &s.t. \theta \in [0, 1]. \end{aligned}$$

The form of optimization problem has been well studied. One way to locate its solution is via gradient decent method [26]. The optimal solution θ^* is $\theta^* = 0.1012$. Substituting this value into Equation (10), we obtain $\boldsymbol{\alpha}$:

$$\boldsymbol{\alpha} = \begin{bmatrix} 0.8988 & 0.0910 & 0.0092 & 0.0009 & 0.00009 \\ 0.0834 & 0.8239 & 0.0834 & 0.0084 & 0.0009 \\ 0.0084 & 0.0828 & 0.8177 & 0.0828 & 0.0084 \\ 0.0009 & 0.0084 & 0.0834 & 0.8239 & 0.0834 \\ 0.00009 & 0.0009 & 0.0092 & 0.0910 & 0.8988 \end{bmatrix}.$$

It implies that requesters can identify the true quality of solutions with probability with at least 0.8 since the diagonal entries of $\boldsymbol{\alpha}$ is larger than 0.8. With probability around 0.1, requesters may over estimate (or under estimate) the product quality by one level.

7.3 Optimal Bundling Size and Reward

We use the above inferred model parameters and apply the optimization framework developed in Section 6 to demonstrate how to determine the bundling size and reward for Elance. Through this we show the efficiency and effectiveness of our incentive mechanism. Table 5 depicts the optimal bundling size n^*, the optimal critical value $\underline{r}^*$ and the corresponding system efficiency and service delay. We vary the tradeoff factor $\widetilde{\mu}$ from 0.1 to 0.999. Consider $\widetilde{\mu} = 0.1$, say a crowdsourcing system operator extremely cares about service delay. The optimal bundling size is $n^* = 2$, the corresponding optimal reward is $r^* = 1.00050 c_L$ (c_L is inferred in Section 7.2). It is interesting to observe that the system efficiency attains $\mathcal{E}^*/\overline{\mathcal{E}} = 0.99950$ of the theoretical maximum system efficiency.

TABLE 5
Optimal Bundling Size and Reward

$\widetilde{\mu}$	0.1	0.5	0.9	0.999
n^*	2	2	2	3
$\underline{r}^*/c_L$	1.00050	1.00050	1.00050	1.00012
$\mathcal{E}^*/\overline{\mathcal{E}}$	0.99950	0.99950	0.99950	0.99988
$\mathcal{D}^*$ (hours)	1.5759	1.5759	1.5759	2.3639

The service delay is $\mathcal{D}^* = 1.5759$ hours. Now consider $\widetilde{\mu} = 0.999$, say a crowdsourcing system operator extremely cares about system efficiency. We have $n^* = 3$ and $r^* = 1.00012c_L$. It is interesting to observe that the system efficiency attains $\mathcal{E}^*/\overline{\mathcal{E}} = 0.99988$ of the theoretical maximum system efficiency. The service delay is $\mathcal{D}^* = 2.3639$ hours. As $\widetilde{\mu}$ varies from 0.1 to 0.999, n^* increases from 2 to 3, system efficiency $\mathcal{E}^*$ improves from $0.99950\overline{\mathcal{E}}$ to $0.99988\overline{\mathcal{E}}$, and service delay increases from 1.5759 hours to 2.3639 hours. These results show that our incentive mechanism achieves a quite high system efficiency and a low service delay.

7.4 Optimal Rating System

We now explore whether Elance need to increase (or reduce) the complexity of its rating system?

We first show that there is no need for Elance to increase the complexity of its rating system. As presented in Table 5, the optimal bundling size for Elance can be 2 or 3. Here we set the bundling size to be $n = 3$. The critical value and system efficiency for Elance are

$$\underline{r} = 1.00012c_L, \mathcal{E}/\overline{\mathcal{E}} = 0.99988.$$

One can observe that Elance can improve its system efficiency by at most $1 - \mathcal{E}/\overline{\mathcal{E}} = 0.00012\%$ via increasing the number of rating points. This improvement is so tiny that there is no need for Elance to increase the number of rating points.

We show that the rating system of Elance is too complex, it should simplify its rating system to a binary rating system (i.e., two rating points). We vary the number of rating points $\widetilde{L}$ from 5 to 2 for Elance. For each of $\widetilde{L}$, we measure the drop ratio on system efficiency. Recall that when the number of rating points is 5, system efficiency is $\mathcal{E} = 0.99988$. We therefore quantify system efficiency drop ratio as

$$\Delta\mathcal{E} = (0.99988 - \mathcal{E})/0.99988$$

Table 6 presents numerical results on $\Delta\mathcal{E}$. We can observe that as the number of rating points drops from $\widetilde{L} = 5$ to $\widetilde{L} = 2$, system efficiency drops from $0.99988\overline{\mathcal{E}}$ to $0.98966\overline{\mathcal{E}}$. Even a binary rating system, i.e., $\widetilde{L} = 2$ can achieve $\mathcal{E}/\overline{\mathcal{E}} = 0.98966$ of the theoretical maximum system efficiency, i.e., a quite high system efficiency. Furthermore, reducing the number of rating points from $\widetilde{L} = 5$ to $\widetilde{L} = 2$ only increases system efficiency drop ratio from $\Delta\mathcal{E} = 0\,\text{percent}$ to $\Delta\mathcal{E} = 1.02\,\text{percent}$. In other word an around one percent drop on system efficiency. They implies that the rating system in Elance is quite too complex, and it should be simplified to a binary rating system (i.e., two rating points).

