Dynamic Contract Design for Heterogenous Workers in Crowdsourcing for Quality Control

Chenxi Qiu*, Anna Cinzia Squicciarini*, Sarah Michele Rajtmajer†, and James Caverlee‡
*College of Information Science and Technology, The Pennsylvania State University
Email: {czq3, acs20}@psu.edu
† Quantitative Scientific Solutions, Arlington, VA 22203
Email: sarah.rajtmajer@qs-2.com
‡ Department of Computer Science and Engineering, Texas A&M University
Email: caverlee@cse.tamu.edu

***Abstract*—Crowdsourcing sites heavily rely on paid workers to ensure completion of tasks. Yet, designing a pricing strategies able to incentivize users' quality and retention is non trivial. Existing payment strategies either simply set a fixed payment per task without considering changes in workers' behaviors, or rule out poor quality responses and workers based on coarse criteria. Hence, task requesters may be investing significantly in work that is inaccurate or even misleading. In this paper, we design a dynamic contract to incentivize high-quality work. Our proposed approach offers a theoretically proven algorithm to calculate the contract for each worker in a cost-efficient manner. In contrast to existing work, our contract design is not only adaptive to changes in workers' behavior, but also adjusts pricing policy in the presence of malicious behavior. Both theoretical and experimental analysis over real Amazon review traces show that our contract design can achieve a near optimal solution. Furthermore, experimental results demonstrate that our contract design 1) can promote high-quality work and prevent malicious behavior, and 2) outperforms the intuitive strategy of excluding all malicious workers in terms of the requester's utility.**

I. Introduction

Crowdsourcing markets have emerged as a popular platform for matching available workers with tasks to complete. Payment to a worker for completing a particular task is typically set by the task requester. Most existing approaches to pricing policies set a fixed payment per task, which is consistent across the set of workers and remains constant for the duration of work [1], [2]. These approaches often do not account for past or anticipated quality of work, and are therefore not optimal for returning value on investment. Some recent works have improved upon fixed-payment strategies by dynamically adjusting workers compensation based on their recent quality of work [3]–[12]. However, these pricing strategies are built based on the assumption that workers are homogeneous, i.e., having the same objective and the same strategy, which leaves quality of work in jeopardy, especially when dishonest workers or malicious workers exist in the system. In practice, the quality of work delivered by crowdsourced workers assigned to a given task is heterogeneous. Besides different expertise and dedication, workers might also have different objectives. While most workers may simply aim at being compensated for their work, others may aim to divert the task toward a different outcome (e.g. provide highly positive reviews for a given product or location), irrespective of whether this outcome is consistent with reality (e.g. the quality of the product or location itself).

In this paper, we consider the case of a set of heterogeneous workers (i.e., some workers are honest and some are malicious) completing a set of tasks posted by a task requester. We design a dynamic contract for repeated tasks whereby individual contracts derived for each worker, for each task, may incentivize high-quality work and prevent malicious behaviors. To provide incentives for high-quality responses from honest workers and prevent pollution from malicious ones, we propose a quality-contingent payment strategy to specify the compensation offered to each worker based on his past work. In particular, we prove that the proposed pricing policy outperforms simple exclusion of all suspected malicious workers, whose malicious behavior may be temporary or targeted in scope or masked through collusion.

Our approach ensures that the goal of the requester, which is to maximize his expected utility (i.e. the value he obtains from the completed work minus the payments made) is met. We call the effort to maximize a task requester's expected utility the *contract design problem*, and treat it moving forward as a bi-level programming (BiP) problem.

We note that given the large worker population in most Crowdsourcing platforms, directly solving the contact design problem by considering all the workers together will yield a large number of *decision variables* involved in BiP, which is computational intractable. Hence, we decompose the BiP problem into a set of subproblems, which can be solved in parallel in the interest of managing complexity. Here, each subproblem aims to determine the contract for a single worker and hence only attempts to solve one worker's problem at time. Furthermore, decomposition is complicated by potential underlying interactions amongst workers. For example, a group of malicious workers may share information for mutual benefit or make joint decisions to further the same objective (we refer to these workers as "collusive"). As a solution, we formulate a subproblem for each collusive community, wherein a community includes malicious workers who are suspected to be collusive based on their historical record and whether they target the same objective.

Finally, to solve each subproblem, we devise an algorithm that can achieve near optimal solution with low time complexity. We provide both upper bound and lower bound of our solution (Theorem 4.1).

We validate this method on an Amazon dataset [13] containing 118,142 product reviews, labelled with ground truth labels on reviewers' honest or malicious intent. We show that our contract not only achieves near optimal return for the task requester, but also incentivises high-quality performance from honest workers and prevents pollution from malicious workers through differential compensation strategies. In addition, we demonstrate that our contract design outperforms a baseline strategy in which all the malicious workers are simply eliminated.

In summary, our contribution is three-folded:

1) We formulate a new contract design problem, of which the goal is to design a contract to each worker based on the worker's performance and objectives, to consequently maximize the requester's utility. To our knowledge, this is the first work studying dynamic contract in crowdsourcing with the consideration of heterogeneous workers. Our model accounts not only for malicious workers, but also for possible interactions within them, addressing the case where workers collude together toward a shared malicious goal.

2) To increase the time-efficiency, we propose to decompose the contract design problem into a set of subproblems, where each subproblem only deals with the contract for a single worker or a single collusive community. We design a theoretically proven algorithm for each subproblem, with tight upper bound and lower bound.

3) We carry out extensive empirical validation using a real Amazon review dataset. Our results demonstrate that our design contract is near to the optimal, and also provide higher compensation to honest workers than to malicious workers.

The remainder of the paper is organized as follows. Section II describes the model and Section III formulates the BiP problem. Section IV presents the approach for the problem. Section V evaluates the performance of our approach. Section VI lists the related works. Finally, Section VII concludes the paper with remarks on our future work.

II. The Model

In this section, we introduce the system model underlying our approach. Throughout the paper, we focus on the case of a task requester who hires a set of workers to write reviews (e.g., for products on Amazon or restaurants on Yelp). As Fig. 1 indicates, the task requester first posts the task, which includes a set of individual subtasks or rounds. For each worker, the requester also produces a contract specifying the payment for completing the task. Each worker decides whether to accept or decline the task requester's offer of work and associated payment. As workers complete tasks, they may be endorsed by other workers (through likes or upvotes) or receive feedback through public comments.

The pool of workers selecting tasks includes both honest and malicious workers. Honest workers aim to provide their

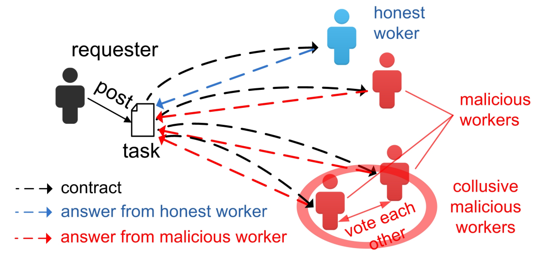


Fig. 1. Framework of the model

services in exchange for compensation, maximizing a utility directly proportional to financial gain and inversely proportional to effort level, as expended effort represents a "cost" to the worker in a general sense (in terms of time, energy, or missed opportunities). For each round, each worker chooses an amount of effort to invest in order to maximize his expected utility for the task.

Malicious workers have an additional hidden agenda, e.g., to introduce bias through their responses, or add noise to the collective task. Furthermore, malicious workers may work independently or in collusion with one another, where collusive workers will work together to spoil the same set of tasks.

The task requester defines the individualized contract based on estimated probability of a given worker being malicious [14], [15], either independently or in collusion with other workers, to evaluate the worker's contribution to the requester's utility.

Note that, although not certain, the requester can estimate an individual worker's performance and expected behavior with some ease, e.g. by comparing a worker's response with the estimated true response from a small number of experts or by the number of positive endorsements a given review receives (other approaches include [14]–[17]).

Further, in deciding the worker's price, the requester also considers other factors, like worker's obtained feedback and actual accuracy[1].

The task requester can then adjust the contract from one round to another within the same task.

The *object* of the requester is to determine an optimal contract or payment *function*, so as to obtain as accurate and valuable reviewsas possible, while minimizing workers' costs.

A. Workers' and task requester's models

Consider the scenario in which a requester hires N workers $\{1, 2, \ldots, N\}$ to perform a set of T individual tasks. Coordination of tasks is assumed, i.e., all workers finish one task before any move to the next. At the start of each task, the requester posts an updated contract for each worker specifying the offered *compensation* for completing that task. The compensation offered to each worker for each task is dependent on his recent performance, where recent is here defined as the single preceding task. We use a single preceding task as

[1] A malicious worker may be biased but still accurate within a certain acceptable range, sufficient to bring gain to the requester's utility

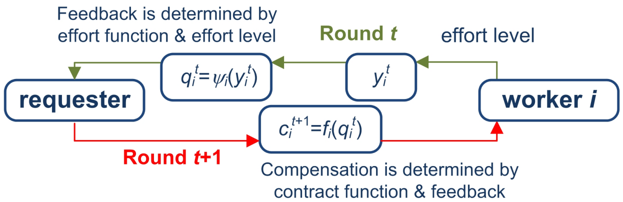


Fig. 2. Relationship between the requester and worker i

a proxy for some finite work history largely for simplicity, but would argue that in the longer-time horizon this decision is justifiable as the number of tasks considered approaches a representative sample. Table I lists the main notations that will be used throughout the model.

