TOWARDS SECURITY AWARE CROWDSOURCING

by
MINGYAN XIAO

Presented to the Faculty of the Graduate School of
The University of Texas at Arlington in Partial Fulfillment
of the Requirements
for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT ARLINGTON
August 2022

To my family

ACKNOWLEDGEMENTS

First of all, I would like to thank my supervisor, Dr. Ming Li, for her guidance and encouragement through my PhD study. Her invaluable advice and rich experience in research is extremely helpful to my PhD study and will continue enlightening me in future academic career. She also offers a lot of helps in my daily life. She is not only a supervisor, but also a role model and a friend of mine. I respect and admire her pursuit of perfection in research, creative ideas, and dedication to students.

I also want to thank my committee members Dr. Hao Che, Dr. Chengkai Li and Dr. Shirin Nilizadeh, for serving on my supervisory committee and offering all the assistance in different stages of my PhD study.

I want to thank all the labmates in the MobiSec group for providing me unconditional help in both research and daily life. They gave me valuable advice and consolation when I was faced with challenges and difficulties. They are Huadi Zhu, Srinivasan Murali, Chaowei Wang, Youngtak Cho, and graduated group members Wenqiang Jin and Tianhao Li.

Lastly, but most importantly, I would like to express my deep gratitude to my parents and my beloved boyfriend who always encourage, console and support me when I need them.

August, 2022

ABSTRACT

TOWARDS SECURITY AWARE CROWDSOURCING

Mingyan Xiao
The University of Texas at Arlington, 2022

Supervising Professor: Dr. Ming Li

Crowdsourcing has emerged as a novel problem-solving paradigm, which facilitates addressing problems by outsourcing them to the crowd. The openness of crowdsourcing renders it vulnerable to misbehaving workers that impair data trustworthiness. They may attempt to submit calibrated data/parameters to manipulate crowdsourcing outcomes for higher beneficial gain. Those misbehaviors would infringe crowdsourcing's process and, overall, its usefulness.

In this dissertation, I intend to secure the crowdsourcing platform from worker's untrustworthy data reporting. The main contributions are mainly threefold. First, we secure task allocation, an essential but vulnerable stage in crowdsourcing, from individual misreporting. To be specific, misbehaving workers may manipulate task allocation outcomes by uploading falsified parameters. Under the framework of incentive mechanism design, we propose a defense scheme that obtains accurate task allocation outcomes even with workers' manipulated parameters. Second, we further consider workers' collusive behaviors in the stage of task allocation. Strategic workers may form coalitions and rig their parameters together to game the system for extra benefit. To suppress collusion, we leverage incentive mechanism design to calibrate proper payment, leaving workers limited motivation to collude. Third, in addition to task allocation, we also investigate the misbehaviors

from strategic workers in the stage of answer collection. A unified framework is developed to protect these two stages from workers' strategic manipulation simultaneously. Our approach still falls into the category of incentive design. Payment rule is carefully designed, such that workers gain more for truth-telling. It thus motivates workers to honestly report genuine data and parameters in both stages.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

Over the last decade, there existed a wide spectrum of crowdsourcing applications. For example, the machine learning community uses crowdsourcing to build training datasets of image labelling, object recognition, video tagging, etc [51]. The success of those applications have driven a series of industrial successes, including micro-task crowdsourcing markets, e.g., Amazon Mturk [129], P2P ride-sharing, e.g., Uber [89], real-time online-to-offline service, e.g., GrubHub [86], and citizen sensing service, e.g., Waze [84].

The effectiveness of crowdsourcing typically relies on the participation of a great number of human workers. In the case of Amazon Mturk, it is reported that there are 250,810 MTurk workers worldwide who have completed at least one Human Intelligence Task (HIT) posted through the TurkPrime platform. Basically, any Internet user can register as a Mturk worker. On the other hand, the very openness which allows anyone to participate, also exposes crowdsourcing to unreliable data and parameters reported by untrustworthy workers. Empirical study has shown that, albeit being banned, parameter manipulation widely exists in crowdsourcing and has a significant impact [52]. In particular, parameter manipulation refers to workers, as both individuals or in a coalition, uploading falsified parameters to manipulate crowdsourcing's task allocation outcomes to increase their benefit. Besides parameter manipulation, misbehaving workers may also report falsified task answers in the answer collection stage to maximize their benefits [135]. Conventional security approaches like encryption or digital signature are ineffective, as the misbehaving workers alter the parameters before they are encrypted and signed. Prior efforts that discard untrustworthy data by detections or reputation systems either rely on assumptions that workers adopt consistent misreport strategies [4, 5, 92], or knowledge of ground truth/historical data [107] that may not be universally available.

In this dissertation, I aim to secure crowdsourcing from untrustworthy data/parameter reporting. The rest of the dissertation is organized as follows.

In Chapter 2, I seek to defend parameter manipulation conducted by individual workers. Crowdsourcing systems usually propose task allocation optimization problems to boost effective resource utilization. In an ideal scenario, workers are trusted to report their accurate parameters, e.g., bids, to the platform, so that task allocation optimization problems can be correctly formulated and calculated. Nonetheless, strategic workers can explore illegal benefit gain by simply uploading falsified parameters. Even worse, such misbehavior

is difficult to detect. To tackle this issue, we ask each worker to report its tunable "indicator" instead of original parameters. We construct a surrogate task assignment optimization problem based on those "indicators". If, with a careful design of "indicators", the optimal solution of the surrogate optimization and the original one are identical, we can bypass the original problem but get its optimal solution by solving the surrogate problem. Then, the problem becomes how to elicit workers to offer our desired "indicators". We design an effective payment rule, under which the worker's utility is maximized when it reports desired "indicators". The proposed scheme is evaluated through simulation studies.

Chapter 3 investigates how to resist parameter manipulation when multiple workers collude. Designing task allocation that discourage workers from cheating and instead encourage them to reveal their true cost information has drawn significant attention. However, the existing efforts have been focusing on tackling individual cheating misbehaviors, while the scenarios that workers strategically form collusion coalitions and rig their parameters, e.g., bids, together to manipulate auction outcomes have received little attention. To fill this gap, in this work, we develop a (t, p)-collusion resistant scheme that ensures no coalition of *weighted cardinality t* can improve its group utility by coordinating the bids at a probability of p. This work takes into account the unique features of crowdsourcing, such as diverse worker types and reputations, in the design. The proposed scheme can suppress a broad spectrum of collusion strategies. Besides, desirable properties, including p-truthfulness and p-individual rationality, are also achieved. To provide a comprehensive evaluation, we first theoretically prove our scheme's collusion resistance and then experimentally verify our analytical conclusion using a real-world dataset.

In addition to the stage of task allocation, Chapter 4 further secures the stage of data collection. In this stage, workers are required to submit their allocated tasks answers to the platform. Misbehaving workers can possibly alter their answers, deteriorating the accuracy of tasks. So far, the existing works treat parameter misreporting (in the task allocation stage) and answer misreporting (in the answer collection stage) separately. Instead, we aim to develop a unified framework that simultaneously protects two different stages from workers' strategic manipulation. We plan to leverage incentive design to motivate workers to honestly reveal both task answers and their parameters. For this, we first derive the sufficient and necessary conditions for answer truthfulness and parameter truthfulness separately. Particularly, to achieve answer truthfulness, we leverage reference answers to evaluate the truthfulness of a given worker's answer. Under the model of Bayesian game, a worker's expected payment for truthtelling is set no less than that when lying, which leaves workers little incentive to lie. The condition of parameter truthfulness is derived via fractional VCG that ensures bids truthfulness. We then construct a payment optimization problem incorporating these conditions as constraints. Its optimal solution lists the pay-

ment to each worker that is expected to elicit answers and parameters jointly. Our proposed mechanism, with a formally proved bounded approximation ratio, ensures that truthtelling is a Bayesian Nash equilibrium. We prototype the mechanism and conduct a series of experiments that involve 30 volunteers to validate the efficacy and efficiency of the proposed mechanism.

Finally, Chapter 5 concludes the dissertation.

CHAPTER 2

SECURING ALLOCATION IN MOBILE CROWD SENSING: AN INCENTIVE DESIGN APPROACH

2.1 Introduction

Mobile crowd sensing (MCS) arises as a new sensing paradigm by exploring a plethora of embedded multi-modal sensors in today's ubiquitous mobile devices. By fusing and analyzing their sensing data, MCS has potential to accelerate the maturity of smart health caring, environment monitoring, traffic surveillance, social event observation, etc. An MCS system mainly consists of three types of entities, task requestors, sensing workers, and the platform. A typical MCS workflow can be divided into three stages: *task allocation*, *task sensing* and *data aggregation/analysis*.

For the stage of task allocation, its main objective is to distribute sensing tasks among workers such that sensing resources of the entire system are efficiently utilized. For this purpose, prevalent approaches are to formulate and solve the task allocation optimization problems at the platform, taking into account various optimization factors from all entities, e.g., [1, 58, 3]. Typically, workers are required to explicitly specify their parameters, including task preferences, computation capacities, affordable travel distances, cost functions. The accuracy of task allocation thus directly relies on the quality of these reported parameters. As pointed out later in this chapter, malicious workers can easily manipulate task allocation of MCS by simply uploading falsified parameters. In this work, we name this type of attack as the *parameter manipulation attack*. Since these parameters are personal data owned by each worker, there is lack of evidence at the platform to decide their accuracy. As a result, parameter manipulation attacks will be difficult to detect.

While there have been some existing works tackling security issues in MCS, most of them focus on the stage of data analysis. Since MCS allows any voluntary participant to contribute data, it is vulnerable to erroneous or even malicious data injection. Great efforts have been devoted to the development of a so-called "reputation system" [4, 5, 6, 7, 8, 9]; they initially establish reputation scores of workers based on the quality of their contributions, and later on use these scores to eliminate reports from less reputable workers.

Instead of tackling data trustworthy issues in the stage of data analysis as the above works[4, 5, 6, 7, 8, 9], in this work, for the first time, we target at the parameter manipulation attack in the stage of task allocation in MCS. To defend against it, *our design goal* is to enable the platform to find accurate and optimal solutions to task allocation formulations,

even under the existence of malicious workers. As an initial work on this topic, we plan to start from a simplified case: workers report falsified objective function (parameters) while the rest parameters, i.e., the ones in constraints, are unaltered. Since the objective function is critical to task allocation and of complex form, it can easily become the adversary's target. However, even under this simplified case, the scheme design is not an easy task.

Since solving task allocation optimization problems alone is already complicated in general, it is infeasible to refer to computationally intensive cryptographic techniques for the defense scheme design. Instead, we leverage the *incentive mechanism*. Our scheme is built on top of an assumption that all workers are *rational* and *self-interest* in a sense that they launch an attack only to maximize their benefits achieved during task sensing. Since workers are skeptical to report genuine parameters for task allocation formulation, instead of collecting them, our scheme asks each worker to submit its tunable "indicator" of willingness for executing tasks. Then a new task allocation formulation, based on collected indicators, is constructed, which is slightly different from the original one. On the other hand, if the new problem's optimal solution is the same with that of the original one, the platform can still derive the accurate task allocation profile via the new problem. However, the challenge is how to make the optimal solutions of these two distinguishing problems identical. As pointed out later, it depends on indicator values chosen by workers. Incentive mechanism design is thus applied to elicit workers to offer "proper indicators" so as to achieve the goal.