TABLE 6
Optimal Bundling Size and Reward ($n = 3$)

$\widetilde{L}$	5	4	3	2
$\underline{r}$	$1.00012c_L$	$1.00012c_L$	$1.00021c_L$	$1.01045c_L$
$\mathcal{E}/\overline{\mathcal{E}}$	0.99988	0.99988	0.99979	0.98966
$\Delta\mathcal{E}$	0%	0%	0.009%	1.02%

8 Proofs

In this section we present the proofs for the lemmas the theorems. We state a lemma which will be frequently used in our proof.

Lemma 8.1. *Let $x_1, \ldots, x_\ell$ denote ℓ real numbers such that $x_\ell > \ldots > x_1 > 0$. Let $\beta_1, \ldots, \beta_\ell$, and $\beta'_1, \ldots, \beta'_\ell$ denote two series of real numbers such that $\beta_i \geq 0, \beta'_i \geq 0, \forall i$ and $\sum_i \beta_i = \sum_i \beta'_i$. Suppose there exists an integer $1 \leq j < \ell$ such that for any $i > j$, it holds that $\beta_i > \beta'_i$, and for any $i < j$, it holds that $\beta_i < \beta'_i$. Then we have $\sum_i x_i\beta_i > \sum_i x_i\beta'_i$.*

Proof. Let $\epsilon_i = \beta_i - \beta'_i$. Observe that for any $i > j$, we have $\epsilon_i > 0$, and for any $i < j$, we have $\epsilon_i < 0$. Since $\sum_i \beta_i = \sum_i \beta'_i$, thus $\sum_i \epsilon_i = 0$. Suppose $\beta_j \leq \beta'_j$ Then it follows that $\sum_i x_i\beta_i - \sum_i x_i\beta'_i = \sum_i \epsilon_i x_i = \sum_{i=1}^{j} \epsilon_i x_i + \sum_{i=j+1}^{\ell} \epsilon_i x_i > \sum_{i=1}^{j} \epsilon_i x_j + \sum_{i=j+1}^{\ell} \epsilon_i x_{j+1} = \sum_{i=j+1}^{\ell} \epsilon_i (x_{j+1} - x_j) > 0$. We therefore conclude this lemma for $\beta_j \leq \beta'_j$. When $\beta_j > \beta'_j$, with a similar derivation we conclude this lemma. □

Proof of Lemma 3.2. Let us prove the necessary condition first. The strategy profile (C_L, C_L) being a unique Nash equilibrium implies that for each worker w_j, the condition $u_j(C_L, C_L|r, 2) > u_j(C_\kappa, C_L|r, 2)$ holds for all $\kappa = 1, \ldots, L-1$. Observe that $u_j(C_L, C_L|r, 2) = r - c_L$ and $u_j(C_\kappa, C_L|r, 2) = -c_\kappa$. Then it follows that $r > \max_\kappa \{c_L - c_\kappa\} = c_L$.

Now we show the sufficient condition. First, for each player, the strategy C_1 is strictly dominated by C_L. Actually, consider a specific player w_j, we can check that $u_j(C_L, C_L|r, 2) - u_j(C_1, C_L|r, 2) \ = r - c_L + c_1 = r - c_L > 0$. Observe that for all $\kappa = 2, \ldots, L-1$, we have $u_j(C_L, C_\kappa|r, 2) - u_j(C_1, C_\kappa|r, 2) = 2r - c_L > 0$. We can therefore eliminate C_1 from the strategy set resulting a reduced game, where each player's strategy set is $\{C_2, \ldots, C_L\}$. With a similar derivation as that for C_1, we can show that for this reduced game, the strategy C_2 is strictly dominated by C_L. We can therefore eliminate C_2 from the strategy set resulting in another reduced game, where each player's strategy set is $\{C_3, \ldots, C_L\}$. Repeating this elimination, we remove $C_3, \ldots, C_{L-1}$ sequentially. And finally we obtain a reduced game where each player only has one strategy C_L. By applying Lemma 3.1, we conclude the sufficient condition. □

Proof of Theorem 4.1. We first derive the expected utility $E[u_j(C_\kappa, s_{-j}|r)]$. For the sake of simplicity, in this proof s_{-j} represents $(C_L, \ldots, C_L)$. Let $E[u_j(C_\kappa, s_{-j}|r)|l]$ denote the conditional expected utility conditioned on that worker w_j receives a rating l. Then, with some basic probability arguments, we can express the expected utility in terms of conditional expected utility as $E[u_j(C_\kappa, s_{-j}|r)] = \sum_{l=1}^{L} \alpha_{\kappa,l} E[u_j(C_\kappa, s_{-j}|r)|l]$. Observe that given worker w_j receiving a rating l, this worker obtains a reward lager than zero, if and only if all the other $n-1$ players receive ratings lower or equal l. Otherwise receives a reward of zero. Let $\Pr[\ell, n-1-\ell|l, s_{-j}]$ denote the probability that ℓ of these $n-1$ players receive ratings equal to l and the other $n-1-\ell$ of them receive ratings lower than l. Observe that for a give ℓ, the utility for worker w_j is actually $\frac{nr}{\ell+1} - c_\kappa$, assuming this worker adopt the strategy C_κ. Then, by enumerating all the cases of $\ell = 0, 1, \ldots, n-1$, we can express the conditioned expected utility as $E[u_j(C_\kappa, s_{-j}|r)|l] = \sum_{\ell=0}^{n-1} \frac{nr}{\ell+1} \Pr[\ell, n-1-\ell|l, s_{-j}] - c_\kappa$. Observe that when a worker contributes a C_L solution, the probability that this worker receives a rating equal to l is $\alpha_{L,l}$, and the probability of receiving a rating lower than l is $\sum_{k=1}^{l-1} \alpha_{L,k}$. Note that $s_{-j} = (C_L, \ldots, C_L)$, then we obtain that $\Pr[\ell, n-1-\ell|l, s_{-j}] = \binom{n-1}{\ell} \alpha_{L,l}^{\ell} [\sum_{k=1}^{l-1} \alpha_{L,k}]^{n-\ell-1}$. Then it follows that $E[u_j(C_\kappa, s_{-j}|r)] = \sum_{l=1}^{L} \sum_{\ell=0}^{n-1} \binom{n}{\ell+1} (\sum_{k=1}^{l-1} \alpha_{L,k})^{n-\ell-1} \alpha_{\kappa,l} \alpha_{L,l}^{\ell} r - c_\kappa$ For a special case of $\kappa = L$, we can further reduce the above expression into a simpler form, namely $E[u_j(C_L, s_{-j}|r)] = r - c_L$. □

