A Budget-Limited Mechanism for Category-Aware Crowdsourcing Systems

Yuan Luo
Imperial College London, London
y.luo@imperial.ac.uk

Nicholas R. Jennings
Imperial College London, London
n.jennings@imperial.ac.uk

ABSTRACT

Crowdsourcing harnesses human effort to solve computer-hard problems. Such tasks often have different levels of difficulty and workers have varying levels of skill at completing them. With a limited budget, it is important to wisely allocate the spend among the tasks and workers such that the overall outcome is as good as possible. Most existing work addresses this budget allocation problem by assuming that workers have a single level of ability for all tasks. However, this neglects the fact that tasks can belong to a variety of diverse categories and workers may have varying abilities across them. To incorporating such category-awareness, we model the interaction between the crowdsource campaign initiator and the workers as a procurement auction and propose a computationally efficient mechanism, INCARE, to achieve high-quality outcomes given a limited budget. We prove that INCARE is budget feasible, incentive compatible and individually rational. Finally, our experiments on a standard real-world data set show that, compared to the state of the art, INCARE: (i) can improve the accuracy by up to 40%, given a limited budget; and (ii) is significantly more robust to inaccuracies in prior information about each task's difficulty.

KEYWORDS

crowdsourcing; budget limit; auction

ACM Reference Format:
Yuan Luo and Nicholas R. Jennings. 2020. A Budget-Limited Mechanism for Category-Aware Crowdsourcing Systems. In *Proc. of the 19th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2020), Auckland, New Zealand, May 9–13, 2020,* IFAAMAS, 9 pages.

1 INTRODUCTION

Crowdsourcing often involves a large number of workers with different backgrounds completing tasks such as item comparison, image labelling and entity resolution. In crowdsourcing platforms, such as Amazon Mechanical Turk and CrowdFlower, a crowdsource campaign initiator ("initiator" hereafter) can recruit hundreds of workers through the Internet to accomplish tasks and the workers are often rewarded for their efforts [10].

However, in many cases, the answers provided by workers are of low quality, as tasks are tedious and workers are non-experts [20]. To increase accuracy, most initiators collect multiple answers from different workers for each task. Although crowdsourcing is often viewed as a cheap source for accomplishing tasks, it is still important to use the initiator's limited budget wisely, especially when the budget is tight compared with the size of the task pool. Moreover, as workers have different levels of expertise and tasks may vary in difficulty, some workers will provide better answers than others. Hence, with a tight budget, the major challenge for an initiator is to design an effective allocation scheme. Specifically, the initiator needs to decide how many answers to collect for each task and how to assign the tasks to appropriate workers.

To address these problems, researchers have adopted several approaches (see Section 2 for details). However, a common drawback of these methods is that they often assume a single level of worker ability for all activities. This is usually incorrect [3]. For example, consider two workers (A, a car enthusiast, and B, an NBA fan) and tasks 1 (which asks workers to select a more fuel efficient car) and 2 (which asks workers to identify a more competitive basketball team). Intuitively, worker A is likely to do better than worker B when performing tasks belonging to category *auto*, while worker B should perform better in tasks related to *NBA*. Thus, an initiator should assign task 1 to worker A and task 2 to worker B. However, most existing approaches do not consider workers' varying abilities across different activities and so may assign task 1 to worker B and task 2 to worker A.

The issues of incorporating such *category-awareness* in the crowdsourcing process have recently been studied in [3, 22]. However, this work neglects the budget limitation problem. Hence, their algorithms cannot operate within a tight budget to collect as many high quality answers as possible for each task. Furthermore, they do not consider the problem of incentivization and merely price each task with an equal monetary reward. As workers have diverse expertise and different costs for performing a task, determining the correct reward for each worker is crucial: low rewards may not attract sufficient workers, causing insufficient answers for a task, while high rewards quickly deplete the budget without progressing much on the task at hand.

There are several works devoted to designing incentive mechanisms in crowdsourcing systems (see Section 2). However, these methods typically assume that workers report their costs truthfully and focus on pricing tasks given a budget. As a worker's cost is his private information, he may misreport it in order to obtain a higher reward. Moreover, this line of work does not consider the designed mechanism's robustness to the inevitable inaccuracies that are likely to exist in prior information about each task's difficulty level. Such robustness is central to the practical success of the mechanism. Specifically, unlike the workers' quality information where the initiator can use golden tasks or workers' historical performance, the prior information about each task's difficulty level cannot easily be estimated by the initiator. This is particularly true for new tasks where there are no historical results. Thus, the initiator will often have to run the mechanism based on inaccurate prior

Proc. of the 19th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2020), B. An, N. Yorke-Smith, A. El Fallah Seghrouchni, G. Sukthankar (eds.), May 9–13, 2020, Auckland, New Zealand.

information. Without robustness to this uncertainty, the mechanism will not progress much on the task at hand, even if it spends a significant amount of the budget [2].

Designing an incentive mechanism that explores workers' diverse abilities across different categories given limited budgets is non-trivial and has not been done before. It is complex because the rewards for workers are related to an initiator's budget allocation scheme and the workers' costs (that are private information). Moreover, the tasks' difficulty levels and workers' abilities across different tasks (both of which are unknown to the initiator) also affect the initiator's allocation scheme. Existing studies do not consider these relationships.

Against this background, this paper designs a truthful mechanism that satisfies a budget limit to incentivize workers to undertake tasks with binary answers[1] (*i.e.*, the answer is either 0 or 1). Our mechanism is called INCARE (INcentivize CAtegory-awaRE crowdsourcing system). In our model, each worker has various qualities in different categories and incurs different costs for accomplishing a task. To elicit workers' truthful costs and design appropriate rewards to incentivize workers to undertake different tasks, we use a procurement auction, which is commonly used in this field (*e.g.*, [1, 13, 20]), to model the interaction between an initiator and workers. In more detail, the initiator publishes a set of tasks, after which workers can submit bids for a subset of tasks according to their preferences. For performing a task, each worker has a private cost and claims it publicly in the bids. The initiator then determines the set of winning bids by which the tasks are allocated to the workers. The initiator pays each winning bid an amount of money based on all submitted bids in exchange for the answer.

This paper advances the state of the art in the following ways. First, INCARE is the first budget-limited mechanism for category-aware crowdsourcing applications. INCARE includes a budget allocation scheme that not only considers the difficulty levels of tasks, but also workers' diverse qualities in different categories. Second, INCARE's payment scheme is *Incentive Compatible* (*i.e.*, a worker will disclose his cost truthfully to maximize his utility which is the difference between the reward received from the initiator and the cost of accomplishing a task) and *Individually Rational* (*i.e.*, the utility of a worker is non-negative if he is selected to accomplish a task). Furthermore, INCARE guarantees budget feasibility, *i.e.*, the initiator's total payment to workers accomplishing tasks is guaranteed to be less than the budget. Finally, our empirical studies show that considering the workers' qualities in different categories and eliciting the costs of workers can significantly improve the accuracy of outcomes given a limited budget by up to 40%. Moreover, compared to the state of the art mechanisms, INCARE is significantly more robust to inaccuracies in the prior information about each task's difficulty level.