We summarize the contributions of this work as follows.

- We address the parameter manipulation attack in the stage of task allocation in MCS, where workers report falsified parameters to for illegal beneficial gain. It is a critical security issue in MCS, yet receives rare attention so far.
- Instead of crypto primitives, we novelly apply the incentive mechanism in our defense scheme development. It guarantees that the platform can still find the accurate task allocation solution, even under the existence of malicious workers.
- We formally prove the security and convergence property of the proposed scheme. A real-world dataset is applied to evaluate the scheme performance.

2.2 Problem Statement

2.2.1 Background

The discussion of this work pertains to a standard MCS system, which mainly consists of a platform, a set of task requesters, and a set of participating sensing workers $\mathscr{U} = \{u_1, \cdots, u_i, \cdots, u_M\}$, who communicate with the platform via wireless connections, such as cellular networks or Wi-Fi. The platform hosts a set of sensing tasks $\mathscr{T} = \{\tau_1, \cdots, \tau_j, \cdots, \tau_N\}$

that are collected from their requesters. Conducting sensing tasks are resource-consuming for workers. Hence, to wisely utilize their resources, a task allocation framework is needed to coordinate among workers until task completion. Typically, optimization problems are formulated, taking into account of constraints from sensing capabilities and travel budget at workers, and sensing quality requirement from sensing tasks. Without loss of generality, in this work we consider a *cost minimization problem*[1]

$$
\begin{aligned}
\mathrm{P}_1: \quad \min_{x_i} \quad & \sum_{i\in[1,M]} C_i(x_i) \\
s.t. \quad & \sum_{j\in[1,N]} x_{ij} \le t_i, \quad \forall i \in [1,M] && (2.1)\\
& \sum_{i\in[1,M]} \theta_{ij} x_{ij} \ge T_j, \quad \forall j \in [1,N] && (2.2)\\
& \sum_{j\in[1,N]} \frac{d_{ij}}{w_{ij}\left(x_{ij}\right)} \le D_i, \quad \forall i \in [1,M] && (2.3)\\
& x_{ij} \ge 0, \quad \forall i \in [1,M], \forall j \in [1,N]. && (2.4)
\end{aligned}
$$

The decision variable x_{ij} ($i \in [1,N]$, $j \in [1,M]$) stands for the sensing time that worker u_i is assigned for task τ_j and x_i is a vector $x_i = \{x_{i1}, x_{i2}, \cdots, x_{iN}\}$. $C_i(\cdot)$ is u_i's component in the cost function and can be of different meanings. For example, in the case of minimizing the payments to workers[1, 58], it is u_i's pricing function; in the case of minimizing the overall travel distance[3], it stands for u_i's travel distance given a specific set of allocated tasks x_i. In a word, $C_i(\cdot)$ is used to represent certain "burden" task sensing causes to worker u_i, and we aim to minimize the overall "burden" from all workers. Following the prevalent setting, we let $C_i(\cdot)$ be a convex function.

Denote by t_i the maximum sensing time that u_i can contribute. Constraint (2.1) states that each worker's total sensing time for all tasks cannot surpass this limit. For a task τ_j, we denote by T_j its minimum sensing time requirement. Constraint (2.2) says that the accumulated weighted sensing time from all workers regarding τ_j should not be less than T_j. Moreover, to carry out a task τ_j, u_i has to travel for a distance d_{ij} from its current location. Let D_i be u_i's maximal distance it is willing to travel. (2.3) says that each worker's total weighted travel distance should be no larger than D_i. $w_{ij}(x_{ij})$ is the weight and calculated by $w_{ij}(x_{ij}) = \frac{\theta_{ij}}{x_{ij}}$. A larger weight $w_{ij}(\cdot)$ implies that a worker w_i is more willing to conduct task τ_j since it can contribute with higher quality but cost shorter sensing time. The idea of (2.3) is taken from [17]. Note that the formulation of P_1 is not the contribution of this

[1]Our scheme also works for the maximization case with mild modification.

work. Once the platform formulates P_1, it can apply its favorite algorithms to solve it. The solution of x_{ij}'s is the final task allocation policy.

2.2.2 Adversary Model

In this work, we target at the parameter manipulation attack in the stage of task allocation in MCS. The malicious workers submit the falsified parameters, i.e., the cost functions, to the platform to manipulate the task allocation outcome, and thus receive illegal beneficial gain. As a result, those from others, including benign workers and the platform, may get harmed. Following a conventional approach, here we leverage "utility" to measure a worker's benefit received in MCS. It is defined as $U_i = P_i(x_i) - C_i(x_i)$, where x_i is u_i's allocated task set and $P_i(x_i)$ is its corresponding payment.

Table 2.1: A toy example.

Parameters				
$T_1 = 3$	$t_1 = 2$ $t_2 = 3$	$d_{11} = 150$ $d_{21} = 100$	$\theta_{11} = 0.7$ $\theta_{21} = 0.8$	$D_1 = 3000$ $D_2 = 2000$
Without attack				
$C_1(x) = 0.1 \cdot e^{0.7x_{ij}}$ $C_2(x) = 0.1 \cdot e^{0.8x_{ij}}$		$x_{11}^* = 3.8$ $x_{21}^* = 0.4$	$U_{11} = 0.1$ $U_{21} = 0.1$	
With attack				
$C_1'(x) = 0.1 \cdot e^{5x_{ij}}$ $C_2(x) = 0.1 \cdot e^{0.8x_{ij}}$		$x_{11}' = 0.9$ $x_{21}' = 3.0$	$U_{11}' = 8.9$ $U_{21}' = 0.1$	

We now use a toy example to better illustrate the parameter manipulation attack. All the system parameters are provided in Table 2.1. Under the attack-free scenario, the allocated sensing time to u_1 and u_2 is $x_{11}^* = 3.8$ and $x_{12}^* = 0.4$, respectively, through optimally solving P_1. Without loss of generality, we set the payment $P_i(x_i) = C_i(x_i) + 0.1$, i.e., each worker gets paid by 0.1 more than its actual cost. Under this setting, we calculate the utility of u_1 and u_2 as $U_{11} = U_{21} = 0.1$. Now, if worker u_1 is malicious and changes ρ_{11} in its cost function to 5, then x_{11}' and x_{21}' become 0.9 and 3.0, respectively. Accordingly, U_{11}' turns to 8.9, which is significantly larger than U_{11} (U_{21}' stays at 0.1 unchanged). Thus, not only does the parameter manipulation attack benefit malicious workers with illegal gain, but also compromises interest of other entities, i.e., the platform has to pay much more than it should.

In this work, malicious workers are modeled as rational and self-interest. When launching a parameter manipulation attack, a malicious worker chooses the strategy that brings itself the greatest benefit. In another word, its objective is solely to maximize its own utility. Therefore, it distinguishes from the attacker who aims to sabotage system

operations. These more aggressive attackers are not the focus of this work. It is worth noting that the rational and self-interest attacker is a widely adopted attack model in game theoretical approaches to tackling security problems, such as attack-defense analysis [18] and security/dependability measurement [19].

2.3 Our Proposed Scheme

2.3.1 KKT Conditions of P_1

Before digging into details of the scheme, we first briefly go through the KKT conditions of P_1, which play a critical role in the scheme design.

In P_1 introduced in Section 4.2.1, its objective and constraint functions are convex. Hence, it admits a unique optimal solution that can be characterized using the necessary and sufficient Karush-Kuhn-Tucker (KKT) conditions[20].

We first derive the *Lagrangian* of P_1 as follows

$$L(\lambda,\mu,\nu,x) = \sum_{i\in[1,M]} C_i(x_i) + \sum_{i\in[1,M]} \lambda_i \cdot (\sum_{j\in[1,N]} x_{ij} - t_i) + \sum_{j\in[1,N]} \mu_j (T_j - \sum_{i\in[1,M]} \theta_{ij} x_{ij}) + \sum_{i\in[1,M]} \nu_i (\sum_{j\in[1,N]} d_{ij}\frac{x_{ij}}{\theta_{ij}} - D_i)$$

where $\lambda \triangleq \{\lambda_i \geq 0 : \forall i \in [1,M]\}$, $\mu \triangleq \{\mu_j \geq 0 : \forall j \in [1,N]\}$, and $\nu \triangleq \{\nu_i \geq 0 : \forall i \in [1,M]\}$ are the vectors of Lagrange multipliers corresponding to constraints (2.1), (2.2), and (2.3), respectively. The KKT conditions that produce the optimal dual solution λ°, μ° and ν°, and the optimal primal solution x° for P_1 are given by the following set of equations $\forall i \in [1,M], \forall j \in [1,N]$,

$$\frac{\partial C_i(x_i{}^\circ)}{\partial x_{ij}} = \mu_j^\circ \theta_{ij} - \frac{\nu_i^\circ d_{ij}}{\theta_{ij}} - \lambda_i^\circ, \tag{2.5}$$

$$\lambda_i^\circ \cdot (\sum_{j\in[1,N]} x_{ij}^\circ - t_i) = 0, \tag{2.6}$$

$$\mu_j^\circ \cdot (T_j - \sum_{i\in[1,M]} \theta_{ij} x_{ij}^\circ) = 0, \tag{2.7}$$

$$\nu_i^\circ \cdot (\sum_{j\in[1,N]} \frac{d_{ij} x_{ij}^\circ}{\theta_{ij}} - D_i) = 0, \tag{2.8}$$

$$x_{ij}^\circ, \lambda_i^\circ, \mu_j^\circ, \nu_i^\circ \geq 0. \tag{2.9}$$

The KKT conditions work well when all workers honestly report their genuine cost functions. However, the platform fails to yield x° by solving (2.5)-(2.9) with any falsified $C_i'(\cdot) \neq C_i(\cdot)$.

2.3.2 Scheme Overview

Our objective is to develop a defense scheme to enable the platform to correctly find out x° even without the correct information of $C_i(\cdot)$. Since workers are suspicious to report manipulated cost functions, rather than having each of them report its cost function $C_i(\cdot)$, it is asked to submit an "indicator" parameter, the willingness of this worker to participate in a specific task. Then we construct a new task allocation optimization problem P_2 (to be presented soon) that takes "indicator" parameters as the coefficients of its objective function, but shares identical constraints with P_1. If, with a carefully designed scheme, the optimal solution of P_2 is the same with that of P_1, x°, then we can bypass P_1 to find x°. It means we no longer need to worry about falsified $C_i'(\cdot)$'s. However, the challenge is how to have the optimal solutions of P_1 and P_2 identical. Apparently, it relies on the indicators submitted by workers. Then the problem becomes how to design a mechanism to elicit workers to offer proper indicators so as to fulfill this goal. Note that malicious workers may still report falsified indicators to manipulate the new problem P_2's task allocation solution.