We now show that for any integer $1 \le \kappa < L$. There exists an integer $1 \le j < L$, such that for any $i > j$, it holds that $\alpha_{L,i} > \alpha_{\kappa,i}$, and for any $i < j$, we have $\alpha_{L,i} < \alpha_{\kappa,i}$. Let ℓ denote the maximum integer satisfying the condition $\alpha_{L,\ell} \le \alpha_{\kappa,\ell}$. In other words, for any $i > \ell$, we have $\alpha_{L,i} > \alpha_{\kappa,i}$. Actually, at least one $\ell < L$ exists, because $\alpha_{L,L} > \alpha_{\kappa,L}$. Now we show that for all $i < \ell$, we have $\alpha_{L,i} < \alpha_{\kappa,i}$. Before that we first show a basic claim, i.e., for all $i \le \kappa$, the condition $\alpha_{L,i} < \alpha_{\kappa,i}$ holds. Recall that each column of the confusion matrix $\boldsymbol{\alpha}$ is singly peaked, and $\alpha_{i,i}$ is the peak of the ith column. Since $i \le \kappa$, and $\kappa < L$, we can therefore conclude $\alpha_{L,i} < \alpha_{\kappa,i}$ holds for all $i \le \kappa$. Consider the case $\ell \le \kappa$. With the above claim, we can conclude this lemma. Consider the case $\ell \ge \kappa$. With the above claim, we can first obtain that $\alpha_{L,i} < \alpha_{\kappa,i}$ holds for all $i = 1, \ldots, \kappa$. The remaining thing is to consider $\kappa \le i < \ell$. Recall that each row of the confusion matrix $\boldsymbol{\alpha}$ is singly peak, and $\alpha_{i,i}$ is the peak of the ith row. Thus for all $\kappa \le i < \ell$, we have $\alpha_{\kappa,i} > \alpha_{\kappa,\ell}$, and $\alpha_{L,i} < \alpha_{L,\ell}$. Then it follows that $\alpha_{\kappa,i} - \alpha_{L,i} > \alpha_{\kappa,\ell} - \alpha_{L,\ell} \ge 0$, holds for all $\kappa \le i < \ell$, which concludes this lemma.

Now we prove the main results. Let us prove the necessary condition first. The condition to sustain the strategy profile $(C_L, \ldots, C_L)$ as an equilibrium, is expressed in Inequality (3). Examining this condition, we obtain one necessary condition: for all w_j, $E[u_j(C_L, s_{-j}|r)] - E[u_j(C_\kappa, s_{-j} \mid r)] > 0$, holds for all $C_\kappa \in \mathcal{C}_L \setminus C_L$, where $s_{-j} = (C_L, \ldots, C_L)$. We can expand this condition as

$$E[u_j(C_L, s_{-j}|r)] - E[u_j(C_\kappa, s_{-j}|r)] = c_\kappa - c_L + \left[1 - \sum_{l=1}^{L} \sum_{\ell=0}^{n-1} \binom{n}{\ell+1} \left(\sum\nolimits_{k=1}^{l-1} \alpha_{L,k}\right)^{n-\ell-1} \alpha_{\kappa,l} \alpha_{L,l}^{\ell}\right] r, \tag{11}$$

where $\kappa = 1, \ldots, L-1$. Observe that $c_\kappa < c_L$. Thus Equation (11) can be larger than zero, only if the multiplying factor for r is larger than zero. We seek to show this by applying Lemma 8.1. Let $\beta(l) = \sum_{\ell=0}^{n-1} \binom{n}{\ell+1} (\sum_{k=1}^{l-1} \alpha_{L,k})^{n-\ell-1} \alpha_{L,l}^{\ell}$, for the sake of simplicity. We can then express the multiplying factor of r derived in Equation (11) as $1 - \sum_{l=1}^{L} \alpha_{\kappa,l} \beta(l)$. We now show that $\beta(l)$ is monotonely increasing in terms of l, i.e., $\beta(1) < \beta(2) < \cdots < \beta(L)$. Recall that $0 \le \alpha_{L,1} \le \alpha_{L,2} \le \ldots \le \alpha_{L,L}$. Thus we have that $\sum_{k=1}^{l-1} \alpha_{L,k}$ is increasing in terms of l. We therefore conclude the increasing property of $\beta(l)$. We obtain that $\alpha_{\kappa,1}, \ldots, \alpha_{\kappa,L}$ and $\alpha_{L,1}, \ldots, \alpha_{L,L}$ satisfy the remaining required condition of Lemma 8.1. We can then apply Lemma 8.1 to obtain $1 - \sum_{l=1}^{L} \alpha_{\kappa,l} \beta(l) > 1 - \sum_{l=1}^{L} \alpha_{L,l} \beta(l) = 0$, where the last step follow that $\sum_{l=1}^{L} \alpha_{L,l} \beta(l) = 1$. This implies that the necessary condition can hold by setting a large enough amount of reward. We can then obtain the analytical expression for the critical value.