2 RELATED WORK

Research topics that consider quality in crowdsourcing tasks include how to aggregate crowd answers [23], how to select which workers should work on a task [12], and how to better design the crowdsourcing tasks to improve the accuracy of workers' answers [11]. By selecting informed workers, an initiator can require fewer answers for each task so that the total cost of the initiator is reduced. However, these methods do not consider differing worker abilities across different categories. Zheng *et al.* [22] do exploit workers' diverse qualities in different categories by designing an algorithm called DOCS. However, they do not consider budget constraints and different costs across diverse tasks. Thus, as shown by our numerical results in Section 5, DOCS will fail to attract high-quality workers and progresses less on the task at hand.

Motivated by the task cost concern, the budget allocation problem in crowdsourcing has attracted considerable attention [1, 14]. However, this work typically assumes that workers can always provide answers without appropriate incentives. The best example of this line of work is by Chen *et al.* [1] who propose an algorithm, called Opt-KG, to address the budget allocation problem with imperfect workers by assuming that the costs of workers are the same. In practice, however, different workers often incur different costs even when they accomplish the same tasks [20]. Without carefully designing rewards that cover workers' costs, this method may fail to incentivize workers to accomplish tasks, which would decrease the outcome quality of the overall task. Furthermore, they do not consider the impact of inaccuracies in prior information about each task's difficulty level. Without considering this commonly occurring situation [2], the performance of Opt-KG degrades significantly when the prior information is less accurate (see our numerical results in Section 5).

There are also several works devoted to incentive mechanism design in crowdsourcing [6, 17, 21]. However, these methods often assume that workers report their costs truthfully and focus on pricing tasks given a budget. As a worker's cost is his private information, he may misreport it in order to obtain a high reward. The state of the art in this area is by Zhang *et al.* [20] who do study the problem of eliciting truthful costs in the budget allocation problem. Their solution is called DI-Greedy-MUL. However, they ignore workers' quality differences in accomplishing tasks in different categories. Thus, they cannot help the initiator to obtain high quality outcomes with a limited budget (as we show in Section 5).

3 PROBLEM FORMULATION

We begin with a formal description of the interaction between the initiator and the workers. Then we will set up notations to capture the behaviors of the workers, the objective of the initiator and the mechanism design problem. Table 1 lists the key notation.

3.1 System Model

We consider a setting where an initiator collects answers for a set $\mathcal{N} = \{1, 2, \ldots, N\}$ of N tasks from a set $\mathcal{K} = \{1, 2, \ldots, K\}$ of K workers under a limited budget B[2]. For ease of presentation, each task is binary with only two choices. To incentivize workers to perform tasks assigned by the initiator, we model the interaction between them as a procurement auction [13], where the initiator publicizes a task set $\mathcal{N}$ and each worker $k \in \mathcal{K}$ replies with a set $\Phi_k = \{\phi_k^1, \phi_k^2, \ldots, \phi_k^{m_k}\}$ of m_k bids. Each submitted bid is a task-price pair $\phi_k^m = (n_k^m, b_k^m)$, where $n_k^m \in \mathcal{N}$ denotes the task selected

[1]We focus on binary answers for analytical convenience but the revision required to the general case are straightforward.

[2]For convenience, we refer to the initiator as "she" and workers as "he".

Table 1: Key Notation

$\mathcal{N} = \{1, \ldots, n, \ldots, N\}$	index set of tasks
$\mathcal{K} = \{1, \ldots, k, \ldots, K\}$	index set of workers
$\mathcal{L} = \{1, \ldots, l, \ldots, L\}$	index set of task categories
$\boldsymbol{r}_n = [r_{n,1}, \ldots, r_{n,l}, \ldots, r_{n,L}]$ $\boldsymbol{r}_n = \{r_{n,l}\}_{l \in \mathcal{L}}$	the distribution that task $n \in \mathcal{N}$ is related to each category $l \in \mathcal{L}$
θ_n	difficulty level of the n-th task
$\delta_{k,l}$	quality of worker k for the l-th category
$y_{n,k}$	worker k's answer to task n
$\Phi_k = \{\phi_k^1, \phi_k^2, \ldots, \phi_k^{m_k}\}$	a set of worker k's submitted bids with total number of m_k bids
$\phi_k^m(n_k^m, b_k^m)$	worker k's m-th bid that denotes the task he can perform and the charge
$\omega^t = \{n^t, k^t\}$	selected task-worker pair in round t
$p(\phi_k^m)$	worker k's payment based on his bid ϕ_k^m
$\hat{R}(\phi, t)$	marginal contributions of bid ϕ in winning bid selection round t
$\hat{U}(\Phi^{\mathrm{W}})$	aggregated marginal contributions determined by the winning bid sequence Φ^{W}

by worker k in his m-th bid and b_k^m is the price worker k claims for accomplishing task n_k^m.

The initiator sequentially determines the winning bid set $\Phi^{\mathrm{W}} \subseteq \Phi = \bigcup_{k \in \mathcal{K}} \Phi_k$ from all submitted bids until the budget or the available bids are exhausted[3]. In each winning bid selection round t, the initiator selects a bid ϕ_k^m into the current winning bid set based on the collected answers up to round $t-1$. This selected bid reveals a task-worker pair, *i.e.*, $\omega^t(\phi_k^m) = \{n^t, k^t\}$, that indicates the task that needs to be finished and the worker who should perform it. To simplify notation, we use ω^t instead of $\omega^t(\phi_k^m)$ to denote the selected task-worker pair based on the selected winning bid in round t. Once the initiator receives the answer from the selected worker, *i.e.*, y_{n^t,k^t}, she updates the information about the tasks and workers based on a Bayesian method (details in Section 4.1) before selecting the new bid in the next winning bid selection round[4].

After the winning bid set Φ^{W} is determined, the initiator pays each winning bid $\phi_k^m \in \Phi^{\mathrm{W}}$ an amount of money $p(\phi_k^m)$.

Worker's Cost and Utility: If worker k's bid ϕ_k^m is selected by the initiator, the cost incurred by worker k when performing task n_k^m is $c(n_k^m)$, which is the private information of worker k. Note that the charge b_k^m claimed by worker k in his submitted bid ϕ_k^m could be different from worker k's actual cost $c(n_k^m)$. Each worker $k \in \mathcal{K}$ strategically determines b_k^m to maximize his payoff which is defined as the difference between the total payment he receives from the initiator and his total cost. Specifically, for worker $k \in \mathcal{K}$, his payoff is

$$U^{\mathrm{UE}}(\Phi_k, \Phi_{-k}) = \sum_{\phi_k^m \in \Phi^{\mathrm{W}}} \left[p(\phi_k^m) - c(n_k^m)\right], \tag{1}$$

where $\Phi_{-k} = \bigcup_{j \in \mathcal{K}, j \neq k} \Phi_j$ is the set of all workers' bids except worker k. Note that a worker's payoff is calculated based on this worker's bids that are selected into the winning bid set Φ^W which is a function of all workers' submitted bids, i.e., Φ_k and Φ_{-k}. As U^{UE} is a cumulative function based on Φ^{W} and whether a worker's bid is selected depends not only on the bid this worker submitted but also on other workers' submitted bids, we write U^{UE} as a function of Φ_k and Φ_{-k}.

Task Difficulty Level: Let Z_n be the true answer for task $n \in \mathcal{N}$ with value either 0 or 1. We model Z_n as a random variable and use $\theta_n = Pr(Z_n = 1)$ to denote the difficulty level of task n[5]. As is common in the literature (*e.g.*, [1, 20]), we assume that θ_n is consistent with the true label in the sense that $Z_n = 1$ (or $Z_n = 0$) if and only if $\theta_n \geq 0.5$ (or $\theta_n < 0.5$). In such cases, the initiator can infer the true answer for each task n by estimating the value of θ_n.