TABLE I
NOTATIONS

Notation	Description
N	The number of workers
T	The number of individual tasks
q_i^t	The feedback of worker i in round t
$q_{\mathscr{A}_j}^t$	The feedback of collusive community j in round t
c_i^t	The compensation to worker i in round t
y_i^t	The effort level of worker i in round t
f_i	The contract function to worker i
ψ_i	The strategy function of worker i
$\psi_{\mathscr{A}_j}$	The strategy function of collusive community j
p^t	The benefit of the requester in round t
ζ_i	The approximated contract function
$\Delta x_{i,l}$	The *contract increment* on $[d_{i,l-1}, d_{i,l})$
$\alpha_{i,l}$	The *contract slope* on $[d_{i,l-1}, d_{i,l})$
w_i^t	The weight of worker i's feedback to the requester
U^t	The utility function in round t. U_{req}^t, $U_{\text{HU }i}^t$, $U_{\text{MU }i}^t$, and $U_{\text{MU }\mathscr{A}_j}^t$ represent the utility function of the requester, honest worker i, malicious worker i, collusive community j
$\mathbf{x}_i$	Discrete compensations

We use c_i^t to represent the compensation that worker i is offered for completing the t-th task. Let c_i^t be given by

$$c_i^t = f_i\left(q_i^{t-1}\right), \tag{1}$$

where $f_i : [0,\infty) \mapsto [0,\infty)$, called the *contract function*, is a mapping from worker i's feedback (e.g. support rate or number of positive endorsements) for task $t-1$, q_i^{t-1} to his compensation for task t. We assume that f_i is *monotonically increasing* with q_i^{t-1}.

Consider Fig. 2 for an exemplified workflow. Here, feedback reflects the positive gain for the requester generated by the completion of the task. For example, for Amazon product reviews, feedback can be measured by the number of positive "likes" or "helpfulness" endorsements obtained from other workers as compared to the average number of positive endorsements for similar products.

According to the contract function provided by the requester, each worker chooses his effort level to maximize his utility. Let y_i^t denote worker i's effort level at a given round t, where y_i^t is defined in a continuous region $\mathbb{R}^* = \mathbb{R}^+ \cup 0$. In particular, if worker i doesn't accept the request for the task in round t, then $y_i^t = 0$. Note that effort level we defined here takes into account not only the time duration a worker takes to complete the task, but also the expertise of the worker. In other words, we define effort level as a combination, e.g., product, of workers' working time and expertise. In practice, expertise and time spent are difficult or impossible to measure, so we use an appropriate proxy for the given context. In the case of Amazon product review, for example, this may be length of generated text.

Malicious workers are dominated by a different utility function. As discussed, the goal of the malicious worker is to maximize the influence of a biased response or insert noise into the dataset, thereby *misleading the task requester* while making a reasonable effort, that is an effort that is bounded by an inherent willingness to do harm.

Formally, for a non-collusive malicious worker i, let ψ_i be the *strategy function*, describing the relationship between the feedback q of worker i and his effort level y for task t:

$$q_i^t = \psi_i(y_i^t). \tag{2}$$

We assume ψ_i is concave and *twice differentiable*. Suppose a set of workers $\mathscr{A}_j$ are not only malicious, but are also colluding with each other. Then, we consider the workers in $\mathscr{A}_j$ as a "single meta worker" and let $\psi_{\mathscr{A}_j}$ describe the relationship between workers' sum feedback and their sum effort levels in each round t:

$$q_{\mathscr{A}_j}^t = \psi_{\mathscr{A}_j}\left(\sum_{i\in\sum_{l\in\mathscr{A}_j}} y_l^t\right). \tag{3}$$

In reality, it is not straightforward to detect which workers may be collusive. According to [13], we can safely assume that two malicious workers collude if they target the same product. The intuition is that collusive workers are recruited from the same source and paid to target the same task. We also define a set of malicious workers $\mathscr{A}$ as a *collusive community* if, for each worker $i \in \mathscr{A}$, exists another worker $j \in \mathscr{A}$ that has the same target product with worker i.

Finally, we define the *benefit* of the requester getting from workers' reviews. On the one hand, the requester cares that workers' reviews are useful to the community. This is exemplified by how many positive endorsements any given review obtains. Hence, the contribution of a worker's review to the requester should be proportional to the worker's ability to collect positive endorsements. On the other hand, the requester also aims at ensuring workers' reviews' "accuracy", i.e., each review should be as truthful as possible. In practice, given the possible level of subjectivity involved in any review of a given service or product, accuracy is measured by how close workers' reviews are to the experts' reviews, where experts are defined as the workers whose accuracy and positive endorsements (along with reputation) are both higher than the thresholds specified by the system. We use l_i^t to represent the review from worker i in round t and use $\overline{l^t}$ to represent the average review score given by experts at round t, and we consider $\overline{l^t}$ as the "ground truth" of the task. The closer a worker i's review l_i^t to $\overline{l^t}$, the more valuable the worker's review is. Besides accuracy, we also give a penalty to worker i's feedback if the worker has probability e_i^{mal} to be malicious and has A_i partners[2] In summary, we describe the benefit of the requester in round t by

$$p^t = \sum_{i=1}^{N} w_i^t q_i^t \tag{4}$$

where

$$w_i^t = \frac{\rho}{|l_i^t - \overline{l^t}|} - \kappa e_i^{\mathrm{mal}} - \gamma A_i \tag{5}$$

where ρ, κ, and γ are coefficients of accuracy, malicious probability, and the number of partners.

III. A Stackelberg Framework for Optimal Contracts

In this section, we model the interaction between the task requester and workers as a Stackelberg competition (Section III-B), a strategic game in which the leader makes the decision first and then the followers move sequentially. In particular, we consider the task requester as the leader (who specifies the contract for each worker) and the workers as the followers (who choose the effort level to complete the task). We assume that the game takes place in discrete time, where each iteration of the game represents the completion of one task.

A. Contract function approximation

Before formulating the problem, we here discuss how we represent each contract function through a group of *decision variables*. These variables, however, can only describe the contract with discrete inputs. Since the input of each contract function (worker's feedback) is defined in continuous region, we partition the worker's feedback region with discrete points, and approximate the function by assuming compensation increases linearly in each interval. Then, each contract function can be represented as a *piecewise linear* function.

Precisely, as Fig. 3 shows, we partition the effort region of workers into m intervals $[0,\delta)$, $[\delta,2\delta)$, ..., $[(m-1)\delta,m\delta)$. Correspondingly, the influence region of each independent worker i is partitioned to $[d_{i,0},d_{i,1})$, $[d_{i,1},d_{i,2})$, ..., $[d_{i,m-1},d_{i,m})$, where each $d_{i,l} = \psi_i(l\delta)$ $(l = 0,1,...,m)$. After that, we can approximate f_i in terms of a set of discrete compensations $\mathbf{x}_i = [x_{i,1},...,x_{i,m}]$, where each $x_{i,l} = f_i(d_{i,l})$ and $x_{i,l} \le x_{i,l+1}$. Let $\Delta x_{i,l} = x_{i,l} - x_{i,l-1}$, $\Delta d_{i,l} = d_{i,l} - d_{i,l-1}$, $\alpha_{i,l} = \Delta x_{i,l}/\Delta d_{i,l}$, and $\Delta x_{i,l}$ and $\alpha_{i,l}$ is called the *contract increment* and *contract slope* on $[d_{i,l-1},d_{i,l})$, respectively. Then, the contract function f_i can be approximated by a piecewise linear function ζ_i:

$$\zeta_i\left(\mathbf{x}_i, q_i^t\right) = \alpha_{i,l}\left(q_i^t - d_{i,l-1}\right) + x_{i,l-1}, \quad q_i^t \in [d_{i,l-1}, d_{i,l}) \tag{6}$$

[2]As stated before, though non-trivial it is possible to estimate a worker's probability of being malicious, e.g., by comparing the worker's reviews with "ground truth" (i.e., expertise reviews) [14], or by applying machine learning techniques to analyze their comments [15].

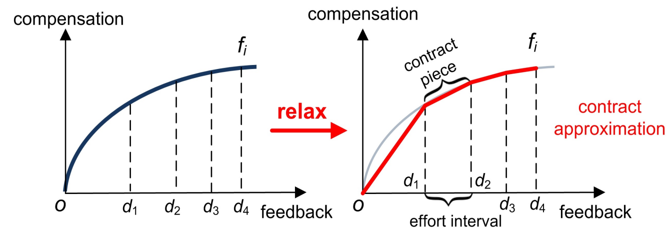


Fig. 3. An example of contract discretization

where $l = 1,...,m$. In the remain part of this part, we call the segment of the piecewise linear contract function defined in $[d_{i,k-1},d_{i,k})$ a *contract piece* on $[d_{i,k-1},d_{i,k})$.