Now we show the sufficiency. Applying Lemma 3.1, one can observe that it is sufficient to show that iterated elimination of strictly dominated strategies eliminates all strategies but $(C_L, \ldots, C_L)$. First, let us show that for each worker w_j the strategy C_1 is strictly dominated by C_L. For simplicity, in the following of this proof s_{-j} denotes that all other workers except w_j play C_L, while s'_{-j} denotes that worker w_κ plays $s_\kappa \in \mathcal{C}_L$, where $\kappa \neq j$. For simplicity, let $\Delta_{(C_L, C_1, s_{-j})} = E[u_j(C_L, s_{-j}|r)] - E[u_j(C_1, s_{-j}|r)]$. It measures worker w_j's incentive to play C_L given two choices C_L, C_1. The higher the value of $\Delta_{(C_L, C_1, s_{-j})}$, the higher the incentive to choose C_L. Similarly we define $\Delta_{(C_L, C_1, s'_{-j})} = E[u_j(C_L, s'_{-j}|r)] - E[u_j(C_1, s'_{-j}|r)]$. Then it follows that $\Delta_{(C_L, C_1, s'_{-j})} - \Delta_{(C_L, C_1, s_{-j})}$ measures the change on worker w'_j incentive to subject to the perturbation of other workers' strategies vary from s_{-j} to s'_{-j}. Now we examine the impact of *confusion matrix*$\boldsymbol{\alpha}$ on this perturbation. Specifically, let $\Delta(\boldsymbol{\alpha}) = \Delta_{(C_L, C_1, s'_{-j})} - \Delta_{(C_L, C_1, s_{-j})}$. Observe that the lower the variability in rating, the higher the differentiation between s_{-j} and s'_{-j}, and the higher the differentiation between C_L and C_1. This implies that the impact of this perturbation becomes more significant. In other words the gap between $\Delta_{(C_L, C_1, s'_{-j})}$ and $\Delta_{(C_L, C_1, s_{-j})}$ enlarges. As one can imagine, this perturbation should change worker w_j's incentive monotonely, i.e., increasing the impact of this perturbation, should only increase (or decrease) worker w_j's incentive. Thus two intuitive bounds on $\Delta(\boldsymbol{\alpha})$ should be that when the variability in rating is the "minimum", i.e., $\boldsymbol{\alpha}$ is an identity matrix, we have $\Delta(\boldsymbol{\alpha}) > \frac{nr}{n_L+1} - r$, where n_L denotes the number of strategies from s'_{-j} that equals to C_L. Or when when the variability in rating is the "maximum", i.e., $\boldsymbol{\alpha}$ is a matrix with all entries $\frac{1}{L}$, we have $\Delta(\boldsymbol{\alpha}) = 0$. Thus we conclude that $\Delta(\boldsymbol{\alpha}) \ge 0$. This implies that $\Delta_{(C_L, C_1, s'_{-j})} \ge \Delta_{(C_L, C_1, s_{-j})} > 0$. Thus we conclude that C_1 is strictly dominated by C_L. We can therefore eliminate C_1, and obtain a reduced game. For this reduced game, we can similarly eliminate C_2. Repeating it, we finally eliminate all strategies but C_L.

Proof of Theorem 5.1. We first derive the expected utility $E[u_j(C_\kappa, s_{-j}|r)]$, where s_{-j} represents $(C_L, \ldots, C_L)$. With some basic probability arguments, we have $E[u_j(C_\kappa, s_{-j}|r)] = \sum_{l=2}^{\widetilde{L}} \alpha_{\kappa, L-\widetilde{L}+l} \sum_{\ell=0}^{n-1} \frac{nr}{\ell+1} \Pr[\ell, n-1-\ell|l, s_{-j}] + \sum_{l=1}^{L-\widetilde{L}+1} \alpha_{\kappa,l} \sum_{\ell=0}^{n-1} \frac{nr}{\ell+1} \Pr[\ell, n-1-\ell|1, s_{-j}]$, where adopt the notations $E[u_j(C_\kappa, s_{-j}|r)|l]$ and $\Pr[\ell,\ n-1-\ell|l, s_{-j}]$ as defined in the proof of Theorem 4.1. Consider a special case of $l=1$. It is impossible to receive a rating below 1, thus $\Pr[\ell, n-1-\ell|l, s_{-j}] = 0$, if $\ell < n-1$. And when $\ell = n-1$, we have $\Pr[n-1, 0|1, s_{-j}] = (\sum_{k=1}^{L-\widetilde{L}+1} \alpha_{L,k})^{n-1}$. Consider the case of $l \geq 2$. Observe that with strategy C_L, a worker receives a rating l with probability $\alpha_{L, L-\widetilde{L}+l}$, and receives a rating lower than l with probability $\sum_{k=1}^{L-\widetilde{L}+l-1} \alpha_{L,k}$. The we have $\Pr[\ell, n-1\ -\ell|l, s_{-j}] = \binom{n-1}{\ell} \alpha^{\ell}_{L,L-\widetilde{L}+l} [\sum_{k=1}^{L-\widetilde{L}+l-1} \alpha_{L,k}]^{n-\ell-1}$. Then we have $E[u_j(C_\kappa, s_{-j}|r)] = \sum_{l=1}^{L-\widetilde{L}+1} \alpha_{\kappa,l} r (\sum_{k=1}^{L-\widetilde{L}+1} \alpha_{L,k})^{n-1} - c_\kappa + \sum_{l=2}^{\widetilde{L}} \sum_{\ell=0}^{n-1} \binom{n}{\ell+1} (\sum_{k=1}^{L-\widetilde{L}+l-1} \alpha_{L,k})^{n-\ell-1} r \alpha_{\kappa, L-\widetilde{L}+l} \alpha^{\ell}_{L, L-\widetilde{L}+l}$. With a similar derivation as Theorem 4.1, one can obtain that when $\kappa = L$, the above expression can be reduced into a simple form, i.e, $E[u_j(C_L, s_{-j}|r)] = r - c_L$. □