Category Vector: We assume there is a set $\mathcal{D} = \{d_1, d_2, \ldots, d_L\}$ of L categories. An example is $\mathcal{D} = \{food, NBA, auto, country\}$. The initiator can obtain this category set by existing knowledge based on question answering systems such as categories in Yahoo Answers, domains in Freebased, and main topics in Wikipedia. The reason for using general topics is that they can interpret a task in a fine-grained manner [22].

Task Category Vector: Let $\boldsymbol{r}_n = [r_{n,1}, r_{n,2}, \ldots, r_{n,L}]$ represent the distribution that task $n \in \mathcal{N}$ is related to each category in $\mathcal{D}$, where $r_{n,l} \in [0,1], l \in \mathcal{L} = \{1, 2, \ldots, L\}$ and $\sum_{l=1}^{L} r_{n,l} = 1$. A higher value of $r_{n,l}$ represents the fact that task n is more related to category l. Let o_n denote task n's true category, and we have $Pr(o_n = l) = r_{n,l}$. For example, considering a task n: "which country produces more cars, China or Japan?" that is related to categories *auto* and *country* in $\mathcal{D}$. This is probably more relevant to *auto*, so a reasonable category vector of task n is $\boldsymbol{r}_n = [0, 0, 0.8, 0.2]$. We assume the initiator knows this category vector for all tasks, as computation of a task's category vector is orthogonal to our mechanism design problem and an initiator can calculate all tasks' category vectors based on the methods proposed in [3, 22] before determining the budget allocation scheme. Let $\eta_{n,l} = Pr(Z_n = 1 | o_n = l)$ represent the value of task difficulty for the l-th category. Based on the definition of $\boldsymbol{r}_n$, we have $\theta_n = \sum_{l=1}^{L} r_{n,l} \cdot \eta_{n,l}$.

Worker Quality Vector: Let $\boldsymbol{\delta}_k = [\delta_{k,1}, \delta_{k,2}, \ldots, \delta_{k,L}]$ model the quality vector of worker $k \in \mathcal{K}$, where $\delta_{k,l} \in [0,1], l \in \{1, 2, \ldots, L\}$ indicates the expertise (accuracy) of worker k in answering tasks in category d_l. A higher value $\delta_{k,l}$ means that worker k has more expertise in category d_l. For example, if worker k is a geographer and an NBA fan, while pays no attention to food and auto, then a suitable quality vector is $\boldsymbol{\delta}_k = [0.2, 0.8, 0.3, 0.9]$ for performing tasks belonging to the set $\mathcal{D} = \{food, NBA, auto, country\}$.

[3]The reason we select winning bids in a sequential fashion is that the initiator does not know the tasks' difficulty or workers' qualities at the beginning of the budget allocation process. Thus, she needs to update this information based on the stream of collected answers in a sequential fashion so that she can adjust the task-worker assignment promptly given a budget limit.

[4]The benefits of using Bayesian modeling are discussed in [1].

[5]We interpret θ_n as the relative frequency that 1 appears when the number of perfectly reliable workers approaches infinity. Hence, when θ_n is close to 0.5, it implies that task n is difficult as the number of perfectly reliable workers whose answers are 1 is the same as that of perfectly reliable workers whose answers are 0 even though the initiator asks a large number of perfectly reliable workers. When θ_n is close to 1 or 0, task n is relatively more easy as the initiator can obtain the answer by asking sufficient numbers of perfectly reliable workers.

The initiator does not perfectly know the workers' qualities at the beginning of the budget allocation process and can learn workers' quality vectors together with the tasks' difficulty level based on workers' answers (details in Section 4.1).

Worker Answer: Let $y_{n,k} \in \{0,1\}$ denote an answer from worker $k \in \mathcal{K}$ to task n. The initiator considers $y_{n,k}$ as a random variable $Y_{n,k}$ before she asks worker k.

We use $v_{k,l}^{t-1}$ to store the aggregated weight of category $l \in \mathcal{L}$ for those tasks answered by worker $k \in \mathcal{K}$ before the winning bid selection round t, *i.e.*, $v_{k,l}^{t-1} = \sum_{j \in \mathcal{N}_k^{t-1}} r_{j,l}$, where $\mathcal{N}_k^{t-1}$ records the tasks that are answered by worker k up to round $t-1$. Let $\mathcal{S}^t$ be a state space in round t that consists of all possible posterior task difficulty level parameters and worker quality parameters, *i.e.*, $\mathcal{S}^t = \{\{\eta_{j,l}^t\}_{j \in \mathcal{N}, l \in \mathcal{L}}, \{\theta_j^t\}_{j \in \mathcal{N}}, \{\delta_{i,l}^t\}_{i \in \mathcal{K}, l \in \mathcal{L}}, \{v_{i,l}^t\}_{i \in \mathcal{K}, l \in \mathcal{L}}\}$.

If the initiator chooses the winning bid ϕ_k^m in the current round t, we can have a task-worker pair $\omega^t = \{n^t, k^t\}$ and calculate the posterior probability $Pr(y_{n^t,k^t} | \mathcal{S}^t, n^t, k^t)$ that we are in the next state $\mathcal{S}^{t+1}$, as follows:

$$
\begin{aligned}
&Pr(y_{n^t,k^t} = 1 | \mathcal{S}^t, n^t, k^t) \\
&= \sum_{l=1}^{L} r_{n,l} \cdot \left[\delta_{k,l}^t \cdot \eta_{n,l}^t + (1-\delta_{k,l}^t) \cdot (1-\eta_{n,l}^t)\right]
\end{aligned} \tag{2}
$$

$$
\begin{aligned}
&Pr(y_{n^t,k^t} = 0 | \mathcal{S}^t, n^t, k^t) \\
&= \sum_{l=1}^{L} r_{n,l} \cdot \left[\delta_{k,l}^t \cdot (1-\eta_{n,l}^t) + (1-\delta_{k,l}^t) \cdot \eta_{n,l}^t\right]
\end{aligned} \tag{3}
$$

The Initiator's Utility: When the budget or the available bids are exhausted, the initiator needs to infer the true answers $\{Z_n\}_{n \in \mathcal{N}}$ for all tasks based on all collected answers from the workers determined by the winning bid Φ^{W}. According to the definition of task difficulty level θ_n in Section 3.1, this is equivalent to inferring the true value of $\theta_n, \forall n \in \mathcal{N}$.

Intuitively, if the value of θ_n is real 0.5, the correct answer of task n is difficult for the initiator to infer as both answers (0 or 1) are likely to be given by the same number of workers. If the value of θ_n is close to 1 or 0, the initiator has more confidence to derive the correct answer. By capturing this idea, we apply entropy, which is common in the literature (*e.g.*, [16, 21, 22]), to define the ambiguity of θ_n, *i.e.*,

$$\mathcal{H}(\theta_n) = -\theta_n \cdot \ln \theta_n - (1-\theta_n) \cdot \ln(1-\theta_n). \tag{4}$$

According to the definition of entropy, the higher the value of $\mathcal{H}(\theta_n)$, the more ambiguous θ_n is, thus the less accurate the answer inferred by the initiator. Hence, the initiator's objective is spending her budget to collect answers that reduce the ambiguity of every $\theta_n, \forall n \in \mathcal{N}$.