B. Problem formulation

We first respectively describe the objectives and constraints of the task requester and the workers in detail.

Task requester. The task requester's goal is to maximize his benefit in each round with minimum cost. Hence, the requester's utility in round t can be expressed as a linear combination of his benefit in round t, p^t, and the total compensation paid to all the workers in round $t+1$, $\sum_{i=1}^N c_i^{t+1}$:

$$U_{\mathrm{req}}^t = p^t - \mu \sum_{i=1}^{N} c_i^t, \tag{7}$$

where $\mu > 0$ is the weight given to the compensation, respectively. Suppose there are N_1 honest workers, N_2 non-collusive malicious workers, and N_3 collusive communities, then the requester's utility can be represented by (without loss of generality, in the following part, we let the index of honest workers and non-collusive malicious workers be $\{1,...,N_1\}$ and $\{N_1+1,...,N_1+N_2\}$, respectively):

$$\begin{aligned} U_{\mathrm{req}}^t &= \sum_{i=1}^{N} w_i^t \psi_i\left(y_i^t\right) - \mu \sum_{i=1}^{N_1+N_2} \zeta_i\left(\mathbf{x}_i, \psi_i\left(y_i^t\right)\right) \\ &- \mu \sum_{j=1}^{N_3} \zeta_{\mathscr{A}_j}\left(\mathbf{x}_i, \psi_{\mathscr{A}_j}\left(\sum_{i\in\sum_{l\in\mathscr{A}_j}} y_l^t\right)\right) \\ &= F^1(\mathbf{x}_1,...,\mathbf{x}_N; y_1^t,...,y_N^t). \end{aligned}$$

where $\zeta_{\mathscr{A}_j}\left(\mathbf{x}_i, \psi_{\mathscr{A}_j}\left(\sum_{i\in\sum_{l\in\mathscr{A}_j}} y_l^t\right)\right)$ represents the sum feedback from $\mathscr{A}_j$. Then, the problem for the task requester is formulated by:

$$\begin{aligned} \max \quad & F^1(\mathbf{x}_1,...,\mathbf{x}_N; y_1^t,...,y_N^t) & (8)\\ \text{s.t.} \quad & y_i^t \in \mathbb{R}^*, x_{i,j} \in \mathbb{R}^*, x_{i,l-1} \le x_{i,l} & (9)\\ & i = 1,...,N, l = 1,...,m & (10) \end{aligned}$$

in which the task requester need to optimize his goal by specifying each $\mathbf{x}_1,...,\mathbf{x}_N$. We use Ω_{req} to represent the problem's feasible region (Equ. (9) and Equ. (10)).

Honest workers. The objective of each honest worker i is to maximize his compensation with low effort. Therefore, in each round t, by combining each honest worker i's compensation in the next round $t+1$ and effort levels in round t, we get their utility:

$$U^t_{\text{HU}\ i} = c_i^{t+1} - \beta y_i^t = \zeta_i\left(\mathbf{x}_i, \psi_i\left(y_i^t\right)\right) - \beta y_i^t = F_i^2(\mathbf{x}_i, y_i^t). \quad (11)$$

where $\beta > 0$ is the weight assigned to the effort level in the utility. We formulate the problem for each honest worker i by

$$\begin{aligned} \max \quad & F_i^2(\mathbf{x}_i, y_i^t) & (12) \\ \text{s.t.} \quad & y_i^t \in \mathbb{R}^* & (13) \end{aligned}$$

and use Ω_i^{hon} to represent the feasible region of the above problem (Equ. (16)).

Non-collusive malicious workers. Besides maximizing the compensation with less effort, non-collusive malicious workers also aim to increase the influence (i.e., feedback) of their reviews to deviate the true value of the products. Accordingly, in each round t, the utility of a non-collusive malicious worker i should be a combination of his compensation in the next round $t+1$, effort levels and feedback in round t:

$$\begin{aligned} & U^t_{\text{MU}\ i} = c_i^{t+1} - \beta y_i^t + \omega q_i^t \\ = \ & \zeta_i\left(\mathbf{x}_i, \psi_i\left(y_i^t\right)\right) - \beta y_i^t + \omega\psi_i\left(y_i^t\right) \\ = \ & F_i^3(\mathbf{x}_i, y_i^t). & (14) \end{aligned}$$

where $\omega > 0$ are the weight assigned to the feedback in the utility. We formulate the problem for each honest worker i by

$$\begin{aligned} \max \quad & F_i^3(\mathbf{x}_i, y_i^t) & (15) \\ \text{s.t.} \quad & y_i^t \in \mathbb{R}^* & (16) \end{aligned}$$

and use Ω_i^{ncm} to represent the feasible region of the above problem (Equ. (16)).

Collusive malicious workers. Collusive malicious workers have the same objective of non-collusive malicious workers. But different from non-collusive malicious workers, collusive malicious workers in the same community can share information and might upvote each other to increase their positive endorsements and therefore apparent contribution. Here, we design the same contract for the malicious workers in the same community, then the utility of the collusive malicious worker set $\mathscr{A}_j$ can be written by

$$\begin{aligned} U^t_{\text{MU}\ \mathscr{A}_j} \ = \ & \zeta_{\mathscr{A}_j}\left(\mathbf{x}_i, \psi_{\mathscr{A}_j}\left(\sum_{i\in\sum_{l\in\mathscr{A}_j}} y_l^t\right)\right) - \beta\sum_{i\in\mathscr{A}_j} y_i^t + \omega q^t_{\mathscr{A}_j} \\ = \ & F_j^3(\mathbf{x}_{\mathscr{A}_j}, y^t_{\mathscr{A}_j}). \end{aligned}$$

where $y^t_{\mathscr{A}_j} = \sum_{i\in\mathscr{A}_j} y_i^t$ and $\mathbf{x}_{\mathscr{A}_j} = \mathbf{x}_i, \ \forall i \in \mathscr{A}_j$. The problem formulated for each collusive community $\mathscr{A}_j$ is

$$\begin{aligned} \max \quad & F_j^3(\mathbf{x}_{\mathscr{A}_j}, y^t_{\mathscr{A}_j}) & (17) \\ \text{s.t.} \quad & y_i^t \in \mathbb{R}^*, \forall i \in \mathscr{A}_j. & (18) \end{aligned}$$

We use $\Omega^{\text{cm}}_{\mathscr{A}_j}$ to represent the problem's feasible region (Equ. (18)).

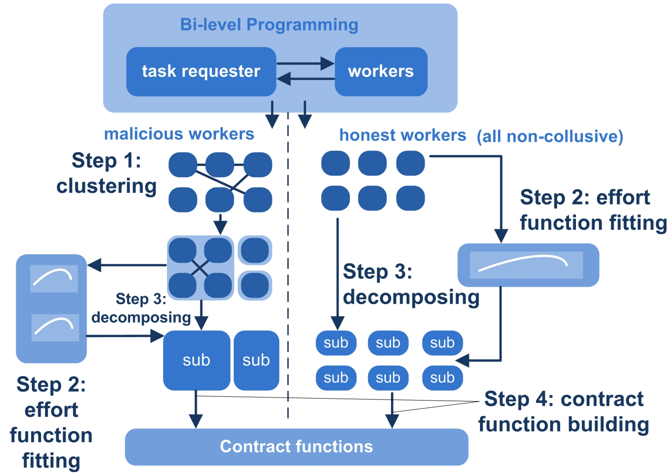


Fig. 4. Strategy framework

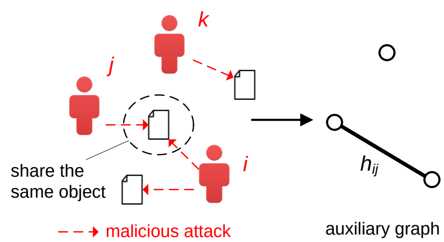


Fig. 5. The auxiliary graph of the collusive workers.