Now we show that for all $1 \leq \kappa < L$, there exists an integer $1 \leq j < \widetilde{L}$, such that $\Pr[i|C_L, \langle \widetilde{\mathcal{L}}, \mathcal{C}_L, \mathcal{R}(\cdot) \rangle] > \Pr[i|C_\kappa, \langle \widetilde{\mathcal{L}}, \mathcal{C}_L, \mathcal{R}(\cdot) \rangle]$, holds $\forall i > j$. And for all $i < j \Pr[i|C_L, \langle \widetilde{\mathcal{L}}, \mathcal{C}_L, \mathcal{R}(\cdot) \rangle] < \Pr[i|C_\kappa, \langle \widetilde{\mathcal{L}}, \mathcal{C}_L, \mathcal{R}(\cdot) \rangle]$ holds. Recall that in the proof of Theorem 4.1, we showed that there exists an integer $j' \in \{1, \ldots, L-1\}$, such that for any $i > j'$, it holds that $\alpha_{L,i} > \alpha_{\kappa,i}$, and $\alpha_{L,i} < \alpha_{\kappa,i}$ holds for all $i < j$. Then examining the definition of $\Pr[i|C_L, \langle \widetilde{\mathcal{L}}, \mathcal{C}_L, \mathcal{R}(\cdot) \rangle]$ in Equation (4), one can observe that we only need to show $\Pr[1|C_L, \langle \widetilde{\mathcal{L}}, \mathcal{C}_L, \mathcal{R}(\cdot) \rangle] < \Pr[1|C_\kappa, \langle \widetilde{\mathcal{L}}, \mathcal{C}_L, \mathcal{R}(\cdot) \rangle]$. This is shown by the following arguments. Actually this inequality holds, because if $j' > L - \widetilde{L} + 1$, then we obtain that for all $i \leq L - \widetilde{L} + 1$, we have $\alpha_{L,i} < \alpha_{\kappa,i}$. Suppose $j' \leq L - \widetilde{L} + 1$. Then it follows that for all $i \geq 2$, we have $\Pr[i|C_L, \langle \widetilde{\mathcal{L}}, \mathcal{C}_L, \mathcal{R}(\cdot) \rangle] \geq \Pr[i|C_\kappa, \langle \widetilde{\mathcal{L}}, \mathcal{C}_L, \mathcal{R}(\cdot) \rangle]$.

Let us prove the necessary condition first. Observe that one necessary condition is that $E[u_j(C_L, s_{-j}|r)] - E[u_j (C_\kappa, s_{-j}|r)] > 0$, holds for all $C_\kappa \in \mathcal{C}_L \setminus C_L$, where $s_{-j} = (C_L, \ldots, C_L)$. We expand this condition as

$$\begin{aligned} &E[u_j(C_L, s_{-j}|r)] - E[u_j(C_\kappa, s_{-j}|r)] = c_\kappa - c_L + \\ &\left[1 - \sum_{l=1}^{L-\widetilde{L}+1} \alpha_{\kappa,l} \left(\sum_{k=1}^{L-\widetilde{L}+1} \alpha_{L,k}\right)^{n-1} - \sum_{l=2}^{\widetilde{L}} \sum_{\ell=0}^{n-1} \binom{n}{\ell+1} \right. \\ &\left. \left(\sum_{k=1}^{L-\widetilde{L}+l-1} \alpha_{L,k}\right)^{n-\ell-1} \alpha_{\kappa, L-\widetilde{L}+l} \alpha^{\ell}_{L, L-\widetilde{L}+l} \right] r, \end{aligned} \tag{12}$$

where $\kappa = 1, \ldots, L-1$. Since $c_\kappa - c_L < 0$, the necessary condition can hold only if the multiplying factor of r derived in Eq. (12) is larger than zero. For the sake of simplicity, we let $\beta_l = \sum_{\ell=0}^{n-1} \binom{n}{\ell+1} (\sum_{k=1}^{L-\widetilde{L}+l-1} \alpha_{L,k})^{n-\ell-1} \alpha^{\ell}_{L,L-\widetilde{L}+l}$, for all $l \geq 2$, and let $\beta_1 = (\sum_{k=1}^{L-\widetilde{L}+1} \alpha_{L,k})^{n-1}$. The multiplying factor of r can be expressed as $1 - \sum_j \beta_l \Pr[l|C_\kappa, \langle \widetilde{\mathcal{L}}, \mathcal{C}_L, \mathcal{R}(\cdot) \rangle]$. Observe that $\beta_L > \cdots > \beta_1$. Then by applying Lemma 8.1, and with a similar derivation as Theorem 4.1 we conclude the necessary condition. The proof for the sufficient condition is similar to that of Theorem 4.1.