Let θ_n^0 denote the prior information of task n's difficulty level. Let $\theta_n | \mathcal{F}^{\mathrm{W}}$ denote the posterior information, where $\mathcal{F}^{\mathrm{W}}$ is the σ-algebra generated by the winning bid selection path that denotes the collected information during the whole winning bid selection process. The initiator's utility is defined as how much ambiguity can be reduced for every task $n \in \mathcal{N}$,

$$U^{\mathrm{IN}} = \mathrm{E}\left[\sum_{n=1}^{\mathcal{N}} \left[\mathcal{H}(\theta_n^0) - \mathcal{H}(\theta_n | \mathcal{F}^{\mathrm{W}})\right]\right], \tag{5}$$

where the expectation is taken over the winning bid selection paths $\mathcal{F}^{\mathrm{W}}$.

3.2 Problem Formulation

Before explaining the problem, we begin by defining three standard game-theoretic terms that are required for subsequent discussions.

Definition 3.1 (Incentive Compatibility). Let $\hat{\phi}_k^m = (n_k^m, c(n_k^m))$ be worker k's truthful bid that reveals his true cost and $\phi_k^m = (n_k^m, b_k^m)$ be the untruthful bid, where $b_k^m \neq c(n_k^m)$. A mechanism satisfies the incentive compatibility (IC) condition if the worker's payoff when performing task n_k^m satisfies [5]

$$p(\hat{\phi}_k^m) - c(n_k^m) \geq p(\phi_k^m) - c(n_k^m). \tag{6}$$

IC means that for any worker $k \in \mathcal{K}$, reporting his true cost will maximize his payoff.

Definition 3.2 (Individual Rationality). A mechanism satisfies individual rationality (IR) if the payoff of every worker $k \in \mathcal{K}$ achieved by his winning bid is non-negative, given that he truthfully reports his cost [5]

$$p(\hat{\phi}_k^m) - c(n_k^m) \geq 0, \tag{7}$$

where $\hat{\phi}_k^m = (n_k^m, c(n_k^m))$.

Assuming (without loss of generality) that the payoff of a worker not performing any task equals zero, IR means that a worker will perform tasks only if his payoff is at least as much as that of a non-participating worker.

Definition 3.3 (Budget Feasibility). A mechanism satisfies budget feasibility, if the initiator's total payment to all winning bids is less than or equal to her total budget B [5], *i.e.*, $\sum_{\phi \in \Phi^{\mathrm{W}}} p(\phi) \leq B$.

Based on the revelation principle [9], the problem of finding a mechanism that maximizes the initiator's utility can be restricted to the set of mechanisms where workers are willing to reveal their private information to the initiator. Moreover, the initiator cannot force workers to accept the task. Given this, we use a procurement auction to model the interaction between the initiator and workers and focus on designing a mechanism that determines a winning bid sequence Φ^{W} given a strict budget constraint, while ensuring truthful reports from workers and their participation. Mathematically, the problem we want to solve is:

$$
\begin{aligned}
&\max \mathrm{E}\left[\sum_{n=1}^{\mathcal{N}} \left[\mathcal{H}(\theta_n^0) - \mathcal{H}(\theta_n | \mathcal{F}^{\mathrm{W}})\right]\right] \\
&\text{subject to } \sum_{\phi \in \Phi^{\mathrm{W}}} p(\phi) \leq B \text{ and} \\
&\qquad \text{IC and IR in equations (6) and (7),}
\end{aligned} \tag{8}
$$

where the expectation is taken over the sample paths $\mathcal{F}^{\mathrm{W}}$.

4 THE INCARE MECHANISM

We present our mechanism to solve the problem defined in (8). We first show how an initiator updates the information about the tasks and workers given a worker's answer. Based on the information update method, we then present INCARE including its winning bid selection scheme and its payment scheme that incentivizes workers to perform suitable tasks given a limited budget. Finally, we formally prove the mechanism's properties.

4.1 Information Update Process

Once the initiator chooses the winning bid ϕ_k^m that determines the task-worker pair ω^t in the current winning bid selection round t, the elements in the state space $\mathcal{S}^t$ will be updated based on the following equations [22], given the answer of worker k is y_{n^t,k^t}.

For task $j \in \mathcal{N}$ and $j \neq n^t$, its difficulty level in $\mathcal{S}^t$ is

$$\eta_{j,l}^t = \eta_{j,l}^{t-1},\ \theta_j^t = \theta_j^{t-1}, \tag{9}$$

which is independent of worker k^t's answer.

For task n^t, its difficulty level in $\mathcal{S}^t$ is

$$\eta_{n,l}^t | y_{n^t,k^t} = \begin{cases} \frac{\eta_{n,l}^{t-1}\cdot\delta_{k,l}^{t-1}}{\eta_{n,l}^{t-1}\cdot\delta_{k,l}^{t-1}+(1-\eta_{n,l}^{t-1})\cdot(1-\delta_{k,l}^{t-1})} & \text{if } y_{n^t,k^t}=1 \\ \frac{\eta_{n,l}^{t-1}\cdot(1-\delta_{k,l}^{t-1})}{\eta_{n,l}^{t-1}\cdot(1-\delta_{k,l}^{t-1})+(1-\eta_{n,l}^{t-1})\cdot\delta_{k,l}^{t-1}} & \text{if } y_{n^t,k^t}=0 \end{cases}$$

$$\theta_n^t | y_{n^t,k^t} = \sum_{l=1}^{L} r_{n,l}\cdot(\eta_{n,l}^t | y_{n^t,k^t}). \tag{10}$$

The basic idea of (10) is to update the information of a task's difficulty for each category based on the selected worker's provided answer and this worker's quality in the same category. High quality means that this worker's answer is more reliable in this category, and the accumulated value added to this task's difficulty level is large.

Based on the result obtained in (10), we update the selected worker's quality in the following equation. In more detail, for worker k^t, his quality in $\mathcal{S}^t$ is

$$\delta_{k,l}^t | y_{n^t,k^t} = \begin{cases} \frac{\delta_{k,l}^{t-1}\cdot v_{k,l}^{t-1}+\theta_n^t\cdot r_{n,l}}{v_{k,l}^{t-1}+r_{n,l}} & \text{if } y_{n^t,k^t}=1 \\ \frac{\delta_{k,l}^{t-1}\cdot v_{k,l}^{t-1}+(1-\theta_n^t)\cdot r_{n,l}}{v_{k,l}^{t-1}+r_{n,l}} & \text{if } y_{n^t,k^t}=0 \end{cases}$$

$$v_{k,l}^t = v_{k,l}^{t-1} + r_{n,l},\ \forall l \in \mathcal{L}. \tag{11}$$

From (11), we can see that if the task's updated result is consistent with the worker's provided answer, the worker's quality value increases.