Inspired by the work [21], we adopt the incentive mechanism to stimulate workers to submit proper indicators. The platform pays workers according to their reported indicators. Since (malicious) workers are rational, they strategically report indicators to maximize their utility. Then, if indicators which produce x°, coincide with the ones which bring worker's maximal utility, then our objective achieves. To summarize, the scheme needs to address two questions. First, what are the values of "proper indicators"? Second, how to elicit all workers to report their "proper indicators"? Next, we are going to answer these two questions in the following two subsections, respectively.

2.3.3 Determination of "Proper Indicators"

Instead of $C_i(\cdot)$, we have each worker u_i submit an indicator vector b_i to the platform, where $b_i \triangleq \{b_{i1}, b_{i2}, \cdots, b_{iN}\}$ and b_{ij} is u_i's indicator value for task $\tau_j \in \mathscr{T}$. A lower value of b_{ij} indicates that u_i is more willing to execute τ_j, while a larger value indicates the less willingness u_i has. As mentioned above, malicious workers may still strategically report their b_i's for illegal utility gain.

Upon receiving all b_i's, the platform formulates an alternative task allocation optimization problem (P_2)

$$\min_{x} \quad \sum_{i\in[1,M]} \sum_{j\in[1,N]} \frac{b_{ij}}{2} x_{ij}^2, \quad s.t. \quad (2.1), (2.2), (2.3) \text{ and } (2.4).$$

P_2 shares the same set of constraints with P_1, but differs in its objective function; P_2 minimizes the overall unwillingness (or maximize the overall willingness), while P_1 minimizes

an overall cost. Besides, P_2's objective function has a much simpler form compared with that of P_1.

Since P_2 has a convex objective and constraint functions, similarly, it admits a unique optimal solution as well. We write the *Lagrange* of P_2 as

$$\tilde{L}(\lambda,\mu,\nu,x) = \sum_{i\in[1,M]}\sum_{j\in[1,N]}\frac{b_{ij}}{2}x_{ij}^2 + \sum_{i\in[1,M]}\lambda_i(\sum_{j\in[1,N]}x_{ij}-t_i) + \sum_{j\in[1,N]}\mu_j(T_j - \sum_{i\in[1,M]}\theta_{ij}x_{ij}) + \sum_{i\in[1,M]}\nu_i(\sum_{j\in[1,N]}d_{ij}\frac{x_{ij}}{\theta_{ij}} - D_i)$$

and denote the optimal primal and dual solutions of P_2 as x^* and λ^*, μ^* and ν^*, respectively. Its corresponding KKT conditions yield a set of equations,

$$x_{ij}^* = \frac{\mu_j^*\theta_{ij} - \frac{\nu_i^* d_{ij}}{\theta_{ij}} - \lambda_i^*}{b_{ij}}, \tag{2.10}$$

$$\lambda_i^* \cdot (\sum_{j\in[1,N]}x_{ij}^* - t_i) = 0, \tag{2.11}$$

$$\mu_j^* \cdot (T_j - \sum_{i\in[1,M]}\theta_{ij}x_{ij}^*) = 0, \tag{2.12}$$

$$\nu_i^* \cdot (\sum_{j\in[1,N]}\frac{d_{ij}x_{ij}^*}{\theta_{ij}} - D_i) = 0, \tag{2.13}$$

$$x_{ij}^*, \lambda_i^*, \mu_j^*, \nu_i^* \geq 0, \tag{2.14}$$

where (2.11)-(2.14) are identical to (2.6)-(2.9) of P_1, while (2.10) differs from (2.5).

Our goal is to have $x^\circ \triangleq x^*$, i.e., the optimal solution of P_1 is identical to that of P_2. Comparing equations (2.10)-(2.14) and (2.5)-(2.9), we observe that if $b_{ij}x_{ij}^* = \frac{\partial C_i(x_i^*)}{\partial x_{ij}}$ then the goal is achieved. Or, equivalently, u_i submits the following indicator for each task τ_j

$$b_{ij} = \frac{1}{x_{ij}^*}\cdot\frac{\partial C_i(x_i^*)}{\partial x_{ij}}. \tag{2.15}$$

Up to now, we have determined exact values of proper indicators b_{ij}'s. When workers submit their indicators following (2.15), the platform can still correctly find out x° by formulating and solving P_2, even without the knowledge of workers' original cost functions.

2.3.4 Elicitation of "Proper Indicators"

Now the remaining issue is how to induce workers to offer indicators strictly following (2.15). This is where the incentive mechanism plays; the platform carefully chooses its payment to workers, so as to elicit them to offer the desirable indicators.

Since each worker is rational and self-interest in a sense to always submit its indicator to maximize its own utility, and thus u_i determines its optimal indicator b_i^* by solving the following utility maximization problem (UMP$_i$)

$$\begin{aligned} \max_{b_i} \quad & P_i(x_i) - C_i(x_i) \\ s.t. \quad & b_{ij} \geq 0, \qquad \forall j \in [1,N]. \end{aligned}$$

The unique optimal solution of the UMP$_i$ meets the following optimality condition

$$\frac{\partial C_i(x_i)}{\partial x_{ij}} = \frac{b_{ij}^2}{\lambda_i + \frac{v_i d_{ij}}{\theta_{ij}} - \mu_j \theta_{ij}} \frac{\partial P_i(x_i)}{\partial b_{ij}}, \ \forall j \in [1,N], \tag{2.16}$$

where we utilize derivative $\frac{\partial x_{ij}}{\partial b_{ij}} = \frac{\lambda_i + \frac{v_i d_{ij}}{\theta_{ij}} - \mu_j \theta_{ij}}{b_{ij}^2}$ derived from (2.10).

Jointly consider (2.5) and (2.16). In order to elicit workers to submit desirable indicator (2.15), a feasible incentive mechanism is to pay u_i with

$$P_i(b_i) = \sum_{j \in [1,N]} \frac{(\lambda_i + \frac{v_i d_{ij}}{\theta_{ij}} - \mu_j \theta_{ij})^2}{b_{ij}}, \tag{2.17}$$

where we express $P_i(b_i)$ as the function of u_i's indicator. Alternatively, we can rewrite $P_i(b_i)$ as the function of u_i's sensing time with the relation (2.10)

$$P_i(x_i) = \sum_{j \in [1,N]} x_{ij}(\mu_j \theta_{ij} - \lambda_i - \frac{v_i d_{ij}}{\theta_{ij}}), \tag{2.18}$$

i.e., the payment to u_i is proportional to its devoted sensing time. (2.18) is intuitive; the more sensing time u_i contributes, the higher payment it receives. Till now, the second question has been answered as well.

Remark I. One may ask rather than eliciting workers to submit proper indicators, why not directly elicit them to submit accurate cost functions? This is because the original cost function (of P$_1$) can be in an arbitrary form. It is extremely difficult to develop an effective incentive mechanism to motivate workers to report the genuine cost functions

when their forms are unknown. With the introduction of indicators, a new problem P_2 is formulated. Due to its fixed and simplified expression of the objective function, it makes the design feasible.

Remark II. It is worth mentioning two relevant but totally different research topics. The first one is to achieve truthfulness (or called incentive compatibility) in auctions. Its objective is to decide winners and their payment such that their best strategy is to submit their true values/costs as bids. The second one is to find out an employee's true *type* in a monopoly market, when such information is unknown to the employer. *Contract theory* has been widely adopted; incentives are provided to employees such that their utilities are maximized when reporting true types. Differently, we do not care about whether workers honestly submit their true cost functions/types or not; instead, we aim to enable the platform to derive accurate task allocation profile even without these information.

2.3.5 Scheme Implementation

With the payment rule (2.17) or (2.18), each worker u_i can compute its optimal indicator by solving UMP_i. Based on collected indicators, the platform then optimally solves P_2 for task allocation. However, UMP_i and P_2 are intertwined with each other. On one hand, u_i has to be aware of the task allocation result x_i to solve UMP_i and derive its indicator. On the other hand, x_i is obtained by the platform via solving P_2, which takes workers' indicators as inputs. Hence, there is a need for an iterative operation that gradually adjusts results of both P_2 and UMP_i to reach the optimum point.

With this in mind, Algorithm 4 outlines our final scheme. The platform first initializes the primal variable x and dual variables λ, μ and ν with their values satisfying KKT conditions (2.11)-(2.14). For example, we can choose $\lambda_i^{(0)} = \mu_j^{(0)} = \nu_i^{(0)} = 0$, with any positive value of $x_{ij}^{(0)}$.

Then the algorithm iteratively computes the primal and dual solutions of P_2 (at the platform) and indicators via UMP_i (at u_i) until convergence. Specifically, the platform first announces dual solutions $\lambda_i^{(t)}$, $\mu_j^{(t)}$, $\nu_i^{(t)}$ of P_2 (line 4). With these values and the knowledge of $C_i(\cdot)$, u_i calculates its optimal indicator $b_i^{(t)}$ by solving UMP_i and submits it to the platform (line 6-7). Then the platform obtains a new allocation rule $x^{(t)}$ by (2.10) (line 8). It also updates dual solutions $\lambda_i^{(t)}$, $\mu_j^{(t)}$ and $\nu_i^{(t)}$ ($\forall i \in [1,M], j \in [1,N]$) by using a gradient descent method (line 9).

Note that $(x)^+$ represents $\max\{x,0\}$. In the end, the platform checks the termination criterion (line 10). When changes of indicators for two consecutive iterations are sufficiently small with $\varepsilon \geq 0$, the iteration terminates. Otherwise, another round of iteration is performed. Once convergence is reached, the solution of $x^{(t)}$ is exactly the optimal solution

Algorithm 1 Our proposed scheme

Output: x^* (x°), $P_i(x_i^*)$, $\forall i \in [1,M]$

1: $t \leftarrow 0$, $conv\_flag \leftarrow 0$;
2: Initialize $x_{ij}^{(0)}$, $\lambda_i^{(0)}$, $\mu_j^{(0)}$, $\nu_i^{(0)}$, $\forall i \in [1,M], j \in [1,N]$;
3: **while** $conv\_flag = 0$ **do**
4: $t \leftarrow t+1$;
5: The platform announces $\lambda_i^{(t)}$, $\mu_j^{(t)}$, $\nu_i^{(t)}$, $\forall i \in [1,M], j \in [1,N]$;
6: Each worker u_i computes its optimal indicator $b_i^{(t)}$ by solving UMP$_i$;
7: Each worker u_i submits its indicator $b_i^{(t)}$ to the platform;
8: The platform computes the new $x^{(t)}$ by (2.10);
9: The platform uses gradient updates for dual variables:

$$\lambda_i^{(t)} = \left(\lambda_i^{(t-1)} + s^{(t)} \cdot (\sum_{j\in[1,N]} x_{ij}^{(t-1)} - t_i)\right)^+, \quad \mu_j^{(t)} = \left(\mu_j^{(t-1)} + s^{(t)} \cdot (T_j - \sum_{i\in[1,M]} \theta_{ij} x_{ij}^{(t-1)})\right)^+,$$

$$\nu_i^{(t)} = \left(\nu_i^{(t-1)} + s^{(t)} \cdot (\sum_{j\in[1,N]} \frac{d_{ij} x_{ij}^{(t-1)}}{\theta_{ij}} - D_i)\right)^+, \quad \forall i \in [1,M], j \in [1,N];$$

10: **if** $\left|\frac{b_{ij}^{(t)} - b_{ij}^{(t-1)}}{b_{ij}^{(t-1)}}\right| < \varepsilon$, $\forall i \in [1,M], j \in [1,N]$ **then**
11: $conv\_flag \leftarrow 1$;
12: **end if**
13: **end while**
14: x_i^* (x_i°) $\leftarrow x_i^{(t)}$, $\forall i \in [1,M]$;
15: The platform computes $P_i(x_i^*), \forall i \in [1,M]$, by (2.18).

x^* and thus x°, i.e., the optimal solution to P_1. Finally, the platform determines the final payment $P_i(x_i^*)$ for each worker with (2.18) (line 15).