Consequently, we can formulate the Bi-MIP problem as follows:

$$\begin{aligned} \max \quad & F^1(\mathbf{x}_1, \ldots, \mathbf{x}_N; y_1^t, \ldots, y_N^t) \\ \text{s.t.} \quad & (\mathbf{x}_1, \ldots, \mathbf{x}_N; y_1^t, \ldots, y_N^t) \in \Omega_{\text{req}} \\ & y_i^t = \arg\max_{y' \in \Omega_i^{\text{hon}}} F_i^2(\mathbf{x}_i, y') \ \ i = 1, \ldots, N_1 \\ & y_i^t = \arg\max_{y' \in \Omega_i^{\text{ncm}}} F_i^2(\mathbf{x}_i, y') \ \ i = N_1+1, \ldots, N_1+N_2 \\ & \mathbf{y}^t_{\mathscr{A}_j} = \arg\max_{\mathbf{y}' \in \Omega^{\text{cm}}_{\mathscr{A}_j}} F_j^3(\mathbf{x}_{\mathscr{A}_j}, \mathbf{y}') \quad j = 1, \ldots, N_3 \end{aligned}$$

IV. Methodology

In this section, we present our methodology to solve the BiP formulated in the last section. Fig. 4 shows the framework of our strategy: We first cluster collusive malicious workers based on their targeted products (Section IV-A), and obtain the effort function for each worker or collusive community by fitting the function with real data (Section IV-B). After that, we decompose BiP into a set of subproblems, where each subproblem only deals with the contract design for a single worker or a single cluster. Finally, we solve each subproblem with a time-efficient algorithm with theoretic bound (Section IV-C).

A. Collusive workers clustering

Recall that we assume that two malicious workers are collusive if they target the same product. We define a group of malicious workers $\mathscr{A}$ as a collusive community if, for each worker $i \in \mathscr{A}$, exists another worker $j \in \mathscr{A}$ that writes a review for the same product as the one reviewed by worker i. Accordingly, we can create a auxiliary graph $\mathscr{G} = (\mathscr{U}, \mathscr{H})$, where $\mathscr{H}$ denotes the set of virtual edges and a virtual edge $h_{i,j}$, is a connection between any pair of workers i and j who share the same target (as Fig. 5 shows). Then, finding collusive communities is equivalent to finding the connected components in $\mathscr{G}$. In particular, we apply depth-first search (DFS) to find each connected component [18]. Based on the results of clustering, we are able to distinguish collusive workers and non-collusive workers with a given probability (see Section V for some empirical evidence supporting this approach).

B. Effort function fitting&Problem decomposition

In this part, we aim to construct an effort function that has the best fit to a given set of data points composed of workers' effort levels and feedbacks. To this end, we fit real review data from 18,176 non-collusive honest workers, 1,312 non-collusive malicious workers and 212 collusive malicious workers in Amazon (we will introduce the data set later in Section V) to the polynomial functions with different orders and compare their norm of residual (NoR) in Table III, which is a measure of the deviation between the correlation and the data, i.e., a lower norm signifies a better fit. By comparing the NoR and the complexities of different fitting curves, we finally choose quadratic functions as effort functions

$$\psi_i(y_i^t) = r_2 y_i^2 + r_1 y_i^1 + r_0. \tag{19}$$

As we already estimated all the collusive worker communities using the clustering algorithm and the effort functions for all the workers by fitting the function to real data, now we decompose the BiP problem into a set of subproblems. First, the objective function of the requester (Equ.(17)) can be written as:

$$F^1(\mathbf{x}_1, ..., \mathbf{x}_N; y_1^t, ..., y_N^t) = \sum_{i=1}^{N_1} F_i^{1,1}\left(\mathbf{x}_i, y_i^t\right) + \sum_{i=j}^{N_2} F_{\mathscr{A}_j}^{1,2}\left(\mathbf{x}_{\mathscr{A}_j}, \mathbf{y}_{\mathscr{A}_j}^t\right)$$

where $F_i^{1,1}(\mathbf{x}_i, y_i^t) = w_i^t \psi_i(y_i^t) - \mu \zeta_i(\mathbf{x}_i, \psi_i(y_i^t))$ and

$$\begin{aligned} F_{\mathscr{A}_j}^{1,2}\left(\mathbf{x}_{\mathscr{A}_j}, \mathbf{y}_{\mathscr{A}_j}^t\right) &= w_{\mathscr{A}_j}^t \psi_{\mathscr{A}_j}\left(\sum_{l \in \mathscr{A}_j} y_l^t\right) \\ &- \mu \zeta_{\mathscr{A}_j}\left(\mathbf{x}_{\mathscr{A}_j}, \psi_{\mathscr{A}_j}\left(\sum_{l \in \mathscr{A}_j} y_l^t\right)\right). \end{aligned} \tag{20}$$

Here, each $F_i^{1,1}(\mathbf{x}_i, y_i^t)$ and $F_{\mathscr{A}_j}^{1,2}\left(\mathbf{x}_{\mathscr{A}_j}, \mathbf{y}_{\mathscr{A}_j}^t\right)$ respectively represent the utilities that the requester obtains from a non-collusive worker i and a collusive worker set $\mathscr{A}_j$. Furthermore, there is no correlation between the utilities of any pair of non-collusive workers, or any pair of collusive communities, or any non-collusive worker and collusive community. Consequently, we can decompose the Bi-MIP into a group of independent subproblems sub_i $(i = 1, ..., N_1)$ for each honest worker i:

$$\begin{aligned} \max \quad & F_i^{1,1}(\mathbf{x}_i, y_i^t) && (21) \\ \text{s.t.} \quad & (\mathbf{x}_i, y_i^t) \in \Omega_i^{\text{req}}, && (22) \\ & y_i^t = \arg \max_{y' \in \Omega_i^{\text{hon}}} F_i^2(\mathbf{x}_i, y') && (23) \end{aligned}$$

sub'_i $(i = N_1 + 1, ..., N_1 + N_2)$ for each non-collusive malicious worker i:

$$\begin{aligned} \max \quad & F_i^{1,1}(\mathbf{x}_i, y_i^t) && (24) \\ \text{s.t.} \quad & (\mathbf{x}_i, y_i^t) \in \Omega_i^{\text{req}}, && (25) \\ & y_i^t = \arg \max_{y' \in \Omega_i^{\text{ncm}}} F_i^3(\mathbf{x}_i, y') && (26) \end{aligned}$$

and sub''_j $(j = 1, ..., N_3)$ for each collusive community $\mathscr{A}_j$

$$\begin{aligned} \max \quad & F_{\mathscr{A}_j}^{1,2}\left(\mathbf{x}_{\mathscr{A}_j}, \mathbf{y}_{\mathscr{A}_j}^t\right) && (27) \\ \text{s.t.} \quad & (\mathbf{x}_{\mathscr{A}_j}, \mathbf{y}_{\mathscr{A}_j}^t) \in \Omega_{\mathscr{A}_j}^{\text{req}}, && (28) \\ & \mathbf{y}_{\mathscr{A}_j}^t = \arg \max_{\mathbf{y}' \in \Omega_{\mathscr{A}_j}^{\text{cm}}} F_j^3(\mathbf{x}_{\mathscr{A}_j}, \mathbf{y}') && (29) \end{aligned}$$

where $\Omega_i^{\text{req}} = \{(\mathbf{x}_i, y_i^t) : y_i^t \in \mathbb{R}^*, x_{i,j} \in \mathbb{R}^*, x_{i,l-1} \leq x_{i,l}\}$ and $\Omega_{\mathscr{A}_j}^{\text{req}} = \{(\mathbf{x}_{\mathscr{A}_j}, \mathbf{y}_{\mathscr{A}_j}^t) : y_i^t \in \mathbb{R}^*, x_{i,j} \in \mathbb{R}^*, x_{i,l-1} \leq x_{i,l},\ \forall i \in \mathscr{A}_j\}$.

C. Contract function building

Even though each subproblem has much smaller scale compared to the original BiP problem, we cannot directly apply the existing optimization approaches (e.g., subgradient methods or interior point methods [19]) due to the non-convexity of the problem. As a solution, we devise a time-efficient algorithm that can achieve near optimal solution. The idea is motivated by the observation that each worker's utility is *twice differentiable* within any effort interval $[(l-1)\delta, l\delta)$ $(l = 1, ..., m)$, indicating that the worker's maximum utility within $[(l-1)\delta, l\delta)$ can be derived given the contract piece $[x_{i,l-1}, x_{i,l})$. Therefore, in the first step, we build a contract function for each effort interval $[(l-1)\delta, l\delta)$, called *candidate contract*, such that the maximum utility in $[(l-1)\delta, l\delta)$ is higher than the maximum utility in any other effort intervals. It implies that, under the candidate contract of $[(l-1)\delta, l\delta)$, workers will always select the effort interval in $[(l-1)\delta, l\delta)$ in order to maximize their utility. In addition, when building each candidate contract function, we also try to minimize the contract increment (or slope) for each contract piece in order to maximize the requester's utility.

After building the candidate contracts for the effort intervals, we select the candidate contract that maximizes the requester's utility as our final solution. Note that there must exist a candidate contract under which the optimal effort level falls in the same interval with that of the optimal contract, indicating that this candidate contract is close to the optimal contract as we try to maximize the requester's utility when building it. Then, our final contract is close to the optimal contract since the final contract is no worse than any candidate contracts. In what follows, we will introduce the details of our algorithm, and we will formally analyze how we approach the optimal solution in Theorem 4.1. Here, due to the lack of space, we only analyze the case of non-collusive malicious workers. Comparing the utility of honest workers (Equ. (11)) and non-collusive malicious workers (Equ. (14)), we can consider honest workers as a special case of non-collusive malicious workers by setting $\omega = 0$, where ω is the weight assigned to feedback in non-collusive malicious workers' utility. Hence, the construction of the contract for honest workers follows the same procedure with that of non-collusive malicious workers. The calculation of contracts for both non-collusive malicious workers and collusive malicious workers are also same since a collusive community can be treated as a "single meta-worker".