Proof of Lemma 6.1. Let n^* denote the optimal bundling size. Let $z(n) = \widetilde{\mu}(1 - \frac{c_L}{\underline{r}}) + (1 - \widetilde{\mu}) \frac{n}{\lambda}$. Then we obtains that $z(n^*) \leq z(2)$. Note that $z(n^*) \geq (1 - \widetilde{\mu}) \frac{n^*}{\lambda}$ and $z(2) \leq \widetilde{\mu} + (1 - \widetilde{\mu}) \frac{2}{\lambda}$. Hence we have $(1 - \widetilde{\mu}) \frac{n^*}{\lambda} \leq \widetilde{\mu} + (1 - \widetilde{\mu}) \frac{2}{\lambda}$, which yields that $n^* \leq \frac{\lambda}{1 - \widetilde{\mu}} (\widetilde{\mu} + (1 - \widetilde{\mu}) \frac{2}{\lambda})$ This proof is then complete. □

9 Related Work

Crowdsourcing has recently drawn a lot of attentions. Many aspects of crowdsourcing are being studied, e.g., applying the concept of crowdsourcing to design new applications and algorithms [9], [11], [27], user behavior study [28], and investigating the performance issues, like quality management [29], [30], incentive design [31], fairness [32], etc. A survey can be found in [33].

A variety of real-world crowdsourcing platforms have been developed. Based on the types of tasks, they can be broadly classified into *microtasking* [14] and *macrotasking* [15]. Microtasking crowdsourcing systems focus on small and repetitive tasks. Typical microtasking crowdsourcing systems include Amazon Mechanical Turk [4], and Microtask [16]. Gamification [27], [34] is a microtasking crowdsourcing system as well. Macrotasking crowdsourcing are mainly designed to solve challenging and innovative tasks, which require special skills. Elance [5] and Fiverr [17] are two real-world macrotasking crowdsourcing systems. A variety works [35], [36] investigated financial incentives for microtasking crowdsourcing systems. This paper focuses on macrotasking crowdsourcing systems, in which the incentive design is quite different from microtasking.

A variety of works applied the "winner-takes-it-all" scheme to model crowdsourcing contest [37], [38], [39]. In particular, they deployed all pay auctions in their mechanisms. These approaches only apply for the scenario that each task requires multiple workers to solve, which results in that the requester needs to set a large reward to incentivize workers. In macrotasking crowdsourcing systems, it is quite common that a requester only employs one worker to solve a task. Our approach applies even when a task require one worker to solve. Furthermore, previous approaches did not consider the human factors in assessing solution quality, while we formally explore this aspect.

A variety of approaches have been proposed to price tasks. One typical approach is mining the price from data [40], [41]. More concretely, building models to infer workers' benefit and cost in solving tasks. Another approach determines the price automatically or dynamically via designing some efficient mechanisms. Typical examples include a multi-armed bandits based pricing mechanism [42], and auction based pricing mechanisms [43], [44]. However, these approaches did not consider the human factors in assessing solution quality, while approach formally explores it. Furthermore, they did not solve the challenge of *free-riding* of workers, or *denial of payment* of requesters. Our incentive mechanism addresses this challenge.

Recently, a few workers investigated incentive mechanism design for crowdsourcing applications. An experiment in an online labor market was conducted to understand the effectiveness of a collection of social and finical incentive schemes [31]. A game-theoretic model of an online question and answer forum was developed in [45], where the authors investigated the impact of various score sharing rules on the incentives for providing solution. A few reputation based incentive protocols were developed in [20], [46], [47], [48], [49], where they model the interactions between workers and requesters as a repeated or stochastic game. They induce incentive via maintaining a reputation for each worker to track the contribution history and penalize workers with low reputation. Our paper is different from theirs in the following aspects. First, we are the first to conduct a unified study on incentive mechanism and rating system design. We studied how a rating system may influence worker participating incentive. Second, our incentive mechanism is simple and we demonstrate its high effectiveness via experiments on a dataset from Elance. Third, we present metrics to characterize the effectiveness of a rating system, and we identify the redundancy of the rating system in Elance. Fourth, we develop an optimization framework to infer model parameters from data, and we demonstrate how to apply our framework to analyze real-world data.

10 Conclusion

The is the first paper which conducts a unified study on incentive and rating system design for crowdsourcing systems. We designed a class of simple but effective incentive mechanisms, which consist of a "*task bundling scheme*" and a "*rating system*". We propose a probabilistic model to capture various human factors, e.g., personal preference or bias in rating, and quantified their impact on the incentive mechanism. We developed a model to characterize the design space rating systems and quantify the impact of a rating system on worker participating incentive. We formulated an optimization framework to select appropriate rating system parameters to tradeoff between "*crowdsourcing system efficiency*", and the "*rating system complexity*". We also formulated an optimization framework to select appropriate bundling size, which can tradeoff between system efficiency and service delay. We conducted experiments on a dataset from Elance. Using our optimization framework, we infer model parameters from the collected data. We showed that using our incentive mechanism can achieve at least 99.95 percent of the theoretical maximum system efficiency with a service delay of at most 2.3639 hours. We found out that the rating system in Elance is too complex, and it can be simplified to a binary rating system.

Acknowledgments

The work of John C. S. Lui was supported in part by the GRF 415112. The authors would like to thank anonymous reviewers for their helpful comments. An earlier conference version of the paper appeared in IEEE MASCOTS 2014 [1]. In this journal version, we add an optimization framework to select parameters for the rating system, an optimization framework to select parameters for the incentive mechanism, and experiments on a dataset from Elance.

References

[1] H. Xie, John C. S. Lui, and W. Jiang, "Mathematical modeling of crowdsourcing systems: Incentive mechanism and rating system design," in *Proc. IEEE 22nd Int. Symp. Modelling, Anal. Simul. Comput. Telecommun. Syst.*, Sep. 2014, pp. 181–186.