Theorem 1. ***(Convergence.)*** *Algorithm 4 converges to the optimal solution of P_2 globally.*

Proof. The formal proof is provided in Appendix 2.7. □

Besides, it is also critical to show that our scheme pays workers properly, without causing them negative utilities, as otherwise workers will be discouraged from participating.

Proposition 1. ***(Non-negative Utility.)*** *Each worker $u_i \in \mathscr{U}$ receives a nonnegative utility via our scheme, i.e.,*

$$P_i(x_i^*) - C_i(x_i^*) \geq 0. \tag{2.19}$$

Proof. The formal proof is provided in the technical report [28]. □

2.4 Enhancing the Convergence Speed

The effectiveness of our defense scheme is in trade of extra interactions between workers and the platform (until Algorithm 1 converges). As a result, calculation delay will be caused in deriving the final task allocation profile. To tackle this side effect, we plan to further accelerate the algorithm convergence speed.

The value of $s^{(t)}$ plays a critical role in the convergence speed of Algorithm 4. Basically, a large value implies a large step size in each iteration toward the optimal solution. However, it may cause oscillation in the algorithm. On the other hand, a small $s^{(t)}$ may lead to extra iterations, and thus a longer running time. Hence, in this section we propose to enhance the convergence speed of Algorithm 4 via adaptive selection of $s^{(t)}$ in each iteration. We adopt the backtracking line search [23], an efficient online search method used in unconstrained convex optimization. Its idea is to determine the maximum step size to move along a given search direction to find the optimal result. Since it operates over unconstrained convex optimization problems, we first transfer P_2 into an unconstrained form. The Lagrange dual function of P_2 is

$$g(\lambda,\mu,\nu) = \inf_{x} \tilde{L}(\lambda,\mu,\nu,x)$$
$$= \sum_{i\in[1,M]} \sum_{j\in[1,N]} \left[\frac{\theta_{ij} f^2 + 2\theta_{ij}\lambda_i f - 2\theta_{ij}^2 \mu_j f + 2\nu_i d_{ij} f}{2 b_{ij}\theta_{ij}} \right] - \sum_{i\in[1,M]} (\lambda_i t_i + \nu_i D_i) + \sum_{j\in[1,N]} \mu_j T_j,$$

where $f = \mu_j \theta_{ij} - \frac{\nu_i d_{ij}}{\theta_{ij}} - \lambda_i$. According to the backtracking line search, our objective is to find the optimal length to maximize the above Lagrange dual function.

Algorithm 2 The selection of suitable s

Input: b_i, $\Delta(\lambda,\mu,\nu)$, α, β
Output: s

1: $s \leftarrow 1$
2: **while** $g((\lambda,\mu,\nu) + s\Delta(\lambda,\mu,\nu)) < g(\lambda,\mu,\nu) + \alpha \cdot s \cdot \nabla g\left(\lambda_i,\mu_j,\nu_i\right)^T \Delta(\lambda,\mu,\nu)$ **do**
3: $\quad s \leftarrow \beta s$.
4: **end while**
5: return s;

The inputs of Algorithm 5 include b_i, the descending direction $\Delta(\lambda,\mu,\nu) = (\frac{d\lambda_i}{dt}, \frac{d\mu_j}{dt}, \frac{d\nu_i}{dt})$ for $g(\lambda,\mu,\nu)$, where $\lambda_i,\mu_j,\nu_i \geq 0$ ($i \in [1,M], j \in [1,N]$), and two predefined constants α and β. The output is the optimal line length s. It starts with a rough estimate, i.e., $s = 1$ (line 1). Then we iteratively adapt the step length (line 3) as long as the criterion

in line 2 holds, where $\nabla g\left(\lambda_i, \mu_j, \nu_i\right)$ represents the local gradient of function g. Note that $g\left((\lambda, \mu, \nu) + s\Delta(\lambda, \mu, \nu)\right) = g\left(\lambda, \mu, \nu\right) + \alpha \cdot s \cdot \nabla g\left(\lambda_i, \mu_j, \nu_i\right)^T \Delta(\lambda, \mu, \nu)$ takes place at the optimal point of g. Following the standard approach in the backtracking line search, we set $\alpha \in (0, 0.5)$ and $\beta \in (0, 1)$.

Once the suitable s is identified via Algorithm 5, it will be used to update $s^{(t)}$ in each iteration of Algorithm 4.

2.5 Performance Evaluation

In this section, we conduct extensive simulations to evaluate the performance of our scheme. The study is based on the Yelp Dataset Challenge [24]. The data is sampled by Yelp from the greater Phoenix, AZ metropolitan area from March 2005 to January 2013. This dataset includes 11537 businesses, 229907 reviews by 229907 users, and 8282 check-in sets in the form of separate JSON or SQL files. Specifically, our evaluation selects a set of Yelp users as workers in the MCS system. We then take the distance between two consecutive check-in locations from the same user as the worker's travel distance for one task and the corresponding time interval in between as the worker's maximum sensing time it can contribute. Note that this dataset has been widely used in many other crowd/social sensing related research, such as [25, 26].

We assume that each worker u_i holds its convex objective function $C_i\left(x_i\right) = 0.1 \cdot \sum_{j \in [1,N]} e^{\rho_{ij} x_{ij}}$ where ρ_{ij} is randomly chosen from $[0.5, 1]$. For the sensing quality θ_{ij}, it is a random value from $[0, 1]$. Besides, we set the termination condition $\varepsilon = 10^{-5}$, i.e., the algorithm terminates if the gap between two consecutive iterations is less than 10^{-5}. Each simulation result is the average over 20 trials.

Effectiveness of Our Scheme. We start from a small-scale MCS with $M = 2$ workers and $N = 2$ tasks. Besides, u_2 is assumed as the malicious worker to manipulate the task allocation outcome. Performances of our scheme, in terms of security and convergence property, have been examined.

Fig. 2.1(a) shows utility of u_1 and u_2 under three scenarios, i.e., without attack (U_{11}^*, U_{21}^*), with attack (U_{11}', U_{21}'), with attack but protected by our scheme (U_{11}, U_{21}). Once the algorithm converges, we observe that $U_{21}' > U_{21}^*$. It means the attacker u_2 gains extra utility by launching the parameter manipulation attack. However, such a utility gain diminishes once our scheme is implemented. Specifically, U_{21} is equal to U_{21}^*. Meanwhile, $U_{11}' < U_{11}^*$, i.e., the benign worker u_1's utility is harmed by u_2's parameter manipulation attack. However, it can be fully defended by our scheme, as $U_{11}^* = U_{11}$.

We further depict in Fig. 2.1(b) worker's total allocated sensing time under the three scenarios same as above. With the implementation of our scheme, we observe $x_{11} = x_{11}^* =$

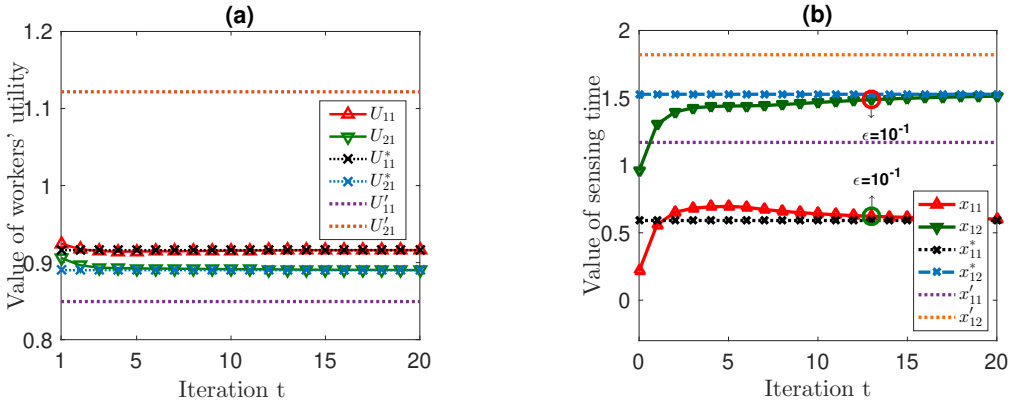


Figure 2.1: Task allocation outcome comparison under different scenarios, i.e., without attack, with attack, with attack but protected by our scheme.

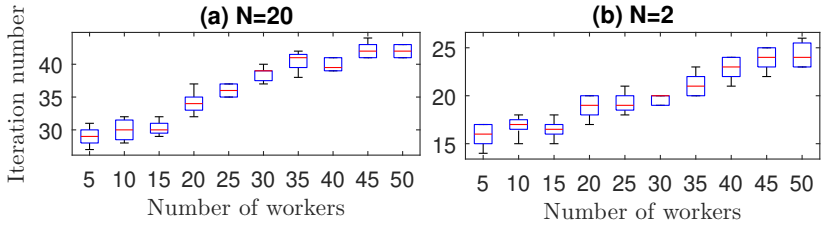


Figure 2.2: Iteration number needed for our algorithm to converge under different MCS sizes.

0.6 and $x_{12} = x_{12}^* = 1.5$ after 20 iterations. It means that the platform is capable of finding the optimal task allocation policy even with the existence of malicious worker u_2. Besides, if setting $\varepsilon = 10^{-1}$ (the gap between two consecutive iterations is less than 10^{-1}), our scheme will stop after only 13 iterations. Hence, if we want to achieve a shorter running time of the scheme, a larger ε is in need.

Impact of MCS Size. We further evaluate in Fig. 3.10 the impact of MCS size, in terms of worker and task numbers, to the scheme performance. For example, in Fig. 3.10(a), when $N = 20$ and $M = 5$, the average iteration number is about 29. It is slightly increased to 42 when $M = 50$. The similar trend is observed in Fig. 3.10(b); when $N = 2$ and M ranges from 5 to 50. We conclude that the algorithm converges pretty fast with moderate system sizes. Moreover, as discussed in the previous section, the platform can further choose a larger ε to accelerate the convergence speed.

Impact of Step Size s. Recall that s stands for the step size of our iterative Algorithm 4. Fig. 2.3 shows the convergence property of Algorithm 4 under different values of s. Still, we consider a system with $M = 50$ workers and $N = 20$ tasks. Note that the dashed line represents the optimal result of P_1 when all workers honestly report their genuine cost functions. We find that the convergence speed is dependent on the step size. For example,

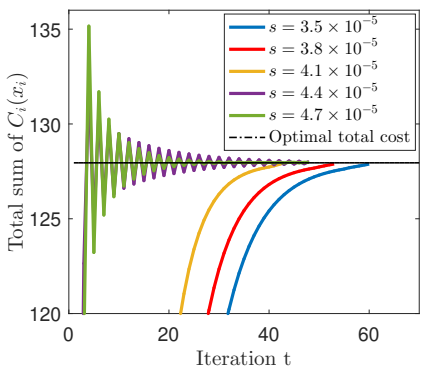


Figure 2.3: Comparison of convergence speed under different step size s.