As we already estimated the effort function by fitting the function to real data, we use $\xi_i(y_i^t) = f_i(\psi_i(y_i^t))$ to represent the mapping from worker i's effort level to his compensation, and use $\xi_i^{(k)}$ to denote the mapping for the candidate contract of $[(k-1)\delta, k\delta)$. In $\xi_i^{(k)}$, let $x_{i,l}^{(k)}$ denote the compensation when the effort is $l\delta$, let $\Delta x_{i,l}^{(k)} = x_{i,l}^{(k)} - x_{i,l-1}^{(k)}$, and let $\alpha_{i,l}^{(k)} = \Delta x_{i,l}^{(k)} / \Delta d_{i,l}$.

The procedure to obtain $\xi_i^{(k)}$ is composed of two parts:

*P*1. For each effort interval $[(l-1)\delta, l\delta)$, analyze the relationship between its contract piece $(x_{i,l-1}^{(k)}, x_{i,l}^{(k)})$ and the worker's maximum utility in the interval.

*P*2. Construct the contract function $\xi_i^{(k)}$ to satisfy the two principles 1) worker i's optimal effort level falls in $[(k-1)\delta, k\delta)$ and 2) the requester's utility is maximized when worker i choose the optimal effort level.

Part 1. We use $y_{i,l}^t$ to denote the effort level that maximizes worker i's utility in effort interval $[(l-1)\delta, l\delta)$:

$$y_{i,l}^t = \arg \max_{y_i^t \in [(l-1)\delta, l\delta)} F_i^3(\mathbf{x}_i, y_i^t). \tag{30}$$

To derive $y_{i,l}^t$, we consider the following three cases:

Case I: $\frac{\partial F_i^3(\mathbf{x}_i, y_i^t)}{\partial y_i^t} \leq 0$, $\forall y_i^t \in [(l-1)\delta, l\delta)$. $F_i^3(\mathbf{x}_i, y_i^t)$ is non-increasing when $y_i^t \in [(l-1)\delta, l\delta)$. Then, $y_{i,l}^t = (l-1)\delta$.

Case II: $\frac{\partial F_i^3(\mathbf{x}_i, y_i^t)}{\partial y_i^t} \geq 0$, $\forall y_i^t \in [(l-1)\delta, l\delta)$. $F_i^3(\mathbf{x}_i, y_i^t)$ is non-decreasing when $y_i^t \in [(l-1)\delta, l\delta)$. Then, $y_{i,l}^t = l\delta$.

Case III: $\exists y \in ((l-1)\delta, l\delta)$ *s.t.* $\frac{\partial F_i^3(\mathbf{x}_i, y_i^t)}{\partial y_i^t}|_{y_i^t = y} = 0$. Since $\psi_i(y_i^t)$ is concave and twice differentiable, we can derive that $\frac{\partial^2 F_i^3(\mathbf{x}_i, y_i^t)}{\partial y_i^{t2}} = \left(\alpha_{i,l}^{(k)} - \beta\right) \psi_i''(y_i^t) \leq 0$, which implies that worker i's utility is maximum at y and hence $y_{i,l}^t = y$. As $\psi_i'(y_i^t)$ is monotonically decreasing, $\psi_i'(y_i^t)$ has its *inverse function* $\psi_i'^{-1}(y_i^t)$. Consequently, we have $y_{i,l}^t = \psi_i'^{-1}\left(\beta / \alpha_{i,l}^{(k)}\right)$. Particularly, if $\psi_i(y_i^t)$ is a quadratic function: $\psi_i(y_i^t) = r_2 y_i^{t2} + r_1 y_i^t + r_0$, then

$$y_{i,l}^t = \frac{\beta}{2r_2\left(\alpha_{i,l}^{(k)} + \omega\right)} - \frac{r_1}{2r_2}. \tag{31}$$

Lemma 4.1 gives necessary and sufficient conditions for the three cases:

Lemma 4.1: A contract piece on $[(l-1)\delta, l\delta)$ is in *Case I* if $\alpha_{i,l}^{(k)} \in \left(-\infty, \frac{\beta}{2r_2(l-1)\delta + r_1} - \omega\right]$, is in *Case II* if $\alpha_{i,l}^{(k)} \in \left[\frac{\beta}{2r_2 l\delta + r_1} - \omega, \infty\right)$, and is in *Case III* if $\alpha_{i,l}^{(k)} \in \left(\frac{\beta}{2r_2(l-1)\delta + r_1} - \omega, \frac{\beta}{2r_2 l\delta + r_1} - \omega\right)$.

Proof *Case I*: As $\frac{\partial F_i^3(\mathbf{x}_i, y_i^t)}{\partial y_i^t}$ is monotonically decreasing, to guarantee $\frac{\partial F_i^3(\mathbf{x}_i, y_i^t)}{\partial y_i^t} \leq 0$ when $y_i^t \in [(l-1)\delta, l\delta)$, we only need to make sure

$$\frac{\partial F_i^3(\mathbf{x}_i, y_i^t)}{\partial y_i^t}|_{y_i^t = l\delta} \leq 0 \ \Rightarrow\ \alpha_{i,l}^{(k)} \geq \frac{\beta}{2r_2(l-1)\delta + r_1} - \omega. \tag{32}$$

Case II: Similarly, to make $\frac{\partial F_i^3(\mathbf{x}_i, y_i^t)}{\partial y_i^t} \geq 0$ when $y_i^t \in [(l-1)\delta, l\delta)$, we need

$$\frac{\partial F_i^3(\mathbf{x}_i, y_i^t)}{\partial y_i^t}|_{y_i^t = (l-1)\delta} \geq 0 \ \Rightarrow\ \alpha_{i,l}^{(k)} \leq \frac{\beta}{2r_2 l\delta + r_1} - \omega. \tag{33}$$

Case III: Finally, to guarantee that

$$\exists y \in [(l-1)\delta, l\delta) \text{ s.t. } \frac{\partial F_i^3(\mathbf{x}_i, y_i^t)}{\partial y_i^t}|_{y_i^t = y} = 0, \tag{34}$$

we need to make sure $\frac{\partial F_i^3(\mathbf{x}_i, y_i^t)}{\partial y_i^t}|_{y_i^t = (l-1)\delta} > 0$ and $\frac{\partial F_i^3(\mathbf{x}_i, y_i^t)}{\partial y_i^t}|_{y_i^t = l\delta} < 0$. which implies that

$$\frac{\beta}{2r_2(l-1)\delta + r_1} - \omega < \alpha_{i,l}^{(k)} < \frac{\beta}{2r_2 l\delta + r_1} - \omega. \tag{35}$$

Part 2. Construct each candidate contract $\xi_i^{(k)}$:

We build $\xi_i^{(k)}$ by iteratively constructing contract pieces with the increasing order of effort level intervals $[0, \delta)$, ..., $[(m-1)\delta, m\delta)$. To make sure that worker i's optimal effort level falls in $[(k-1)\delta, k\delta)$, when constructing $\xi_i^{(k)}$ on interval $[(l-1)\delta, l\delta)$ in iteration l, we need to satisfy the following equations:

$$\begin{cases} F_i^3(\mathbf{x}_i, y_{i,l}^t) - F_i^3(\mathbf{x}_i, y_{i,l-1}^t) > 0, & \text{when } l \leq k \\ F_i^3(\mathbf{x}_i, y_{i,l-1}^t) - F_i^3(\mathbf{x}_i, y_{i,l}^t) > 0, & \text{when } l \geq k+1 \end{cases} \tag{36}$$

That is, when worker i's effort level falls on the left side (right side) of $[(k-1)\delta, k\delta)$, worker i tends to increase (decrease) the effort level to increase his utility.

On the other hand, to maximize the utility of the requester, when building each contract piece, we need to find the minimum compensation slope satisfying Equ. (36).

According to the analysis above, determining the contract pieces defined on $[k\delta, \infty)$ is trivial: Let $x_l = x_{l-1}$ for each $l = k+1, ..., m$, then worker i's utility with effort level in $[k\delta, \infty)$ is always lower than that in $[(k-1)\delta, k\delta)$, since his effort level in $[k\delta, \infty)$ is higher than in $[(k-1)\delta, k\delta)$, but with the same compensation.