[2] J. Howe, "The rise of crowdsourcing," *Wired Mag.*, vol. 14, no. 6, pp. 1–4, 2006.

[3] J. Howe, *Crowdsourcing: Why the Power of the Crowd is Driving the Future of Business.* New York, NY, USA: Crown, 2008.

[4] (2005). Amazon Mechanical Turk, [Online]. Available: https://www.mturk.com

[5] (1999). Elance, [Online]. Available: https://www.elance.com/

[6] (2005). Yahoo! Answers, [Online]. Available: http://answers.yahoo.com

[7] (2000). iStockphoto, [Online]. Available: http://www.istockphoto.com/

[8] (2000). Threadles, [Online]. Available: http://www.threadless.com/

[9] W. Mason and S. Suri (2012). Conducting behavioral research on amazon mechanical turk. *Behavior Res. Methods. 44(1)*, pp. 1–23. [Online]. Available: http://dx.doi.org/10.3758/s13428-011-0124-6

[10] G. Paolacci, J. Chandler, and P. Ipeirotis, "Running experiments on amazon mechanical turk," *Judgment Decision Making*, vol. 5, no. 5, pp. 411–419, 2010.

[11] A. Ghosh, S. Kale, and P. McAfee, "Who moderates the moderators?: Crowdsourcing abuse detection in user-generated content," in *Proc. 12nd ACM Conf. Electron. Commerce*, 2011, pp. 167–176.

[12] P. Naghizadeh and M. Liu, "Perceptions and truth: A mechanism design approach to crowd-sourcing reputation," *IEEE/ACM Trans. Netw.*, vol. 24, no. 1, pp. 163–176, Feb. 2014.

[13] D. R. Choffnes, F. E. Bustamante, and Z. Ge, "Crowdsourcing service-level network event monitoring," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 40, no. 4, pp. 387–398, Aug. 2010.

[14] (2012). Microtasking, [Online]. Available: https://en.wikipedia.org/wiki/Microwork

[15] (2013). Macrotasking, [Online]. Available: https://en.wikipedia.org/wiki/Macrotasking

[16] (2009). Microtask, [Online]. Available: http://www.microtask.com/

[17] (2010). Fiverr, [Online]. Available: https://www.fiverr.com/

[18] L. A. Adamic, J. Zhang, E. Bakshy, and M. S. Ackerman, "Knowledge sharing and yahoo answers: Everyone knows something," in *Proc. 17th Int. Conf. World Wide Web*, 2008, pp. 665–674.

[19] C.-J. Ho and J. W. Vaughan, "Online task assignment in crowdsourcing markets," presented at the AAAI 26th Conf. Artificial Intelligence, Toronto, ON, Canada, 2012.

[20] H. Xie, John C.S. Lui, and D. Towsley, "Incentive and reputation mechanisms for online crowdsourcing systems," in *Proc. IEEE 23rd Int. Symp. Quality Serv.*, Jun. 2015, pp. 207–212.

[21] H. Lauw, E.-P. Lim, and K. Wang, "Bias and controversy in evaluation systems," *IEEE Trans. Knowl. Data Eng.*, vol. 20, no. 11, pp. 1490–1504, Nov. 2008.

[22] H. W. Lauw, E.-P. Lim, and K. Wang, "Quality and leniency in online collaborative rating systems," *ACM Trans. Web*, vol. 6, no. 1, pp. 4-1–4-27, 2012.

[23] R. Gibbons, *Game Theory for Applied Economists*. Princeton, NJ, USA: Princeton Univ. Press, 1992.

[24] E. I. Sparling and S. Sen, "Rating: How difficult is it?" in *Proc. 5th ACM Conf. Recommender Syst.*, 2011, pp. 149–156.

[25] (2016). DataSet, [Online]. Available: https://dl.dropboxusercontent.com/u/18801699/Elance.txt

[26] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.

[27] L. von Ahn, R. Liu, and M. Blum, "Peekaboom: A game for locating objects in images," in *Proc. SIGCHI Conf. Human Factors Comput. Syst.*, 2006, pp. 55–64.

[28] A. Kittur, E. H. Chi, and B. Suh, "Crowdsourcing user studies with mechanical turk," in *Proc. SIGCHI Conf. Human Factors Comput. Syst.*, 2008, pp. 453–456.

[29] P. G. Ipeirotis, F. Provost, and J. Wang, “Quality management on amazon mechanical turk,” in *Proc. ACM SIGKDD Workshop Human Comput.*, 2010, pp. 64–67.

[30] D. R. Karger, S. Oh, and D. Shah, “Efficient crowdsourcing for multi-class labeling,” in *Proc. ACM SIGMETRICS/Int. Conf. Meas. Model. Comput. Syst.*, 2013, pp. 81–92.

[31] W. Mason and D. J. Watts, “Financial incentives and the ”performance of crowds,” in *Proc. ACM SIGKDD Workshop Human Comput.*, 2009, pp. 77–85.

[32] P. Hyman, “Software aims to ensure fairness in crowdsourcing projects,” *Commun. ACM*, vol. 56, no. 8, pp. 19–21, 2013.

[33] M.-C. Yuen, I. King, and K.-S. Leung, “A survey of crowdsourcing systems,” in *Proc. IEEE 3rd Int. Conf. Privacy, Security, Risk Trust IEEE 3rd Int. Conf. Social Comput.*, Oct. 2011, 766–773.

[34] L. Von Ahn, “Games with a purpose,” *Computer*, vol. 39, no. 6, pp. 92–94, 2006.