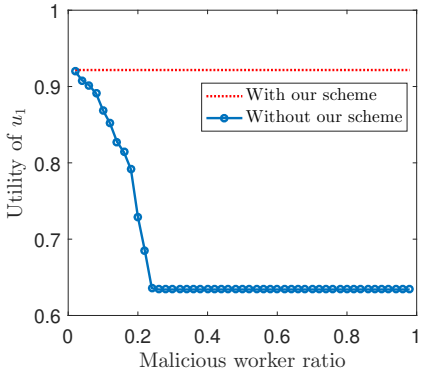


Figure 2.4: u_1's utility with or without our scheme under different malicious worker ratios.

when $s = 3.5 \times 10^{-5}$, it takes 61 iterations to reach the optimal result. When choosing a larger $s = 4.1 \times 10^{-5}$, i.e., a larger step size, the iteration number decreases to 43. However, a larger step size does not necessarily lead to a faster convergence speed. For example, when $s = 4.7 \times 10^{-5}$, the iteration number becomes 48. Thus, a suitable s is critical to the convergence speed of our scheme.

Impact of Malicious Worker Ratio. We now examine the impact of malicious worker ratio to the effectiveness of our scheme. It is defined as the percentage of malicious workers to the entire worker set. We consider an MCS consisting of $M = 50$ workers and $N = 2$ sensing tasks, where u_1 is set as a benign worker. Fig. 2.4 shows u_1's utility U_1. We first examine the case without our scheme. When all workers honestly report their cost functions, i.e., the ratio is 0, we obtain $U_1 = 0.92$. As the ratio increases, U_1 drops fast. In particular, when the ratio is equal to 26%, i.e., there are 13 malicious workers, U_1 becomes 0.64. Moreover, the red dashed line (with our scheme) in this figure clearly demonstrates that U_1 keeps at 0.92, under different malicious worker ratios. Thus, we conclude that the malicious worker ratio does not impact the effectiveness of our scheme.

Performance Enhancement by Integrating Algorithm 5. We now validate the effectiveness of Algorithm 5. Fig. 2.5(a) compares the iteration number needed for Algorithm 4 to converge with and without the integration of Algorithm 5. Apparently, the former is significantly smaller than the latter in all cases when the number of workers is from 5 to 50. For example, when there are 30 workers, the enhanced Algorithm 4 requires 11 iterations to converge, while the other one requires 36 iterations; the former is about 1/3 of the latter. Under the same parameter setting, we further show in Fig. 2.5(b) the running time ratio between these two algorithms. We observe that the enhanced Algorithm 4 only needs about half running time than the other one. We have a similar observation in Fig.

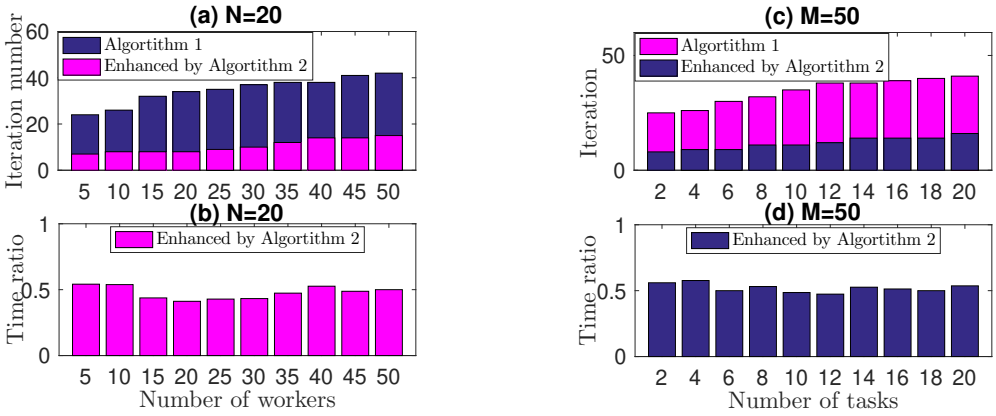


Figure 2.5: Iteration numbers and time needed.

2.5(c) and Fig. 2.5(d) under $M = 50$. Thus, we conclude that Algorithm 5 can significantly improve the performances of Algorithm 4, in terms of both iteration number and running time.

2.6 Related Work

Security issues in MCS. Among the existing works the one that is closest to ours is [27]. It aims to thwart malicious behaviors in worker recruitment of crowdsourcing via the reverse game theory. However, it assumes that the worker's attack statistics are available at the platform. Besides, it formulates worker recruitment into a simple optimization problem, where no constraint is involved. For the other existing works that address security issues in MCS, such as [4, 5, 9], they mainly target at the data analysis stage. As sensing data are reported by workers, they can possibly be altered by malicious ones. This raises the issue of data trustworthiness. Effective schemes are proposed to either detect untrustworthy workers or avoid their reports during data analysis. The research focuses on how to evaluate trustworthiness of the shared data and how to maintain the reputation of various workers. Realizing that the collected data may include sensitive information regarding their reporters, such as locations, daily commute, behavior patterns and habits, the works [6, 7, 8] further guarantee data privacy and/or worker anonymity in their reputation systems design. As we aim to tackle the security issue in the stage of task allocation, the corresponding approach will be quite different.

Incentive mechanism design for MCS. To motivate mobile users to participate in MCS, the incentive mechanism is an effective approach [12, 13, 14, 15, 16]. Workers get paid by the platform to compensate their cost in task sensing. To model the interaction between the platform and workers as well as among workers themselves, auction theory and game theory are widely adopted. Different objectives are discussed, such as maximizing

social welfare[15], maximizing the platform's profit[12], minimizing social cost[16], minimizing privacy loss[13], or minimizing the platform's payment[14]. Although these works also resort to incentive mechanisms, they are resistant to individual misreporting workers' costs.

2.7 Appendix

We first cite a notation that has been used in [22]. It will be extensively used in our analysis. Given $g(x)$ an arbitrary function, and x and y arbitrary real values, $(g(x))_y^+$ is defined as

$$(g(x))_y^+ = \begin{cases} g(x), & y > 0, \\ \max(g(x), 0), & y = 0. \end{cases} \tag{2.20}$$

To prove Theorem 1, it is critical to derive the following lemma first.

Lemma 1. *Given λ, μ and ν[2] as dual solutions obtained in an arbitrary iteration of Algorithm 4, and λ^*, μ^* and ν^* the optimal dual solutions, we have*

$$\begin{aligned}
&(\lambda_i - \lambda_i^*)(\sum_{j\in[1,N]} x_{ij} - t_i)_{\lambda_i}^+ \le (\lambda_i - \lambda_i^*)(\sum_{j\in[1,N]} x_{ij} - t_i), \\
&(\mu_j - \mu_j^*)(T_j - \sum_{i\in[1,M]} \theta_{ij} x_{ij})_{\mu_j}^+ \le (\mu_j - \mu_j^*)(T_j - \sum_{i\in[1,M]} \theta_{ij} x_{ij}), \\
&(\nu_i - \nu_i^*)(\sum_{j\in[1,N]} \frac{d_{ij} x_{ij}}{\theta_{ij}} - D_i)_{\nu_i}^+ \le (\nu_i - \nu_i^*)(\sum_{j\in[1,N]} \frac{d_{ij} x_{ij}}{\theta_{ij}} - D_i).
\end{aligned}$$

Proof. We focus on the first inequality. According to the update rule to λ in Algorithm 4, we have $\lambda \succeq 0$. Then for $\lambda_i \in \lambda$, we discuss under two cases: $\lambda_i = 0$ and $\lambda_i > 0$.

Case I: $\lambda_i = 0$. We have

$$(\lambda_i - \lambda_i^*)(\sum_{j\in[1,N]} x_{ij} - t_i)_{\lambda_i}^+ \le (\lambda_i - \lambda_i^*)(\sum_{j\in[1,N]} x_{ij} - t_i) \tag{2.21}$$

$$\Longleftrightarrow \quad (\sum_{j\in[1,N]} x_{ij} - t_i)_{\lambda_i}^+ \ge \sum_{j\in[1,N]} x_{ij} - t_i \quad \Longleftrightarrow \quad \max\{\sum_{j\in[1,N]} x_{ij} - t_i, 0\} \ge \sum_{j\in[1,N]} x_{ij} - t_i.$$

Therefore, the first inequality holds.

[2] In the following, we use λ, μ and ν to represent $\lambda^{(t)}$, $\mu^{(t)}$ and $\nu^{(t)}$, respectively, without causing confusion.

Case II: $\lambda_i > 0$. We have

$$(\lambda_i - \lambda_i^*)(\sum_{j\in[1,N]} x_{ij} - t_i)^+_{\lambda_i} \leq (\lambda_i - \lambda_i^*)(\sum_{j\in[1,N]} x_{ij} - t_i)$$
$$\iff \quad (\lambda_i - \lambda_i^*)(\sum_{j\in[1,N]} x_{ij} - t_i) \leq (\lambda_i - \lambda_i^*)(\sum_{j\in[1,N]} x_{ij} - t_i).$$

Thus, the first inequality also holds.

In all, the first inequality holds in both cases. The proof for the rest two inequalities similarly follows. □

With Lemma 1, we are now ready to prove Theorem 1. We first rewrite it here again.