Now, we turn our attention to the contract pieces defined in $[0, k\delta)$. The basic idea is to reduce the contract slope $\alpha_{i,l}^{(k)}$ for each contract piece while satisfying Equ. (36). As we analyze in step 1, there are three cases to consider: For Case I, worker

i always choose the lowest effort level in each effort interval. As we aim to incentivise workers to increase their effort level when it is in $[0, k\delta)$, we try to avoid building contract piece in Case I. As for Case II, Lemma 4.1 has proved that constructing a contract piece in case II always has higher contract slope than in Case III.

Accordingly, to reduce contract slopes, we try to build each contract piece in Case III, i.e., is to find each $\alpha_{i,l}^{(k)} \in \left(\frac{\beta}{2r_2(l-1)\delta+r_1} - \omega, \frac{\beta}{2r_2 l\delta+r_1} - \omega\right)$ such that Equ. (36) is satisfied. By plugging Equ. (31) (worker i's optimal utility in Case III) into Equ. (36), we obtain

$$
\begin{aligned}
& F_i^3(\mathbf{x}_i, y_{i,l}^t) - F_i^3(\mathbf{x}_i, y_{i,l-1}^t) \\
= & \left(\frac{\beta^2}{4r_2} \frac{1}{\left(\alpha_{i,l}^{(k)} + \omega\right)\left(\alpha_{i,l-1}^{(k)} + \omega\right)} + r_0 - \frac{r_1^2}{4r_2} - d_{i,l-1} \right) \\
\times & \left(\alpha_{i,l}^{(k)} - \alpha_{i,l-1}^{(k)}\right)
\end{aligned} \tag{37}
$$

According to Lemma 4.1, $\alpha_{i,l}^{(k)} > \frac{\beta}{2r_2(l-1)\delta+r_1} - \omega$ and $\alpha_{i,l-1}^{(k)} < \frac{\beta}{2r_2(l-1)\delta+r_1} - \omega$, indicating that $\alpha_{i,l}^{(k)} - \alpha_{i,l-1}^{(k)} > 0$. Hence, to meet Equ. (31), we need to guarantee

(38)

from which we derive that $\alpha_{i,l}^{(k)}$ needs to be larger than $\frac{\beta^2}{\left(\alpha_{i,l-1}^{(k)}+\omega\right)(r_1+2r_2\delta(l-1))^2}$. Then, we set the slope of the contract piece on $[(l-1)\delta, l\delta)$ by

$$
\alpha_{i,l}^{(k)} = \beta^2 / \left(\left(\alpha_{i,l-1}^{(k)} + \omega\right)(r_1 + 2r_2\delta(l-1))^2 \right) + \varepsilon_{i,l}^{(k)} - \omega, \tag{39}
$$

where $\varepsilon_{i,l}^{(k)}$ is a small positive real number given by

$$
\varepsilon_{i,l}^{(k)} = 4\beta r_2^2 \delta^2 \left((r_1 + 2r_2\delta(l-1))^2 (r_1 + 2r_2\delta l) \right). \tag{40}
$$

The following two equations show that $\alpha_{i,l}^{(k)}$ given by Equ. (39) is always in $\left(\frac{\beta}{2r_2(l-1)\delta+r_1}, \frac{\beta}{2r_2 l\delta+r_1}\right)$:

$$
\begin{aligned}
& \alpha_{i,l}^{(k)} - \left(\frac{\beta}{2r_2(l-1)\delta + r_1} - \omega\right) \\
= & \frac{\beta^2}{\frac{\Delta x_{i,l-1}^{(k)}}{\Delta d_{i,l-1}^{(k)}} (r_1 + 2r_2\delta(l-1))^2} + \varepsilon_{i,l}^{(k)} - \frac{\beta}{2r_2(l-1)\delta + r_1} \\
\geq & \frac{\beta}{r_1 + 2r_2\delta(l-1)} + \varepsilon_{i,l}^{(k)} - \frac{\beta}{2r_2(l-1)\delta + r_1} > 0
\end{aligned} \tag{41}
$$

$$
\begin{aligned}
& \alpha_{i,l}^{(k)} - \left(\frac{\beta}{2r_2 l\delta + r_1} - \omega\right) \\
= & \frac{\beta^2}{\frac{\Delta x_{i,l-1}^{(k)}}{\Delta d_{i,l-1}^{(k)}} (r_1 + 2r_2\delta(l-1))^2} + \varepsilon_{i,l}^{(k)} - \frac{\beta}{2r_2 l\delta + r_1} \\
< & \frac{\beta(r_1 + 2r_2\delta(l-2))}{(r_1 + 2r_2\delta(l-1))^2} + \varepsilon_{i,l}^{(k)} - \frac{\beta}{2r_2 l\delta + r_1} \\
= & \frac{-4\beta r_2^2\delta^2}{(r_1 + 2r_2\delta(l-1))^2 (r_1 + 2r_2\delta l)} + \varepsilon_{i,l}^{(k)} = 0
\end{aligned} \tag{42}
$$

Lemma 4.2: The compensation paid to each worker in $\xi_i^{(k)}$ is upper bounded by $\frac{-2\beta r_2 k\delta^2}{2r_2(k-1)\delta+r_1} + \beta k\delta$.

Lemma 4.3: Let $c_{i,\min}^{t,(k)}$ be the minimum compensation paid to worker i given that worker i's optimal effort level falls in $[(k-1)\delta, k\delta)$. Then, $c_{i,\min}^{t,(k)}$ is lower bounded by $\beta(k-1)\delta$.

The proofs for both the above Lemmas are reported in Appendix for lack of space.

After collecting all the candidate contract functions $\xi_i^{(1)}$, ..., $\xi_i^{(m)}$, we pick up the one in which the requester has the highest utility when the worker selects the optimal effort level, denoted by $\xi_i^{k_{\text{opt}}}$, where

$$
k_{\text{opt}} = \arg\max_k \left(\xi_i^{(k)}(y_{i,k}^t) - \beta y_{i,k}^t + \omega \psi_i(y_{i,k}^t) \right), \tag{43}
$$

Consequently, we can derive the contract f_i from ξ_i by

$$
f_i(d_{i,l}) = \xi_i^{k_{\text{opt}}}(l\delta), \quad l = 1, ..., m. \tag{44}
$$

Theorem 4.1 provides the upper bound and lower bound of the requester's utility using the contract calculated by our algorithm.

Theorem 4.1: The utility that the requester obtain from worker i is upper bounded by $\max_l\{\psi_i(l\delta) - \mu(l-1)\delta\}$ and lower bounded by $\psi_i((k_{\text{opt}}-1)\delta) + \frac{2\beta r_2 k_{\text{opt}}\delta^2}{2r_2(k_{\text{opt}}-1)\delta+r_1} - \mu k_{\text{opt}}\delta$.

Proof 1) *Upper bound*: In the case that worker i's optimal effort level falls in $[(l-1)\delta, l\delta)$, according to Lemma 4.3, the compensation paid to worker i is lower bounded by $\beta(l-1)\delta$, which implies that the utility gained from worker i is upper bounded by $\psi_i(l\delta) - \mu(l-1)\delta$. Finally, the upper bound of the requester's utility is derived from the maximum value of maximum utility of all the case, i.e., $\max_l\{\psi_i(l\delta) - \mu(l-1)\delta\}$.
1) *Lower bound*: According to Lemma 4.2, using the contract designed by our algorithm, the requester's utility

$$
\begin{aligned}
U_{\text{req}}^t &= \psi_i(y_{i,k_{\text{opt}}}^t) - \xi_i^{k_{\text{opt}}}(y_{i,k_{\text{opt}}}^t) \qquad (45) \\
&\geq \psi_i((k_{\text{opt}}-1)\delta) + \frac{2\beta r_2 k_{\text{opt}}\delta^2}{2r_2(k_{\text{opt}}-1)\delta + r_1} - \mu k_{\text{opt}}\delta.
\end{aligned}
$$

In Fig. 6, we depict the utility of the requester using our designed contract for a single honest worker, as well as the lower bound and upper bound derived by Theorem 4.1. For the parameters in workers and the requester's utility, we set $\mu = 10$, $\beta = \alpha = 1$, and $\kappa = \gamma = 0.1$ and still use this setting in our performance evaluation part (Section V). From the figure, we find that the utility got from our method get closer to the upper bound as the number of intervals in the effort region increases. Note that the optimal utility must be within the gap between our calculated utility and its upper bound. Hence, we can conclude the requester's utility converge to the optimal as the partition in effort region goes denser.

V. Performance Evaluation

Our empirical analysis is based on a dataset of crowd-sourced product reviews from Amazon.com. Like most online retailers, Amazon relies on consumers to provide feedback on purchased items in the form of posted reviews. Reviews and

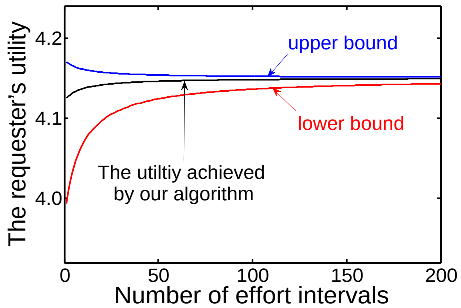


Fig. 6. Numeric result of the requester's upper bound and lower bound.