Similar to (11), we replace the old information about this task's answer with the latest information in (12) and update the quality information of workers who have previously performed this task. Specifically, let $\mathcal{K}^{n^t}$ store the workers that perform task n^t before round t. For worker $i \in \mathcal{K}^{n^t}$ his quality in $\mathcal{S}^t$ is

$$\delta_{i,l}^t | y_{n^t,k^t} = \begin{cases} \frac{\delta_{i,l}^{t-1}\cdot v_{i,l}^{t-1}-\theta_n^{t-1}\cdot r_{n,l}+\theta_n^t\cdot r_{n,l}}{v_{i,l}^{t-1}} & \text{if } y_{n^t,i}=1 \\ \frac{\delta_{i,l}^{t-1}\cdot v_{i,l}^{t-1}-(1-\theta_n^{t-1})\cdot r_{n,l}+(1-\theta_n^t)\cdot r_{n,l}}{v_{i,l}^{t-1}} & \text{if } y_{n^t,i}=0 \end{cases}$$

$$v_{i,l}^t = v_{i,l}^{t-1}. \tag{12}$$

For other workers $i \in \mathcal{K} \setminus \mathcal{K}^{n^t}, i \neq k^t$ who have not answered task n^t before, their quality in $\mathcal{S}^t$ is

$$\delta_{i,l}^t = \delta_{i,l}^{t-1},\ v_{i,l}^t = v_{i,l}^{t-1},\ \forall l \in \mathcal{L}. \tag{13}$$

4.2 Winning Bid Selection and Payments

Based on the information update method presented in Section 4.1, the initiator can update information about the workers' quality vectors together with the tasks' difficulty levels based on workers' answers in each winning bid selection round. Then we can turn to address the problem defined in (8), *i.e.*, how to select the task-worker pair in each winning bid selection round based on the initiator's learnt information and a limited budget, while satisfying the IC and IR conditions defined in equations (6) and (7).

Note that the problem defined in (8) is NP-hard [20]. Therefore, we adopt a randomized knowledge gradient (KG) scheme[6] [4], which is essentially a single-step look-ahead approach, to solve the problem. In more detail, randomized KG greedily selects the next winning bid that generates the largest expected reward:

$$\phi_k^m = (n_k^m, b_k^m) = \arg\max_{\phi\in\Phi}\left(\hat{R}(\phi) = \mathcal{H}(\theta_n^{t-1}) - \hat{\mathcal{H}}(\theta_n^t)\right). \tag{14}$$

When there is a tie in (14), we randomly break it instead of selecting the one with the smallest index during the winning bid selection. We calculate the value of $\hat{\mathcal{H}}(\theta_n^t)$ as follows:

$$\begin{aligned}\hat{\mathcal{H}}(\theta_n^t) =& Pr(y_{n^t,k^t}=1|n^t,k^t)\cdot\mathcal{H}(\theta_n^t|y_{n^t,k^t}=1) \\ &+ Pr(y_{n^t,k^t}=0|n^t,k^t)\cdot\mathcal{H}(\theta_n^t|y_{n^t,k^t}=0),\end{aligned} \tag{15}$$

where the value of $\theta_n^t|y_{n^t,k^t}$ is calculated by (10). Note that $\hat{R}(\phi)$ calculates bid ϕ's marginal contribution between two successive rounds.

By applying randomized KG, the objective function of (8) becomes equivalent to designing a mechanism that determines a winning bid sequence Φ^{W*} that maximizes the aggregated marginal contributions, *i.e.*,

$$\begin{aligned}&\max \hat{U}(\Phi^W) = \sum_{\phi\in\Phi^W}\hat{R}(\phi),\\ &\text{subject to} \sum_{\phi\in\Phi^W} p(\phi) \le B \text{ and}\\ &\text{IC and IR in equations (6) and (7).}\end{aligned} \tag{16}$$

As the randomized KG scheme involves a single look-ahead step, we do not need to take the expectation over the whole sample paths as in (8). To solve the above problem, we first need to check whether the function $\hat{U}(\Phi^W)$ falls in the family of monotone submodular functions. If so, then we can revise the greedy allocation scheme in [15] to tackle the problem.

Definition 4.1 (Monotone Submodular Function). Let $\mathcal{A}$ be a finite set. For any $X \subseteq Y \subseteq \mathcal{A}$ and $x \in \mathcal{A}$ and $x \notin Y$, a function

[6]There are other possible approximate schemes such as the randomized scheme [8], the Gittins index scheme [19] and the new labeling uncertainty scheme [7]. However, none of them has better performance or lower computation complexity than KG [1].

Algorithm 1 Winning Bid Selection Scheme

The initiator sets the prior value for tasks' difficulties $\{\theta_n^0\}_{n\in\mathcal{N}}$, $\{\eta_{n,l}^0\}_{n\in\mathcal{N},l\in\mathcal{L}}$, workers' qualities $\{\delta_{k,l}^0\}_{k\in\mathcal{K},l\in\mathcal{L}}$, the budget B and the task category vectors $\{\boldsymbol{r}_n\}_{n\in\mathcal{N}}$;
Initialize: $t \leftarrow 0$, $\Phi^{\mathbf{W}*} \leftarrow \emptyset$, $\Omega \leftarrow \emptyset$
while $\Phi \neq \emptyset$ **do**
  Let $\phi^* = \{n_{k^*}^*, b_{k^*}^*\} \leftarrow \arg\max_{\phi_k^m\in\Phi} \frac{\hat{R}(\phi_k^m)}{b_k^m}$
  if $b_{k^*}^* \leq \frac{B}{\alpha}\frac{\hat{R}(\phi^*)}{\hat{U}(\Phi^{\mathbf{W}}\cup\{\phi^*\})}$ **then**
    $\Omega \leftarrow \Omega \cup \{n^*, k^*\}$, $\Phi^{\mathbf{W}*} \leftarrow \Phi^{\mathbf{W}*} \cup \{\phi^*\}$;
    Acquire the label $y_{n^*,k^*} \in \{0,1\}$;
    Update $\eta_{n^*,l}^t$, $\forall l \in \mathcal{L}$ **and** $\theta_{n^*}^t$ **based on** (10);
    Update $\delta_{k^*,l}^t$, $\forall l \in \mathcal{L}$ **and** $v_{k^*,l}^t$ **based on** (11);
    for worker $i \in \mathcal{K}^{n^*}$, $i \neq k^*$ **who has answered the task** n^* **before do**
      Update $\delta_{i,l}^t$, $\forall l \in \mathcal{L}$ **and** $v_{i,l}^t$ **based on** (12);
    end for
    $\eta_{j,l}^{t+1} \leftarrow \eta_{j,l}^t$, $\theta_j^{t+1} \leftarrow \theta_j^t$, $\forall j \in \mathcal{N}, j \neq n^*$;
    $\delta_{i,l}^{t+1} \leftarrow \delta_{i,l}^t$, $v_{i,l}^{t+1} \leftarrow v_{i,l}^t$, $\forall i \in \mathcal{K} \setminus \mathcal{K}^{n^*}$, $i \neq k^*$, $\forall l \in \mathcal{L}$,;
    $t \leftarrow t+1$;
  end if
  $\Phi \leftarrow \Phi \setminus \phi^*$;
end while

$f : 2^{\mathcal{A}} \to \mathbb{R}$ is monotonic if and only if $f(X) \leq f(Y)$, and the function f is submodular if and only if

$$f(X \cup \{x\}) - f(X) \geq f(Y \cup \{x\}) - f(Y). \tag{17}$$

Proposition 4.2. *The aggregated marginal contributions of the winning bid set $\hat{U}(\Phi^W)$ is monotone submodular.*

Due to the space limit, we only sketch the proofs of all propositions and theorems.