Theorem 1. *Algorithm 4 converges to the optimal solution of P_2 globally.*

Proof. We consider a very small time slot, and hence assume that dual solutions are updated according to the differential equations

$$\frac{d\lambda_i}{dt} = (\sum_{j\in[1,N]} x_{ij} - t_i)^+_{\lambda_i}, \tag{2.22}$$

$$\frac{d\mu_j}{dt} = (T_j - \sum_{i\in[1,M]} \theta_{ij} x_{ij})^+_{\mu_j}, \tag{2.23}$$

$$\frac{d\nu_i}{dt} = (\sum_{j\in[1,N]} \frac{d_{ij} x_{ij}}{\theta_{ij}} - D_i)^+_{\nu_i}. \tag{2.24}$$

We first define the Lyapunov function

$$Z(\lambda,\mu,\nu) = \sum_{i=1}^{M} \frac{(\lambda_i - \lambda_i^*)^2}{2} + \sum_{j=1}^{N} \frac{\left(\mu_j - \mu_j^*\right)^2}{2} + \sum_{i=1}^{M} \frac{(\nu_i - \nu_i^*)^2}{2}.$$

If we can prove that $\frac{dZ(\lambda,\mu,\nu)}{dt} \leq 0$, it indicates that $Z(\lambda,\mu,\nu)$ is stable and thus our algorithm converges to the optimal solution of P_2. By applying the chain rule and taking the derivative with respect to t, we obtain

$$\frac{dZ(\lambda,\mu,\nu)}{dt} = \sum_{i\in[1,M]} (\lambda_i - \lambda_i^*) \frac{d\lambda_i}{dt} + \sum_{j\in[1,N]} \left(\mu_j - \mu_j^*\right) \frac{d\mu_j}{dt} + \sum_{i\in[1,M]} (\nu_i - \nu_i^*) \frac{d\nu_i}{dt},$$

which can be rewritten below with (2.22)-(2.24)

$$\begin{aligned}\frac{dZ(\lambda,\mu,\nu)}{dt} = &\sum_{i\in[1,M]} (\lambda_i - \lambda_i^*) \cdot (\sum_{j\in[1,N]} x_{ij} - t_i)^+_{\lambda_i} + \sum_{j\in[1,N]} (\mu_j - \mu_j^*) \cdot (T_j - \sum_{i\in[1,M]} \theta_{ij} x_{ij})^+_{\mu_j} \\ &+ \sum_{i\in[1,M]} (\nu_i - \nu_i^*) \cdot (\sum_{j\in[1,N]} \frac{d_{ij} x_{ij}}{\theta_{ij}} - D_i)^+_{\nu_i}.\end{aligned}$$

From Lemma 1, we directly derive

$$\frac{dZ(\lambda,\mu,\nu)}{dt} \leq \sum_{i\in[1,M]}(\lambda_i-\lambda_i^*)\cdot(\sum_{j\in[1,N]}x_{ij}-t_i)+\sum_{j\in[1,N]}(\mu_j-\mu_j^*)\cdot(T_j-\sum_{i\in[1,M]}\theta_{ij}x_{ij})$$
$$+\sum_{i\in[1,M]}(\nu_i-\nu_i^*)\cdot(\sum_{j\in[1,N]}\frac{d_{ij}x_{ij}}{\theta_{ij}}-D_i).$$

Next, we make some modification in the right-hand-side of the above inequality. Hence, we get

$$\dot{Z} \leq \sum_{i\in[1,M]}(\lambda_i-\lambda_i^*)\cdot(\sum_{j\in[1,N]}x_{ij}-\sum_{j=1}^{N}x_{ij}^*)+\sum_{i\in[1,M]}(\lambda_i-\lambda_i^*)\cdot(\sum_{j\in[1,N]}x_{ij}^*-t_i)$$
$$+\sum_{j\in[1,N]}(\mu_j-\mu_j^*)\cdot(\sum_{i\in[1,M]}\theta_{ij}x_{ij}^*-\sum_{i\in[1,M]}\theta_{ij}x_{ij})+\sum_{j\in[1,N]}(\mu_j-\mu_j^*)\cdot(T_j-\sum_{i\in[1,M]}\theta_{ij}x_{ij}^*)$$
$$+\sum_{i\in[1,M]}(\nu_i-\nu_i^*)\cdot(\sum_{j\in[1,N]}\frac{d_{ij}x_{ij}}{\theta_{ij}}-\sum_{j\in[1,N]}\frac{d_{ij}x_{ij}^*}{\theta_{ij}})+\sum_{i\in[1,M]}(\nu_i-\nu_i^*)\cdot(\sum_{j\in[1,N]}\frac{d_{ij}x_{ij}^*}{\theta_{ij}}-D_i).$$

The second, forth and sixth terms in RHS of the above inequality are nonpositive due to (2.1)-(2.3) and (2.11)-(2.13). Thus,

$$\dot{Z} \leq \sum_{i\in[1,M]}(\lambda_i-\lambda_i^*)\cdot(\sum_{j\in[1,N]}x_{ij}-\sum_{j\in[1,N]}x_{ij}^*)+\sum_{j\in[1,N]}(\mu_j-\mu_j^*)\cdot(\sum_{i\in[1,M]}\theta_{ij}x_{ij}^*-\sum_{i\in[1,M]}\theta_{ij}x_{ij})$$
$$+\sum_{i\in[1,M]}(\nu_i-\nu_i^*)\cdot(\sum_{j\in[1,N]}\frac{d_{ij}x_{ij}}{\theta_{ij}}-\sum_{j\in[1,N]}\frac{d_{ij}x_{ij}^*}{\theta_{ij}})=\sum_{i\in[1,M]}\sum_{j\in[1,N]}(x_{ij}-x_{ij}^*)(\frac{\partial C_i(x_i^*)}{\partial x_{ij}}-\frac{\partial C_i(x_i)}{\partial x_{ij}}).$$

The last equation comes from (2.5). The RHS of the above inequality is nonpositive because of the following property that holds for any convex function $f(\cdot)$ [20]

$$f(u) \geq f(v)+\nabla f(v)^T(u-v). \tag{2.25}$$

To be specific, for any x_i and the optimal solution x_i^*, we have

$$\left[\frac{\partial C_i(x_i^*)}{\partial x_{i1}},\cdots,\frac{\partial C_i(x_i^*)}{\partial x_{iN}}\right]^T\cdot(x_i-x_i^*)\leq C_i(x_i)-C_i(x_i^*)$$
$$\left[\frac{\partial C_i(x_i)}{\partial x_{i1}},\cdots,\frac{\partial C_i(x_i)}{\partial x_{iN}}\right]^T\cdot(x_i^*-x_i)\leq C_i(x_i^*)-C_i(x_i).$$

Then, add the above two inequalities,

$$\left[\frac{\partial C_i(x_i^*)}{\partial x_{i1}} - \frac{\partial C_i(x_i)}{\partial x_{i1}}, \cdots, \frac{\partial C_i(x_i^*)}{\partial x_{iN}} - \frac{\partial C_i(x_i)}{\partial x_{iN}}\right]^T (x_i - x_i^*)$$
$$= \sum_{j\in[1,N]} \left(x_{ij} - x_{ij}^*\right) \left(\frac{\partial C_i(x_i^*)}{\partial x_{ij}} - \frac{\partial C_i(x_i)}{\partial x_{ij}}\right) \leq 0.$$

By adding together the above inequalities for all x_i's ($i \in [1, M]$), we have

$$\sum_{i\in[1,M]} \sum_{j\in[1,N]} \left(x_{ij} - x_{ij}^*\right) \left(\frac{\partial C_i(x_i^*)}{\partial x_{ij}} - \frac{\partial C_i(x_i)}{\partial x_{ij}}\right) \leq 0.$$

Thus, $\dot{Z} \leq 0$ holds, which ends the proof. □

CHAPTER 3

COLLUSION-RESISTANT WORKER RECRUITMENT IN CROWDSOURCING SYSTEMS

3.1 Introduction

Crowdsourcing marketplace emerges as a new paradigm that makes it easier for individuals and businesses to outsource their processes and jobs to a large group of human workers who can perform these tasks virtually. This could include anything from conducting simple data validation and research to more subjective tasks like survey participation, content moderation, and more. Crowdsourcing enables companies to harness the collective intelligence, skills, and insights from a global workforce to streamline business processes, augment data collection and analysis, and accelerate machine learning development. Due to these promising features, recent years have witnessed the prosperity of several commercialized crowdsourcing platforms, such as Amazon Mechanical Turk [129], Uber [89] and Guru [83], etc.

Crowdsourcing has a wide spectrum of potential applications. For example, quite a few research propose to harness the sensing power of distributed mobile devices for spectrum monitoring/sensing of a large geographic area [32, 33, 34, 35, 36, 37, 38]. Under the framework of crowdsourcing, mobile devices are hired to sense the spectrum occupancy/vacancy of their present locations. The aggregated sensing results can produce a real-time fine-grained spectrum usage map over a large geographic area. Crowdsourcing has also gained great interest in the field of wireless signal fingerprinting based indoor/outdoor localization [39, 40, 41, 42, 45, 46]. To reduce the effort of a manual calibration for the site survey, especially in a multi-floor building or a large geographic area, various kinds of crowdsourcing-based indoor localization methodologies have been successfully applied. In addition, many mobile crowdsourcing tasks also exist in commercial crowdsourcing platforms. For example, in Waze [84] some tasks hire workers with mobile devices to carry out geolocation-aware image collection, image tagging, road traffic monitoring, etc. In Taskrabbit [85], the platform publishes spatial tasks such as cleaning a house or walking a dog. Typically, these tasks are only accessible by workers nearby.

Participating in crowdsourcing is usually costly for individual workers, since they spend time and wisdom in task execution. Therefore, effective incentive mechanisms are essential to stimulate worker participation. Great efforts have been devoted to this research area. *Reverse auctions* [130, 131, 132] that rely one workers' costs, have been extensively

adopted. Specifically, workers compete with each other by submitting to the platform their costs, i.e., bids (i.e., the minimum payment they accept for the task), then winners and payments are decided accordingly. As proved in these works, such competition can effectively bring down the platform's expense in hiring cheaper labor and thus significantly enhance economic efficiency.

Despite the appealing properties, auction-based markets are deemed vulnerable to bidder misbehaviors [73]. Strategic bidders, individually or in groups, may seek to game the system by coordinating their bids to manipulate auction outcomes. To make the best use of crowdsourcing systems, a worker recruitment auction must discourage workers from cheating and instead encourage them to reveal their true cost regarding task execution to the platform. In this context, the existing works [133, 134, 58, 59, 61, 62, 70, 71, 72, 142] have been focusing on *truthfulness*; no worker, individually, can improve its utility by bidding other than its actual cost.

Truthful auctions in crowdsourcing, however, become ineffective when workers collude, i.e., they strategically form coalitions and rig their bids together for illegitimate beneficial gain. Albeit being legally banned, collusions have widely appeared in past commercial auctions and have had significant effects, e.g., FCC spectrum auctions [74, 75, 76, 77], treasure auctions [78, 79], eBay online auctions [87, 88, 90], and auctions in P2P systems [91, 93, 94]. Empirical analysis on these auctions reveals that most collusion groups are small, less than 6 members per group [77, 90, 94]. In the domain of crowdsourcing, which typically involves a large number of workers, such small-size collusion groups are thus easy to form among friends and close relatives. Moreover, since crowdsourcing auctions are conducted online among anonymous workers, collusions can be hard to detect. Thus, it renders collusion an even more challenging issue to tackle in the crowdsourcing marketplace.

Collusion resistance has rarely been studied in the context of crowdsourcing, except [95] that targets a particular form of collusion. Instead, we aim to resist a broad set of collusion attacks. In fact, designing collusion-resistant mechanisms is a nontrivial task. According to the *impossibility results* proved in [96, 112], there is no "strong" collusion-resistant mechanism, which can optimize or even approximate any nontrivial objective function without any assumption over auction settings. Only a handful of collusion resistance works exist for general auctions so far [96, 97, 98, 99]. However, most of them rely on assumptions such as incomplete information sharing among colluders [97, 98] and the auctioneer having prior knowledge over bidder behaviors [99]. Neither of the above assumptions holds in practical crowdsourcing systems. To avoid these constraints, a scheme called APM is proposed by Goldberg and Hartline [96]. Notice that the above works are designed for generic auctions, while the worker recruitment in crowdsourcing is featured

with unique characteristics. For example, crowdsourcing tasks are typically imposed with quality requirements from their requesters. Besides, workers are profiled with their reputations and "types", such as age, region, education level, etc. All these factors reform the winner selection process by introducing various constraints to the worker recruitment formulation. As a result, existing mechanisms are not readily applicable. Thus, an effective collusion resistance scheme that is suitable for crowdsourcing is in dire need.