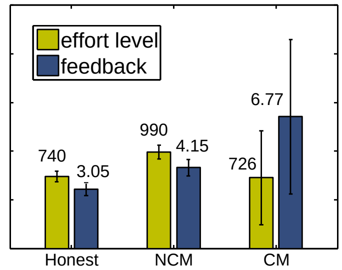


Fig. 7. Comparison of the three types of workers.

reviewers are endowed with additional information, including possible designation as an expert reviewer, whether the reviewer's purchase was "verified", and the number of other customers who found the review helpful.

The notion driving this process is that real customers will volunteer their time to write a short review of a product in service to the system that in turn helps them in making future purchases. However, the importance of reviews for product sales brings with it the motivation and potential for malicious behavior. Product representatives may opt to hire workers to generate positive reviews of products they have not necessarily purchased or tried [20]. These reviews bring up the average number of stars and search-engine visibility of these products. If malicious reviews are well-written and deemed "helpful" by others, they will rise to the top of the page and garner even further visibility.

Dataset. We consider an Amazon dataset [13] containing 118,142 product reviews generated by 19,686 reviewers for 75,508 different products [3]. The dataset includes ground truth labels for 1,524 malicious reviewers obtained by crawling underground Internet sites which recruit workers to write biased reviews on Amazon. Reviewed items include electronics, books, beauty products and medications. This dataset was previously studied in the context of detecting fake product reviews; here we leverage that work to serve as a real-world trace of the presence of malicious behavior in reponses to a crowdsourced task.We parametrize our model as follows:

1) *Feedback* of a review: the number of positive upvotes (i.e., voted as "helpful") awarded by other workers.

2) *Expertise* of a worker: the average feedback (upvotes) over all reviews written by that worker;

3) *Length* of a review: the number of characters included in the review;

4) *Effort level* for a review: the product of the worker's expertise and the length of the review.

Collusive worker clustering. We first identify the collusive communities among malicious workers using our collusive worker clustering algorithm. We note that our approach for detecting clusters of malicious workers appears effective, as discussed next. We identify a total of 212 collusive workers in 47 collusive communities, where Table II lists the distribution of the size of the collusive communities. After clustering the malicious workers, we compare the average effort levels and average feedback for honest workers, non-collusive malicious (NCM) workers and collusive malicious (CM) workers in Fig. 7. From the figure, we observe that though the effort levels of the three types of workers are similar, collusive malicious workers have much higher feedback than the other two types. We suggest that this effect is a result of malicious workers in the same collusive community upvoting each others' reviews.

TABLE II
DISTRIBUTION OF COLLUSIVE COMMUNITY SIZE

Size	2	3	4	5	6	≥10
Percentage (%)	51.2	22.0	7.3	2.4	9.8	4.9

Effort function fitting. We fit the effort function of three types of workers respectively. For honest workers, we use 18176 data points (including feedback and effort level) from honest workers to fit the honest workers' effort function, and use 1312 data points and 212 data points to fit the effort function of non-collusive malicious workers and collusive malicious workers, respectively. Table III lists the NoRs of different fitting functions, which measure the deviation between the correlation and the data. From Table III, we observe that the norm of residual of all fitting curves are close. Considering the complexity of the functions, we choose quadratic functions as the effort functions of workers.

TABLE III
COMPARISON OF NOR FOR DIFFERENT FITTING FUNCTIONS

	linear	quad	cubic	4th	5th	6th
Honest workers	13.8	13.7	13.7	13.7	13.7	13.7
NC-Mal workers	2.60	2.60	2.60	2.59	2.59	2.59
C-Mal workers	11.3	11.3	11.3	11.3	11.3	11.3

Performance of our contract design algorithm. For performance evaluation, we first select 200 honest workers (those who have at least 20 reviews in history) and draw their paid compensation and the compensation's lower bound (derived by Lemma 4.3) in Fig. 8(a), with the number of intervals in effort region m equals to 10, 20, and 40, respectively. By comparing the three figures in Fig. 8(a), we find that the gap between the compensation and its lower bound becomes

[3]our dataset, although used by [13] to detect malicious reviewers has not been studied for the purpose of pricing strategy.

smaller as the number of intervals increases, which indicates the compensation converges to the optimal as the partition in workers' effort region goes denser.

We also compare the average, 5th percentile, and 95th percentile compensation paid to all the honest workers, non-collusive malicious workers, and collusive malicious workers in Fig. 8(b), with $\mu = 1.0$, 0.9 and 0.8. Recall that μ is the weight assigned to the compensation paid to workers in the requester's utility. From the figure, we have two observations: (1) the workers' compensation increases as μ decreases and (2) the compensation paid to the three types of workers follows: honest workers > non-collusive malicious workers > collusive malicious workers. For observation (1), as μ is the weight given to the compensation paid to workers, a lower μ indicates a "generous" requester, who cares more about the benefit brought by workers' feedback, but less about the cost paid to workers. As for observation (2), recall that, in the requester's utility, we have a penalty for the weight of feedback (defined by Equ. (5)) from non-collusive malicious workers (penalty is $\kappa e_i^{\mathrm{mal}}$) and collusive malicious workers (penalty is $\kappa e_i^{\mathrm{mal}} + \gamma A_i$). Considering the penalties, the requester will value the feedback from collusive and non-collusive workers less compared to honest workers when designing the contract, which consequently leads to less compensation paid to collusive and non-collusive malicious workers.

In addition, as shown in Fig. (8)(c), we compare the requester's utility of our contract design and a baseline approach, in which all the malicious workers are simply excluded from the system. As shown, our contract design outperforms the baseline method. This is because that, when assigning weights to workers' feedback, our strategy can take advantage of some of the malicious workers, specifically those who might be biased but are still accurate within a certain acceptable range. As for the malicious workers whose reviews are too distant from the average reviews by experts, the weight of their feedback will be close to 0, and hence will be automatically eliminated by our method.

VI. Related Work

Many current crowdsourcing platforms offer a limited capability to the requester in designing the pricing policies, mostly limiting them to a single fixed price [1], [2], [21]. These approaches do not take into account the change of workers' behaviors, and hence cannot incentivise a high-quality work from workers. Recently, more efforts have been devoted to studying dynamic pricing strategies, in which the worker's compensation can be adjusted based on their performance for tasks [3]–[10]. For example, Thanh *et al.* [7] formulated the dynamic pricing problem as a multi-armed bandit (MAB) problem, and applied bandit algorithms to maximize the number of tasks. Zhang *et al.* [5] studied three crowdsourcing models, involving cooperation and competition among the service providers.

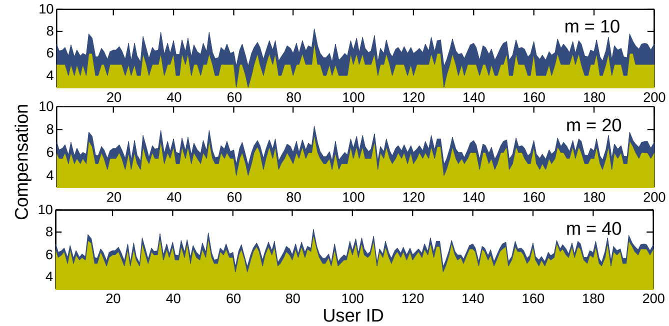
Singer [8] has initiated a budget feasibility framework, of which the objective is to design incentive compatible mechanisms to maximize a requesters utility under a budget. The framework has been adapted to different settings [4], [5], [9], [10]. For example, Zhang *et al.* [4] focused on incentivizing crowd workers to label a set of binary tasks under strict budget constraint. Singer and Mittal [5] presented constant-competitive incentive compatible mechanisms to maximize the number of tasks with a budget limit, and to minimize payments given a fixed number of tasks to complete. Ho *et al.* [22] formalized an online task assignment problem with budget limit, and presented a framework for matching workers with requesters based on the workers' expertise.

Though the above dynamic pricing strategies have their own merits, all of them rely on the two assumptions: 1) the worker effort level is directly observable, and 2) workers are homogenous with the same objective and strategy. In contrast to the existing works, our work completely removes both assumptions.

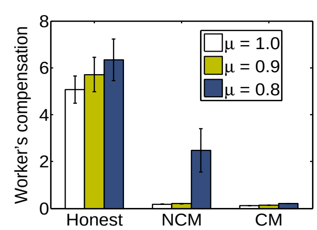
VII. Conclusion

In this paper, we presented a dynamic contract strategy to incentivise high-quality work from heterogeneous workers. Both theoretic analysis and experimental results prove that our contract design achieves a near optimal solution. Our experiments prove that our approach can incentivise high-quality honest workers while prevent pollution from malicious workers.