Proof Sketch. As a sum of monotonic submodular functions is still submodular, we only need to prove that function $\hat{R}(\phi)$ is monotonically non-increasing with the number of collected answers for every $\theta_n, n \in \mathcal{N}$. Let Y and X represent the collected answers from workers for task n after t rounds and $j \leq t$ rounds of selection, respectively. Let $x \notin Y$ be the collected answer from a worker whose answer is not included in set Y in round $t+1$. Then we need to prove

$$\begin{aligned} &-\mathcal{H}(\theta_n|X\cup\{x\}) + \mathcal{H}(\theta_n|X) \\ &\quad \geq -\mathcal{H}(\theta_n|Y\cup\{x\}) + \mathcal{H}(\theta_n|Y). \end{aligned} \tag{18}$$

which is equivalent to proving

$$-\sum_{\theta_n, Y, x} Pr(\theta_n, Y, x) \log \frac{Pr(\theta_n|X)Pr(\theta_n|Y,x)}{Pr(\theta_n|X,x)Pr(\theta_n|Y)} \geq 0. \tag{19}$$

Considering workers accomplish task n independently and the value of Kullback-Leibler distance is always positive, we can prove the result. □

With Proposition 4.2 in place, we can describe our mechanism. INCARE first obtains the bids Φ from all workers, then runs the winning bid selection scheme to select the winning bid and makes observations. Finally, INCARE computes the payment for each winning bid.

4.2.1 Winning Bid Selection Scheme. Algorithm 1 presents the winning bid selection scheme. The main idea of this scheme is to greedily select the winning bid ϕ_k^m in round t that maximizes the marginal value $\hat{R}(\phi_k^m), \forall \phi_k^m \in \Phi$ per unit cost. Similar to the scheme proposed by [13], we use a proportional share rule on a reduced budget B/α, $\alpha \geq 1$ to ensure budget feasibility[7]. Intuitively, a larger α better guarantees the budget constraint, while a smaller α better utilizes the budget. The choice of α that achieves the budget feasibility will be shown in the proof of Theorem 4.5.

4.2.2 Payment Scheme. The payment scheme is based on the characterization of threshold payments that is widely used in existing work (*e.g.*, [13, 18]) to incentivize truthful reports from workers. Specifically, given the winning set $\Phi^{W*} = \{\phi^1, \ldots, \phi^i, \ldots, \phi^{|\Phi^{W*}|}\}$ that reflects the winning bid selection order, we analyze the threshold payment for the i-th winning bid $p(\phi^i)$ which is a maximum of all the possible prices that the worker with this winning bid ϕ^i can declare and still get selected.

Consider running the winning bid selection scheme on an alternate bid set $\hat{\Phi} = \Phi \setminus \{\phi^i\}$ and getting the new winning set $\hat{\Phi}^{W*} = \{\hat{\phi}^1, \hat{\phi}^2, \ldots, \hat{\phi}^j, \ldots, \hat{\phi}^{|\hat{\Phi}^{W*}|}\}$. If the worker with bid $\{\phi^i\}$ wants to be selected in $\hat{\Phi}^{W*}$ in round j, the maximum price that the agent with bid $\{\phi^i\}$ can claim in his bid is $b_{i(j)} = \min\{\beta_{i(j)}, \eta_{i(j)}\}$, where $\beta_{i(j)} = \frac{b(\hat{\phi}^j)\cdot\hat{R}_{i(j)}}{\hat{R}(\hat{\phi}^j)}$, $\hat{R}_{i(j)}$ is the marginal contribution of worker i in round j, and $\eta_{i(j)} = \frac{B}{\alpha}\cdot\frac{\hat{R}_{i(j)}}{\hat{U}\left(\{\hat{\phi}^t\}_{t=1}^{j-1}\cup\{\phi^i\}\right)}$. Based on the intuition of threshold payments [15], we can see that the payment to the worker with winning bid $\{\phi^i\}$ in the winning set Φ^{W*} should be

$$p(\phi^i) = \max_{j\in\hat{\Phi}^{W*}} b_{i(j)}, \tag{20}$$

where the maximum value is taken over the possible $|\hat{\Phi}^{W*}|$ rounds in $\hat{\Phi}^{W*}$.

4.3 Analysis of INCARE

In this section, we will analyze the mechanism.

Theorem 4.3. *INCARE satisfies the IC condition.*

Proof Sketch. We first prove that the winning bid selection scheme has the monotonicity property, *i.e.*, if a bid $\phi_k^m = (n_k^m, b_k^m)$ is selected, then a new bid $\tilde{\phi}_k^m = (n_k^m, \tilde{b}_k^m)$ with $\tilde{b}_k^m \leq b_k^m$ is also selected. Then we show that the payment has a threshold nature. According to the results of [9], we can get the result. □

Theorem 4.4. *INCARE satisfies the IR condition.*

Proof Sketch. Recall the payment to a winning bid $\phi^i \in \Phi^W$ is calculated based on (20). We first show that $b_{i(j)} \geq b_i$ for $j \in \hat{\Phi}^{W*}$. Then, based on Theorem 4.3, each worker should reveal his true cost in the submitted bid. Thus, we have $p(\phi^i) \geq b_i = c(n_k^m)$, from which we can get the result. □

[7] As shown by [13], standard incentive compatible mechanisms such as the VCG and its variants are not applicable in a budget-limited setting.

Theorem 4.5. *INCARE is budget feasible if the budget fraction ratio $\alpha = 2$. Moreover, an initiator can compute a specific tighter bound on $\alpha \leq 2$ to better utilize the budget.*

Proof Sketch. We first show that the maximum price that a worker with a winning bid ϕ can claim is upper bounded by $2B\frac{\hat{R}(\phi)}{\hat{U}(\Phi^{\mathrm{w}})}$ irrespective of the winning bid selection round. Then we prove that the summation over these upper bounds is equal to B when $\alpha = 2$. □

Theorem 4.6. *INCARE is computationally efficient, having complexity $O(|\Phi|^2 \log|\Phi| + |\Phi^{W}| \cdot |\Phi|^2 \log|\Phi| + |\hat{\Phi}^{W*}|)$.*

Proof Sketch. As shown in Algorithm 1, the computation complexity for winning bid selection is $O(|\Phi|^2 \log|\Phi|)$. INCARE needs to run $|\Phi^{W}|$ times for every winning bid $\phi^i \in \Phi^{W}$ on a set $\Phi \setminus \{\phi^i\}$. Then INCARE chooses the maximum value from a set $\hat{\Phi}^{W*}$. Thus, the computation complexity of payment calculation is $O(|\Phi^{W}| \cdot |\Phi|^2 \log|\Phi| + |\hat{\Phi}^{W*}|)$. □

5 EMPIRICAL EVALUATION

We conduct an empirical study to evaluate the performance of INCARE. This complements the theoretical analysis by showing how eliciting workers' true costs and learning their diverse abilities across different categories can achieve a high quality outcome given a limited budget. We compare INCARE to other state of the art mechanisms that do not include these two actions.