To resist collusions, we take a proactive prevention approach, because uncovering collusion coalitions is hard due to its tacit nature and complex auction structure. Specifically, we design the rules for winner selection and payment determination to diminish the utility gain of coalitions, leaving workers little or no incentive to collude. We resort to a "soft" approach that suppresses collusions in a probabilistic manner. A (t, p)-collusion resistance scheme is developed. Particularly, with a probability of p, no coalition of weighted cardinality t or less can improve its group utility by coordinating the bids. Besides, the proposed scheme also achieves p-truthfulness and p-individual rationality. Additionally, we provide a formal analysis of the platform's extra cost caused by trading for collusion avoidance.

The main contribution of this work is summarized as follows

- We address the critical collusion issue in auction-based crowdsourcing systems. This issue has rarely been discussed so far.
- We develop a (t, p)-collusion resistance scheme. It successfully defends against strategic behaviors from coalitions with weighted cardinality t at a probability p.
- We conduct comprehensive theoretical analysis over the critical economic and collusion resistant properties achieved by our scheme.
- A real-world dataset extracted from the commercial crowdsourcing platform Guru [83] is used to evaluate the performances of the proposed scheme.

3.2 System Model and Problem Statement

In this section, we first introduce the system model. The framework of auction-based worker recruitment in crowdsourcing is introduced. Then we examine the formation and the impact of worker collusion therein. It shows that the property of worker recruitment auctions provides a fertile breeding ground for collusions, causing a significant revenue loss at the platform.

Table 3.1: Notations.

w_i	the i-th worker	T_j	the j-th type
b_i	bid of w_i	x_i	decision indicator
η_i	b_i/k_i	k_i	reputation of w_i
b	$\{b_i : i \in [1,N]\}$	$\mathscr{G}$ or $\mathscr{G}_j$	coalition
$b_{\mathscr{G}}/c_{\mathscr{G}}$	bid/cost set of $\mathscr{G}$	$c_{\mathscr{W}\setminus\mathscr{G}}$	cost set of $\mathscr{W}\setminus\mathscr{G}$
$P_i(b)$	payment to w_i	$\mathscr{E}$	$\{E_j : j \in [1,M]\}$
a	interval of $\mathscr{A}$	$u_{\mathscr{G}}$	utility of coalition $\mathscr{G}$
c_i	cost of w_i	$\mathscr{T}$	$\{T_j : j \in [1,M]\}$
$\mathscr{W}$	$\{w_i : i \in [1,N]\}$	$h_u^{\theta}(\cdot)$	rounded up function
$u_i(b)$	utility of w_i	η_j	$\{\eta_i : \forall i\ w_i \in E_j\}$

t_j	weighted cardinality of $\mathscr{G}_i$ in T_j
$\underline{k}$	reputation threshold of w_i
l_j	reputation threshold of T_j
E_j	sorted worker set of T_j according to η_i
$\mathscr{A}$	discrete value set $\{a,\cdots,r\cdot a,\cdots,R\cdot a\}$
$\Gamma_{r_j\cdot a}(\eta_j)$	subset of η_j; $\forall\eta_i \in \Gamma_{r_j\cdot a}(\eta_j)$, $\eta_i \le r_j\cdot a$
$\mathscr{A}_j$	subset of $\mathscr{A}$; $\forall r_j\cdot a \in \mathscr{A}_j$, $\sum_{i:\eta_i\in\Gamma_{r_j\cdot a}(\eta_j)} k_i \ge l_j$
$\mathscr{H}$	function set of the form $h_u^{\theta}(\cdot)$ with $u \in [0,1]$

3.2.1 System Model

The crowdsourcing system considered in this work consists of a platform and a large set of workers $\mathscr{W} = \{w_1,\cdots,w_i,\cdots,w_N\}$. The platform hosts a set of to-do tasks, while the workers are intelligent laborers that are willing to carry out tasks in trade of monetary rewards. Following many prior works in this field [58, 59, 61, 62, 70, 71, 72, 142], the platform adopts the framework of *reverse auction* to recruit workers. The platform and workers play the role of the auctioneer and bidders, respectively. In a generic reverse auction, bidders compete with each other by offering the bid, i.e., the minimum payment one accepts for conducting a task. Upon the collection of all bids, the auctioneer picks the winner(s) who offer the lowest bids and determines the corresponding payment that should be paid to a winning bidder. Table 3.1 lists the notions used in this work.

Unlike generic auctions, in crowdsourcing worker's "reputation" is taken into account for worker recruitment. Here we use *weight*, denoted by $k_i \in [0,1]$, to represent worker w_i's reputation. For the rest of the paper, we use the terms "reputation" and "weight" interchangeably. This value can be maintained and updated by the platform from a long-term observation. Typically, highly-rated workers overweight the bad-mouthed ones. For example, in Amazon Mechanical Turk [129] and Guru [83], task requesters are allowed to set the preference to workers with high ratings. A task typically requires workers of

different backgrounds to work on. For instance, for a task that collects public opinions on the best picture out of a given set, it is desirable to recruit workers covering comprehensive demography, as people of different ages, regions, education levels, etc. may have distinct perceptions. To reflect this property, following [142, 150], each worker is exclusively classified into one of the following *types* $\mathscr{T} = \{T_1, \cdots, T_j, \cdots, T_M\}$. We overload the notation $w_i \in T_j$, meaning that w_i is a type T_j worker. Such information is provided by workers during registration[1]. Besides, we assume that the accomplishment of a task needs diverse workers covering the type set $\mathscr{T}$[2].

Denote by c_i worker w_i's associated cost toward the task, indicating the minimum payment it accepts. c_i is private and known only to w_i itself. To compete for task execution opportunities, w_i submits bid b_i. Upon receiving bids $b = \{b_1, \cdots, b_i, \cdots, b_N\}$ from all workers, the platform formulates a worker recruitment problem that aims to minimize the accumulated payment from to each winning worker, denoted as $P_i(b)$[3], while taking into account the task result quality and crucial economic properties in auctions.

$$
\begin{aligned}
\min \quad & \sum_{i:w_i \in \mathscr{W}} P_i(b) x_i \\
s.t. \quad & \sum_{i:w_i \in T_j} k_i x_i \geq l_j, \quad \forall j \in [1, M], && (3.1) \\
& k_i \geq \underline{k}, \quad \forall i \in \{i : x_i = 1\}, && (3.2) \\
& \bigcup_{j:w_i \in T_j, \forall x_i = 1} T_j = \mathscr{T}, && (3.3) \\
& x_i \in \{0, 1\}, \quad \forall i \in [1, N], \\
& \text{IC, IR and Collusion resistance.}
\end{aligned}
$$

x_i is a binary decision variable. $x_i = 1$ means that w_i is recruited; $x_i = 0$ otherwise. The above problem aims to minimize the platform's overall payment.

To guarantee the task result quality, multiple workers should be hired for each type. Constraint (3.1) says that the *weighted cardinality* of the hired worker set for each type cannot be lower than l_j, a threshold determined by the platform to provide quality-guaranteed services. The operation of summing up reputations from workers has been adopted in real-world crowdsourcing applications. One example is the web application iSpot which

[1] We pertain the discussion over bid collusion in this work, collusion of misreporting type information [69] and tasks' answers [63, 102] have been studied and the joint collusion over bids, worker types or worker answers will be considered in our future work.

[2] Our design also fits for the case where a task only needs workers from a subset of types $\mathscr{T}' \subseteq \mathscr{T}$ with minor modification to the scheme.

[3] $P_i(b)$ is expressed as the function of the entire bid set b because the former is dependent on the latter.

exploits crowdsourcing to identify species accurately in biodiversity science [116, 117]. iSpot calculates the total reputational weight attached to one label of a given unknown species as the sum of the reputation of workers who reports this label. If this total exceeds a pre-set threshold, iSpot marks this label as the ID of the species. In constraint (3.2), $\underline{k}$ denotes the minimal reputation a worker must have for being recruited. By tuning $\underline{k}$, the platform effectively filters out the workers that are disqualified with low reputation. Constraint (3.3) requires that all worker types should be covered. Take crowdsourcing-based spectrum monitoring/sensing as an illustration. Each mobile device is only able to obtain the spectrum usage at a specific location given its limited sensing range. To derive a complete spectrum usage map over a large geographic area, it is desirable for the service provider to recruit a set of workers covering all locations. Treating locations as types, constraint (3.3) guarantees the hiring of workers of all types, i.e., mobile devices at all locations. Moreover, any solution to the above problem should also satisfy some inherent economic properties, such as *truthfulness* (also called *incentive compatibility* (IC)) and *individual rationality* (IR). Finally, the platform calculates payment $P_i(b)$ to each winning worker w_i. A loser does not execute any task and receives zero payment.

To facilitate the scheme design, we formally present a worker's utility and a coalition group's utility in Definition 1 and Definition 2, respectively. Worker w_i's utility is denoted as $u_i(b)$. It is expressed as a function of bid set b as the former is dependent of the latter.

Definition 1. *(A Worker's Utility.) Given the bid set b, the utility of a worker $w_i \in \mathscr{W}$ is*

$$u_i(b) = (P_i(b) - c_i)x_i.$$

Definition 2. *(A Coalition's Group Utility.) Given the bid set b, the group utility of a coalition $\mathscr{G}$ is*

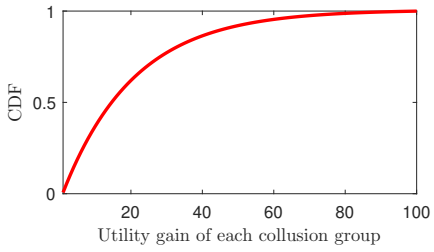
$$u_{\mathscr{G}} = \sum_{i:w_i \in \mathscr{G}} u_i(b),$$

i.e., the sum of individual utility from all workers in the same coalition $\mathscr{G}$.

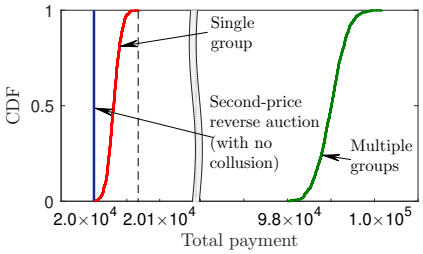
3.2.2 Problem Statement

Workers are modeled as rational and self-interested; they may game the system for higher beneficial gain. Thus, collusions occur in an auction when a group of bidders form a coalition, rig their bids to manipulate auction outcomes, and gain higher group utility. In below, we first use a simple example to illustrate how it impacts the crowdsourcing marketplace.

Assume that there are four workers ($N = 4$), all of whom are of the same type T and weight $k = 1$, and the task only needs type T workers to execute. Their associated



(a) Gain in group utility.



(b) Payment caused to the platform.