[35] N. Kaufmann, T. Schulze, and D. Veit, “More than fun and money. worker motivation in crowdsourcing-a study on mechanical turk,” in *Proc. 17th Amer. Conf. Inf. Syst.*, 2011, vol. 11, pp. 1–11.

[36] G. Kazai, “In search of quality in crowdsourcing for search engine evaluation,” in *Proc. 33rd Eur. Conf. Adv. Inf. Retrieval*, 2011, pp. 165–176.

[37] S. Chawla, J. D. Hartline, and B. Sivan, “Optimal crowdsourcing contests,” in *Proc. 23rd Annu. ACM-SIAM Symp. Discr. Algorithms*, 2012, pp. 856–868.

[38] D. DiPalantino and M. Vojnovic, “Crowdsourcing and all-pay auctions,” in *Proc. 10th ACM Conf. Electron. Commerce*, 2009, pp. 119–128.

[39] H. Xu and K. Larson, “Improving the efficiency of crowdsourcing contests,” in *Proc. Int. Conf. Auton. Agents Multi-Agent Syst.*, 2014, pp. 461–468.

[40] D. F. Bacon, D. C. Parkes, Y. Chen, M. Rao, I. Kash, and M. Sridharan, “Predicting your own effort,” in *Proc. 11st Int. Conf. Auton. Agents Multiagent Syst.*, 2012, pp. 695–702.

[41] J. J. Horton and L. B. Chilton, “The labor economics of paid crowdsourcing,” in *Proc. 11st ACM Conf. Electron. Commerce*, 2010, pp. 209–218.

[42] L. Tran-Thanh, S. Stein, A. Rogers, and N. R. Jennings, “Efficient crowdsourcing of unknown experts using multi-armed bandits,” in *Proc. 20th Eur. Conf. Artif. Intell.*, 2012, pp. 768–773.

[43] Y. Singer and M. Mittal, “Pricing mechanisms for crowdsourcing markets,” in *Proc. 22nd Int. Conf. World Wide Web*, 2013, pp. 1157–1166.

[44] A. Singla and A. Krause, “Truthful incentives in crowdsourcing tasks using regret minimization mechanisms,” in *Proc. 22nd Int. Conf. World Wide Web*, 2013, pp. 1167–1178.

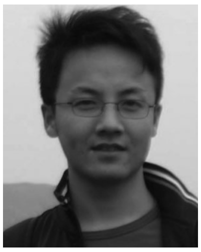
[45] S. Jain, Y. Chen, and D. C. Parkes, “Designing incentives for online question and answer forums,” in *Proc. 10th ACM Conf. Electron. Commerce*, 2009, pp. 129–138.

[46] C.-J. Ho, Y. Zhang, J. Vaughan, and M. van der Schaar, “Towards social norm design for crowdsourcing markets,” presented at the 4th Human Computation Workshop, Austin, TX, USA, 2012.

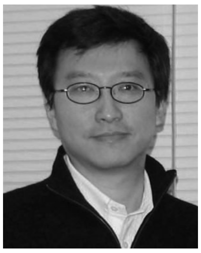
[47] Y. Xiao, Y. Zhang, and M. van der Schaar, “Socially-optimal design of crowdsourcing platforms with reputation update errors,” in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, May 2013, pp. 5263–5267.

[48] H. Xie, John C.S. Lui, J. W. Jiang, and W. Chen, “Incentive mechanism and protocol design for crowdsourcing systems,” in *Proc. 52nd Annu. Allerton Conf. Commun., Control, Comput.*, Sep./Oct. 2014, pp. 140–147.

[49] Y. Zhang and M. van der Schaar, “Reputation-based incentive protocols in crowdsourcing applications,” in *Proc. 31st Annu. IEEE Int. Conf. Comput. Commun.*, Mar. 2012, pp. 2140–2148.



Hong Xie received the BEng degree from the School of Computer Science and Technology, the University of Science and Technology of China, Hefei, China, in 2010, and the PhD degree with the Department of Computer Science and Engineering, the Chinese University of Hong Kong, Hong Kong, in 2015, proudly under the supervision of Prof. J. C. S. Lui. He is currently a postdoctoral research fellow at the School of Computing, National University of Singapore, Singapore, working closely with Prof. R. T. B. Ma. His research interests include network economics, data analytics, machine learning, stochastic modeling, approximation algorithms, crowdsourcing and online social networks, etc. His personal interests include movie, hiking, pingpang, basketball, etc. He is a member of IEEE and a member of ACM.



John C. S. Lui received the PhD degree in computer science from the University of California, Los Angeles, CA, USA. He is currently the Choh-Ming Li chair professor in the Department of Computer Science and Engineering, Chinese University of Hong Kong, Hong Kong. His current research interests include communication networks, system security (e.g., cloud security, mobile security, etc.), network economics, network sciences, large-scale distributed systems and performance evaluation theory. He serves in the editorial board of IEEE/ACM *Transactions on Networking*, *IEEE Transactions on Computers*, *IEEE Transactions on Parallel and Distributed Systems*, *Journal of Performance Evaluation and International Journal of Network Security*. He was the chairman of the CSE Department from 2005 to 2011. He received various departmental teaching awards and the CUHK Vice-Chancellors Exemplary Teaching Award. He is also a corecipient of the IFIP WG 7.3 Performance 2005 and the IEEEIFIP NOMS 2006 Best Student Paper Awards. His personal interests include films and general reading. He is an elected member of the IFIP WG 7.3, fellow of the ACM, fellow of the IEEE, and Croucher senior research fellow.