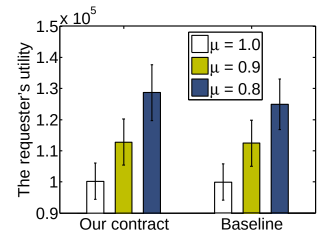
Next, we plan to account for more sophisticated malicious workers or collusive malicious workers, and will study how the contract functions are supposed to deal with these behaviors. We also plan to extend our model from review tasks to a more general case, which can be applied to different crowdsoucing applications, like classification.



(a) Comparison of the compensation paid to 200 workers in our strategy (dark blue) and the compensation's lower bound (green).



(b) Compensation comparison of different types of workers



(c) Utility comparison of our contract design and a baseline method

Fig. 8. Real data analysis from the Amazon review trace.

References

[1] J. J. Horton and L. B. Chilton. The labor economics of paid crowdsourcing. In *Proc. of EC*, 2010.

[2] Long Tran-Thanh, Trung Dong Huynh, Avi Rosenfeld, Sarvapali Ramchurn, and Nicholas R. Jennings. Budgetfix: Budget limited crowdsourcing for interdependent task allocation with quality guarantees. In *Proc. of AAMAS*, 2014.

[3] Yanjiao Chen, Baochun Li, and Qian Zhang. Incentivizing crowdsourcing systems with network effects. In *Infocom*, 2016.

[4] Qi Zhang, Yutian Wen, Xiaohua Tian, Xiaoying Gan, and Xinbing Wang. Incentivize crowd labeling under budget constraint. In *Infocom*, 2015.

[5] Yaron Singer and Manas Mittal. Pricing mechanisms in crowdsourcing markets. In *Proc. of WWW*, 2013.

[6] Xiang Zhang, Guoliang Xue, Ruozhou Yu, Dejun Yang, and Jian Tang. Truthful incentive mechanisms for crowdsourcing. In *Infocom*, 2015.

[7] L. Tran-Thanh, S. Stein, A. Rogers, and N. R. Jennings. Efficient crowdsourcing of unknown experts using multi-armed bandits. *Frontiers in Artificial Intelligence and Applications*, 2012.

[8] Y. Singer. Budget feasible mechanisms. In *Proc. of FOCS*, 2010.

[9] Y. Singer. How to win friends and influence people, truthfully: Influence maximization mechanisms for social networks. In *Proc. of WSDM*, 2012.

[10] S. Dobzinski, C. Papadimitriou, and Y. Singer. Mechanisms for complement-free procurement. In *Proc. of EC*, 2011.

[11] Yibo Wu, Yi Wang, Wenjie Hu, Xiaomei Zhang, , and Guohong Cao. Resource-aware photo crowdsourcing through disruption tolerant networks. In *Proc. of ICDCS*, 2016.

[12] Ioannis Boutsis and Vana Kalogeraki. On task assignment for real-time reliable crowdsourcing. In *Proc. of ICDCS*, 2015.

[13] Amir Fayazi, Kyumin Lee, James Caverlee, and Anna Squicciarini. Uncovering crowdsourced manipulation of online reviews. In *Proc. of SIGIR*. ACM, 2015.

[14] Tianyi Wang, Gang Wang, Xing Li, Haitao Zheng, and Ben Y. Zhao. Characterizing and detecting malicious crowdsourcing. In *Proc. of Sigcomm*, 2013.

[15] Gang Wang, Tianyi Wang, Haitao Zheng, and Ben Y. Zhao. Man vs. machine: Practical adversarial detection of malicious crowdsourcing workers. In *Proc. of Usenix Security*, 2014.

[16] John Le, Andy Edmonds, Vaughn Hester, and Lukas Biewald. Ensuring quality in crowdsourced search relevance evaluation: The effects of training question distribution. In *SIGIR 2010 workshop on crowdsourcing for search evaluation*, pages 21–26, 2010.

[17] David Oleson, Alexander Sorokin, Greg P Laughlin, Vaughn Hester, John Le, and Lukas Biewald. Programmatic gold: Targeted and scalable quality assurance in crowdsourcing. *Human computation*, 11(11), 2011.

[18] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms 2nd Edition*. The MIT Press, 2001.

[19] Frederick S. Hillier. *Linear and Nonlinear Programming*. Stanford University, 2008.

[20] David Streitfeld. The best book reviews money can buy. *New York Times*, 25, 2012.

[21] Chien-Ju Ho, Shahin Jabbari, and Jennifer Wortman Vaughan. Adaptive task assignment for crowdsourced classification. In *Proc. of ICML*, 2013.

[22] C.-J. Ho and J. W. Vaughan. Online task assignment in crowdsourcing markets. In *Proc. of AAAI*, 2012.

Appendix

Proof of Lemma 4.2

Proof In Case III,

Since $\frac{1}{\left(\alpha_{i,l}^{(k)}+\omega\right)\left(\alpha_{i,l-1}^{(k)}+\omega\right)} \geq \frac{(2r_2l\delta+r_1)(2r_2(l-1)\delta+r_1)}{\beta^2}$ and $\alpha_{i,l}^{(k)} - \alpha_{i,l-1}^{(k)} \leq \frac{\beta}{2r_2l\delta+r_1} - \frac{\beta}{2r_2(l-2)\delta+r_1} = \frac{-4\beta r_2\delta}{(2r_2l\delta+r_1)(2r_2(l-2)\delta+r_1)}$, we have

$$\begin{aligned}
&\sum_{l=1}^{k}\left(F_i^3(\mathbf{x}_i,y_{i,l}^t)-F_i^3(\mathbf{x}_i,y_{i,l-1}^t)\right) \qquad (46)\\
\leq\ & \sum_{l=2}^{k}\left(\frac{(2r_2(l-1)\delta+r_1)(2r_2l\delta+r_1)}{4r_2}-\frac{(r_1+2r_2\delta(l-1))^2}{4r_2}\right)\\
\times\ & \frac{-4\beta r_2\delta}{(2r_2l\delta+r_1)(2r_2(l-2)\delta+r_1)}\\
<\ & \sum_{l=2}^{k}\frac{-2\beta r_2\delta^2}{2r_2l\delta+r_1} < \sum_{l=2}^{k}\frac{-2\beta r_2\delta^2}{2r_2(k-1)\delta+r_1} = \frac{-2\beta r_2k\delta^2}{2r_2(k-1)\delta+r_1}
\end{aligned}$$

Accordingly,

$$\begin{aligned}
c_{i,k}^t &= F_i^3(\mathbf{x}_i,y_{i,l}^t)+\beta y_{i,k}^t = \sum_{l=1}^{k}\left(F_i^3(\mathbf{x}_i,y_{i,l}^t)-F_i^3(\mathbf{x}_i,y_{i,l-1}^t)\right)+\beta y_{i,k}^t\\
&< \frac{-2\beta r_2k\delta^2}{2r_2(k-1)\delta+r_1}+\beta e_k \qquad (47)
\end{aligned}$$

$$\begin{aligned}
&\sum_{l=1}^{k}\left(F_i^2(\mathbf{x}_i,y_{i,l}^t)-F_i^2(\mathbf{x}_i,y_{i,l-1}^t)\right)\\
\leq\ & \sum_{l=2}^{k}\left(\frac{(2r_2(l-1)\delta+r_1)(2r_2l\delta+r_1)}{4r_2}-\frac{(r_1+2r_2\delta(l-1))^2}{4r_2}\right)\\
\times\ & \frac{-4\beta r_2\delta}{(2r_2l\delta+r_1)(2r_2e_{i,l-2}+r_1)} \qquad (48)\\
<\ & \sum_{l=2}^{k}\frac{-2\beta r_2\delta^2}{2r_2l\delta+r_1} \qquad (49)\\
<\ & \sum_{l=2}^{k}\frac{-2\beta r_2\delta^2}{2r_2(k-1)\delta+r_1} \qquad (50)\\
=\ & \frac{-2\beta r_2k\delta^2}{2r_2(k-1)\delta+r_1} \qquad (51)
\end{aligned}$$

Accordingly,

$$\begin{aligned}
c_{i,k}^t &= F_i^2(\mathbf{x}_i,y_{i,l}^t)+\beta y_{i,k}^t\\
&= \sum_{l=1}^{k}\left(F_i^2(\mathbf{x}_i,y_{i,l}^t)-F_i^2(\mathbf{x}_i,y_{i,l-1}^t)\right)+\beta y_{i,k}^t\\
&< \frac{-2\beta r_2k\delta^2}{2r_2(k-1)\delta+r_1}+\beta e_k \qquad (52)
\end{aligned}$$

Proof of Lemma 4.3.

Proof For the sake of contradiction, assume that $c_{i,\min}^{t,(k)} < \beta(k-1)\delta$, then $F_i^3(\mathbf{x}_i,y_{i,k}^t) = c_{i,\min}^{t,(k)} - \beta y_{i,k}^t < \beta(k-1)\delta - \beta y_{i,k}^t \leq 0$. It implies that, using the optimal effort level $y_{i,k}^t$, worker i has utility lower than 0, which is even lower than the utility with 0 effort level. A contradiction.