(1) DOCS: As per Section 2, DOCS sequentially selects a task-worker pair by exploiting workers' diverse qualities in different categories. However, it does not consider the workers' varying costs across diverse tasks, nor the expense of eliciting the true costs from workers. Based on the IR condition (Definition 3.2), a worker will perform tasks only if his payoff is non-negative, *i.e.*, the reward he received should at least cover his cost. As DOCS does not know workers' costs, we assume the initiator pays each selected worker the average value of all workers' costs. We run DOCS until the budget is depleted or the available workers are exhausted.
(2) Opt-KG: As per Section 2, Opt-KG tackles the budget allocation problem without knowing the tasks' difficulties and workers' qualities. Opt-KG treats a worker's quality as a single level and does not elicit workers' true costs. Thus, similar to DOCS, Opt-KG assumes the initiator pays each selected worker the average cost of all workers and stops the budget allocation process when the budget is depleted or the available workers are exhausted.
(3) DI-Greedy-MUL: As per Section 2, DI-Greedy-MUL not only addresses the budget allocation problem but also elicits the true costs of workers. Similar to Opt-KG, it treats a worker's quality as a single level.

We compare the different schemes on a standard real-world data set for comparing two items based on specified comparison criteria [3]. We choose this item comparison data set because it contains category information for each comparison task. There are 3665 instances with four categories: *food*, *NBA*, *auto*, and *country*, where each instance is an item comparison pair. Each comparison task asks workers to compare two items and give a binary choice.

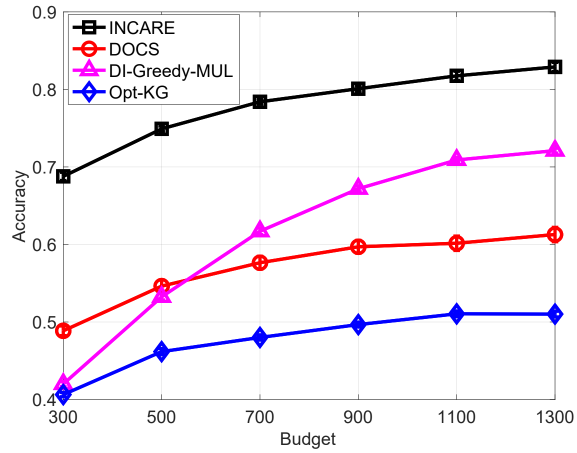

Figure 1: The accuracy vs. different budget.

For example, a task belonging to category *food* asks workers to compare two food items, *e.g.*, Chocolate and Honey, and say which has more calories. There are 360 comparison tasks where each category has 90 tasks and 126 different workers. We set a worker k's cost for performing a task based on his quality in answering previous tasks in the same category (the results for other costs functions are broadly similar). For example, if a quality vector for worker k for performing tasks belonging to the category set $\mathcal{D} = \{food, NBA, auto, country\}$ is $\boldsymbol{\delta}_k = [0.6, 0.9, 0.5, 0.2]$, his costs for performing tasks belong to these four categories are $\boldsymbol{c} = 0.01 + \boldsymbol{\delta}_k = [0.61, 0.91, 0.5, 0.2]$. Here, 0.01 makes sure that even if a worker's quality is 0 for performing tasks in a given category, performing a task still incurs some cost to this worker as he spends time and energy to finish a task. We use a linear function (*i.e.*, $0.01 + \delta_{k,l}$) to calculate a worker' cost for illustrative purposes (the results for other costs functions are broadly similar).

As Opt-KG and DI-Greedy-MUL require the prior distributions of $\theta_n, n \in \mathcal{N}$ and $\boldsymbol{\delta}_k, k \in \mathcal{K}$ to be a beta distribution, we assume a $Beta(1,1)$ prior distribution for each θ_n and a $Beta(4,1)$ prior distribution for each $\boldsymbol{\delta}_k$ for illustrative purpose. Other priors lead to similar results and are thus omitted here. For a fair comparison, we let $\theta_n^0 = 1/2 = 0.5$ and $\delta_{k,l}^0 = 4/5 = 0.8$ as the prior information in INCARE. We run each scheme 1000 times and report the mean of the accuracy given a limited payment.

Figure 1 illustrates the achieved accuracy with different budgets. Here, accuracy is defined as the percentage of tasks whose true answers are inferred correctly by the mechanism. In this figure, the error bars are too small to be visible. As we can see, the accuracy achieved by INCARE is higher than that of other policies under the same limited budget. Compared to DI-Greedy-MUL and Opt-KG which both treat a worker's quality as a single level, the improvement regarding the accuracy is at least 15% when the initiator's budget is high (i.e., $B = 1300$) and 75% when the initiator has a small amount of money (i.e., $B = 300$). This is because INCARE chooses task-worker pairs more wisely by considering workers' varying abilities for different tasks. As DI-Greedy-MUL elicits every worker's true cost and incentivizes workers to participate, it performs better than Opt-KG which only pays the average cost for every selected worker and fails to attract workers that perform the

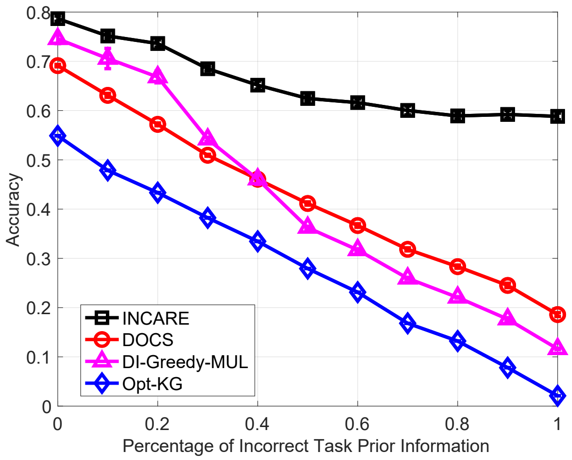

Figure 2: The accuracy vs. different percentages of incorrect task prior information.

tasks to a high standard. Compared to DOCS which also exploits workers' diverse qualities, INCARE improves the outcome accuracy by up to 40%. Because INCARE can pay the selected worker based on his real cost and incentivize all the selected workers to participate, it utilizes the limited budget wisely to achieve higher quality outcomes. Figure 1 also shows that DOCS outperforms DI-Greedy-MUL only when the budget is small ($B \leq 500$). This is because with a limited budget, it is more important to exploit workers' diverse qualities so that appropriate workers can be recruited for performing tasks across various categories. With the increased budget, the initiator has sufficient money to learn information about tasks by recruiting more workers. In such cases, incentivizing workers' participation is more important. As DOCS only pays selected worker the average cost, it fails to incentivize high-quality workers to participate. Hence, the performance of DOCS is worse than DI-Greedy-MUL.

Having shown INCARE is cost efficiency, we now investigate how robust it is for different degrees of accuracy of prior information about each task's difficulty level $\theta_n, n \in \mathcal{N}$.

In this simulation, we let the percentage of incorrect workers prior information be 30% and fix the budget $B = 1300$ for illustrative purposes (other values give broadly similar results). Note that in many practical crowdsourcing systems, the initiator can know workers' historical performance so that she can keep the percentage of incorrect workers' prior information at a relatively low level. We compare the different mechanisms with different percentages of incorrect prior information about the tasks' difficulty level, where 0 represents the situation that the initiator knows each task's difficulty $\theta_n, n \in \mathcal{N}$ perfectly and 1 denotes totally incorrect information[8]. We run the simulation 1000 times and calculate the accuracy based on the average value in these 1000 runs.

Figure 2 illustrates the accuracy with different percentages of incorrect prior information which ranges from 0 to 1. The error bars are again too small to be visible. As we can see, INCARE is more robust to inaccurate prior information about the task's difficulty level than the other mechanisms. Even with a totally incorrect prior distribution, INCARE still achieves 58% accuracy. This is because INCARE elicits every worker's true cost and so can utilize its payments wisely to identify and incentivize the appropriate workers to correctly update the information about the tasks' difficulty levels. As DOCS pays every selected worker the average cost, it fails to recruit workers of high quality and so progresses less on the task at hand. With the increased percentage of incorrect task prior information, DOCS does not have a sufficient number of high-quality workers to correct the information about the task difficulty, which results in the low accuracy outcome.

Figure 2 also shows that the performance of both DI-Greedy-MUL and Opt-KG highly depend on the task prior information. This is because they both use the variational approximation with a moment matching technique to update the tasks' and workers' information simultaneously, where the first and second order moment of tasks' and workers' prior information couple together. Recall that 70% of prior information about workers' qualities are in line with workers' true qualities. When the tasks' prior information is consistent with the actual situation, both mechanisms can update information about tasks' difficulty levels and workers' qualities correctly and the coupling relationship can improve the accuracy. However, with less accurate prior information about tasks, these two mechanisms update the first and second order moment of workers' information incorrectly based on the incorrect first and second order moment of tasks' information. Here, the coupling relationship enlarges the impact of inaccurate prior information and degrades the performance of both DI-Greedy-MUL and Opt-KG significantly. Compared to Opt-KG, DI-Greedy-MUL incentivizes workers' participation and so the performance degradation of DI-Greedy-MUL is lower than that of Opt-KG.

6 CONCLUSIONS

We present a new mechanism that can efficiently deal with crowdsourced task-worker allocations under budget constraints. By incentivizing workers to report their costs truthfully and considering workers' diverse abilities across different task categories, INCARE produces significant savings for the initiator. We believe that only by dealing with the interdependencies between workers, tasks, budgets and incentives can we make steps towards developing practical, yet theoretically well-founded techniques for crowdsourcing systems. INCARE is the first such mechanism. Our next step is to extend the tasks to more complicated crowdsourced activities such as text editing and semantic analysis, where workers need to provide subjective information rather than just choosing one answer from multiple choices. Exploring these complicated settings will lead to a different formulation and method that will further the scope of crowdsourcing systems that have strong theoretical underpinning. Furthermore, we plan to extend INCARE to a highly dynamic setting where workers can receive and perform tasks at any time online. Due to the unpredicatable arrival of workers, designing a budget allocation mechanism that anticipates unknown workers requires further study.

[8] When the task n's true difficulty is θ_n, we set $\theta_n^0 = 1 - \theta_n$ as the incorrect prior information about task n. As Opt-KG and DI-Greedy-MUL require the prior distributions to be a beta distribution, we let the value of θ_n^0 be the mean of a Beta distribution and select the corresponding parameters for this Beta distribution (*e.g.*, $\theta_n^0 = 0.8$ corresponds to a $Beta(4, 1)$ prior distribution).

REFERENCES

[1] Xi Chen, Qihang Lin, and Dengyong Zhou. 2015. Statistical decision making for optimal budget allocation in crowd labeling. *JMLR* 16 (2015), 1–46.

[2] Jeffrey S Chrabaszcz, Joe W Tidwell, and Michael R Dougherty. 2017. Crowdsourcing prior information to improve study design and data analysis. *PloS one* 12, 11 (2017), e0188246.

[3] Ju Fan, Guoliang Li, Beng Chin Ooi, Kian-lee Tan, and Jianhua Feng. 2015. icrowd: An adaptive crowdsourcing framework. In *ACM SIGMOD*. 1015–1030.

[4] Peter I Frazier, Warren B Powell, and Savas Dayanik. 2008. A knowledge-gradient policy for sequential information collection. *SICON* 47, 5 (2008), 2410–2439.

[5] Drew Fudenberg and Jean Tirole. 1991. *Game theory.* Technical Report. The MIT press.

[6] George Iosifidis, Lin Gao, Jianwei Huang, and Leandros Tassiulas. 2014. Enabling crowd-sourced mobile internet access. In *INFOCOM*. 451–459.

[7] Panagiotis G Ipeirotis, Foster Provost, Victor S Sheng, and Jing Wang. 2014. Repeated labeling using multiple noisy labelers. *Data Mining and Knowledge Discovery* 28, 2 (2014), 402–441.

[8] Chao Liu and Yi-Min Wang. 2012. TrueLabel+ confusions: a spectrum of probabilistic models in analyzing multiple ratings. In *ICML*. 17–24.

[9] Roger B Myerson. 1979. Incentive compatibility and the bargaining problem. *Econometrica* (1979), 61–73.

[10] Chenxi Qiu, Anna Squicciarini, and Benjamin Hanrahan. 2019. Incentivizing Distributive Fairness for Crowdsourcing Workers. In *AAMAS*. 404–412.

[11] Chenxi Qiu, Anna Squicciarini, Dev Rishi Khare, Barbara Carminati, and James Caverlee. 2018. CrowdEval: A Cost-Efficient Strategy to Evaluate Crowdsourced Worker's Reliability. In *AAMAS*. 1486–1494.

[12] Anshuka Rangi and Massimo Franceschetti. 2018. Multi-armed bandit algorithms for crowdsourcing systems with online estimation of workers' ability. In *AAMAS*. 1345–1352.

[13] Yaron Singer. 2010. Budget feasible mechanisms. In *FOCS*. 765–774.

[14] Yaron Singer and Manas Mittal. 2013. Pricing mechanisms for crowdsourcing markets. In *ACM WWW*. 1157–1166.

[15] Adish Singla and Andreas Krause. 2013. Incentives for privacy tradeoff in community sensing. In *HCOMP*.

[16] Yuyin Sun, Adish Singla, Dieter Fox, and Andreas Krause. 2015. Building Hierarchies of Concepts via Crowdsourcing.. In *IJCAI*. 844–853.

[17] Nhat VQ Truong, Sebastian Stein, Long Tran-Thanh, and Nicholas R Jennings. 2018. Adaptive Incentive Selection for Crowdsourcing Contests. In *AAMAS*. 2100–2102.

[18] Jun Wu, Yuan Zhang, Yu Qiao, Lei Zhang, Chongjun Wang, and Junyuan Xie. 2019. Multi-unit Budget Feasible Mechanisms for Cellular Traffic Offloading. In *AAMAS*. 1693–1701.

[19] Jing Xie and Peter I Frazier. 2013. Sequential bayes-optimal policies for multiple comparisons with a known standard. *IINFORMS OR* 61, 5 (2013), 1174–1189.

[20] Qi Zhang, Yutian Wen, Xiaohua Tian, Xiaoying Gan, and Xinbing Wang. 2015. Incentivize crowd labeling under budget constraint. In *INFOCOM*. 2812–2820.

[21] Dong Zhao, Xiang-Yang Li, and Huadong Ma. 2014. How to crowdsource tasks truthfully without sacrificing utility: Online incentive mechanisms with budget constraint. In *IEEE INFOCOM*. 1213–1221.

[22] Yudian Zheng, Guoliang Li, and Reynold Cheng. 2016. Docs: a domain-aware crowdsourcing system using knowledge bases. *VLDB* 10 (2016), 361–372.

[23] Yao Zhou and Jingrui He. 2016. Crowdsourcing via Tensor Augmentation and Completion.. In *IJCAI*. 2435–2441.