Figure 3.1: Small-size collusions have severe impact on auction-based crowdsourcing marketplace.

cost are set to $c_1 = 15$, $c_2 = 17$, $c_3 = 20$ and $c_4 = 60$. Let $l = 2$. To achieve minimum payment, truthfulness, and individual rationality simultaneously, we adopt the widely used *second-price reverse auction* [130] here for worker recruitment. Specifically, winners are the ones who bid with the lowest prices, and their payments are given by the lowest losing bid. When all workers bid with their true costs ($b_i = c_i, \forall i$), the winners are w_1 and w_2, each paid at 20. Now assume that w_2 and w_3 collude, i.e., $\mathscr{G} = \{w_2, w_3\}$ and w_3 raises its bid to 59. In this case where $b_i = c_i, i \in \{1,2,4\}$ and $b_3 = 59$, w_1 and w_2 win while w_3 still loses. Under the second-price reverse auction, winner's payment is the lowest losing bid, i.e., b_3. Hence, w_2's payment becomes 59, which is significantly larger than 20 received without collusion. Since w_2 and w_3 form a coalition, w_2 can transfer some part of its extra income to w_3. As a result, each of them achieves a higher utility.

We then further examine collusion impacts via larger-scale simulations. Assume that there are 5000 workers ($N = 5000$) that are categorized into 10 types ($M = 10$). Let $l_j = 100$ ($j \in [1,10]$). As the example above, we consider small-size coalitions of size 2. There are 1000 such coalitions in the system. Each of them adopts the similar collusion strategy introduced above, i.e., one worker honestly submits its cost, while the other increases the bid. In the simulation, each worker's cost and weight are randomly selected from $[1,100]$ and $[0,1]$, respectively. In Figure 3.1(a) we plot the group utility improvement of each coalition in 100 trials. It shows that workers have incentives to collude, since collusions are easy to perform and are highly beneficial. The red curve and green curve in Figure 3.1(b) represent the distribution of the total payment caused to the platform when only one collusion group and 1000 collusion groups exist, respectively. Apparently, the impact on payment is limited when only one collusion group is present, but the payment increases by five times when 1000 collusion groups are present. The above illustrations show that small-size collusions are particularly effective in raising worker group utility and imposing extra costs on the platform.

So far, we have only considered one kind of collusions, where some workers from a coalition offer bids higher than their true costs, while the others report genuine values. In fact, colluders are feasible to adopt a broad spectrum of strategies, e.g., they can arbitrarily raise or lower bids, as long as it brings a higher group utility. In this study, we aim to defend against such general collusions.

Let $b_{\mathscr{G}}$ and $c_{\mathscr{G}}$ be the bid set and true cost set of all members from coalition $\mathscr{G}$, respectively. $c_{\mathscr{W}\setminus\mathscr{G}}$ stands for the true cost set from all the other workers except the ones from coalition $\mathscr{G}$. Then $u_i(b_{\mathscr{G}}, c_{\mathscr{W}\setminus\mathscr{G}})$ is the utility of worker w_i, a member of $\mathscr{G}$, when it and its fellows from $\mathscr{G}$ collaborate to arbitrarily rig their bids. Similarly, $u_i(c_{\mathscr{G}}, c_{\mathscr{W}\setminus\mathscr{G}})$ is the utility of worker w_i, when it and its fellows from $\mathscr{G}$ honestly report their costs. Then, possible collusion strategies are defined as follows.

Definition 3. *(Collusion strategies.) Workers from a coalition $\mathscr{G}$ arbitrarily raise/decrease their bids to increase coalition's group utility $u_{\mathscr{G}}$, i.e.,*

$$\sum_{i:w_i\in\mathscr{G}} u_i(b_{\mathscr{G}}, c_{\mathscr{W}\setminus\mathscr{G}}) > \sum_{i:w_i\in\mathscr{G}} u_i(c_{\mathscr{G}}, c_{\mathscr{W}\setminus\mathscr{G}}),$$

where $b_{\mathscr{G}} \diamond c_{\mathscr{G}}$, denoting some elements of $b_{\mathscr{G}}$ and $c_{\mathscr{G}}$ satisfy $b_i > c_i$, some of them satisfy $b_i < c_i$, and the rest are the same.

The symbol $b_{\mathscr{G}} \diamond c_{\mathscr{G}}$ denotes that some of colluders offer bids higher than their true cost $b_j > c_j$, some of them offer bids lower than their true cost $b_j < c_j$, and the rest keep their bids unchanged $b_j = c_j$. A collusion is success, if the accumulated utility from all members of coalition with collusion is higher than that without collusion, i.e., $\sum_{i:w_i\in\mathscr{G}} u_i(b_{\mathscr{G}}, c_{\mathscr{W}\setminus\mathscr{G}}) > \sum_{i:w_i\in\mathscr{G}} u_i(c_{\mathscr{G}}, c_{\mathscr{W}\setminus\mathscr{G}})$.

"Utility" is terminology in economics. It stands for the benefit of consuming or providing a good/service. The utility of a worker in our paper is the net income of selling its service. It is equal to the payment received from the platform minus its cost, as shown in Definition 1. Similarly, group utility is the net income of a coalition for selling their offered services, which is equal to the sum of the utility of all workers in the coalition.

Note that multiple coalitions may coexist in an auction. Besides, one worker w_i can participate in different coalitions, say $\mathscr{G}_1$, $\mathscr{G}_2$, $\cdots$, the collusion is deemed success if any of the following inequality holds, $\sum_{i:w_i\in\mathscr{G}_1} u_i(b_{\mathscr{G}_1}, c_{\mathscr{W}\setminus\mathscr{G}_1}) > \sum_{i:w_i\in\mathscr{G}_1} u_i(c_{\mathscr{G}_1}, c_{\mathscr{W}\setminus\mathscr{G}_1})$, $\sum_{i:w_i\in\mathscr{G}_2} u_i(b_{\mathscr{G}_2}, c_{\mathscr{W}\setminus\mathscr{G}_2}) > \sum_{i:w_i\in\mathscr{G}_2} u_i(c_{\mathscr{G}_2}, c_{\mathscr{W}\setminus\mathscr{G}_2})$, $\cdots$. To prevent worker w_i from collusion, our goal is to ensure none of these inequalities exist under careful mechanism design. While workers may collude by misreporting other information in addition to bids, we focus the discussion on bid collusion in this work, due to the design complexity.

3.3 A Basic Scheme without Collusion Resistance

In this section, we first develop a basic worker recruitment auction scheme without collusion resistance. The discussion of this basic scheme is critical, as it serves as the cornerstone for our comprehensive collusion-resistant worker recruitment that will be presented in the next section.

3.3.1 Basic Scheme Design

Upon receiving each worker w_i's bid b_i, the platform checks the worker's type via accessing its profile. It first rules out workers whose reputations are lower than the threshold $\underline{k}$. The platform then lists all workers of type T_j ($j \in [1,M]$) and sorts them in an ascending order according to the *per unit weight bid* $\eta_i = b_i/k_i$. Denote by E_j this sorted worker set. Let $\mathscr{E} = \{E_j : j \in [1,M]\}$ and $\eta_j = \{\eta_i : \forall i\ w_i \in E_j\}$. In order to recruit workers covering all types, winners are selected in each E_j.

Here we elaborate the design rationality of the above-mentioned steps. Recall that the worker recruitment optimization problem aims to minimize the platform's overall payment with the constraints from task quality and task type coverage. As this problem is NP-hard, we resort to the above heuristic steps to approximately solve it. The intuition of ranking η_i from low to high is that the worker with a lower η_i is preferred by the platform compared with the one with a higher η_i, since a lower η_i indicates a lower bid (and thus potentially lower payment) but a higher weight/reputation. Then the platform selects winning workers starting from the lowest η_i.

The platform maintains a set of discrete values $\mathscr{A} = \{a, \cdots, r \cdot a, \cdots, R \cdot a\}$, such that

$$a \leq \min_i \{\eta_i\}, \quad R \cdot a \geq \max_i \{\eta_i\}. \tag{3.4}$$

We present the following definition which is critical for our scheme design.

Definition 4. *Let y be an ascending vector. Denote by $\Gamma_x(y)$ the set of elements in y with the value at most x.*

For each $E_j \in \mathscr{E}$, the platform first generates a set $\mathscr{A}_j = \{r_j \cdot a : \sum_{i:\eta_i \in \Gamma_{r_j \cdot a}(\eta_j)} k_i \geq l_j, r_j \in [1,2,\cdots,R]\}$ from $\mathscr{A}$. It then identifies the value $r_j^* \cdot a$ from $\mathscr{A}_j$ such that $(r_j^* \cdot a) \cdot \sum_{i:\eta_i \in \Gamma_{r_j^* \cdot a}(\eta_j)} k_i$ is minimized. The winners in E_j, i.e., of type T_j, are the workers whose per unit weight bid η_i is at most $r_j^* \cdot a$, each receiving the payment at $k_i \cdot (r_j^* \cdot a)$. The rest workers lose. The platform's total payment for hiring type T_j workers is calculated as $(r_j^* \cdot a) \cdot \sum_{i:\eta_i \in \Gamma_{r_j^* \cdot a}(\eta_j)} k_i$. Note that $r_j^* \cdot a$ can be viewed as winner's *per unit weight payment*. It has the following property.

Proposition 2. *For $r_j^* \cdot a$ of any $E_j \in \mathscr{E}$ that satisfies*

$$r_j^* \cdot a = \arg \min_{r_j \cdot a \in \mathscr{A}_j} \big((r_j \cdot a) \cdot \sum_{i:\eta_i \in \Gamma_{r_j \cdot a}(\eta_j)} k_i\big),$$

we have

$$\sum_{i:\eta_i \in \Gamma_{r_j^* \cdot a}(\eta_j)} k_i \geq l_j, \tag{3.5}$$

$$\sum_{i:\eta_i \in \Gamma_{(r_j^*-1)\cdot a}(\eta_j)} k_i < l_j. \tag{3.6}$$

Proof. First of all, we directly have (3.5) according to how $r_j^* \cdot a$ is derived in the basic scheme. We then prove (3.6) via the contradiction method. Assume that $\sum_{i:\eta_i \in \Gamma_{(r_j^*-1)\cdot a}(\eta_j)} k_i \geq l_j$. We have

$$(r_j^* \cdot a) \cdot \sum_{i:\eta_i \in \Gamma_{r_j^* \cdot a}(\eta_j)} k_i \leq [(r_j^* - 1) \cdot a] \cdot \sum_{i:\eta_i \in \Gamma_{(r_j^*-1)\cdot a}(\eta_j)} k_i, \tag{3.7}$$

as otherwise $(r_j^* - 1) \cdot a$ will become winner's per unit weight payment. From Definition 4, we know that

$$\sum_{i:\eta_i \in \Gamma_{(r_j^*-1)\cdot a}(\eta_j)} k_i < \sum_{i:\eta_i \in \Gamma_{r_j^* \cdot a}(\eta_j)} k_i,$$

and thus

$$[(r_j^* - 1) \cdot a] \cdot \sum_{i:\eta_i \in \Gamma_{(r_j^*-1)\cdot a}(\eta_j)} k_i < (r_j^* \cdot a) \cdot \sum_{i:\eta_i \in \Gamma_{r_j^* \cdot a}(\eta_j)} k_i.$$

which contradicts with (3.7). It implies that the assumption $\sum_{i:\eta_i \in \Gamma_{(r_j^*-1)\cdot a}(\eta_j)} k_i \geq l_j$ is invalid. Thus, (3.6) holds. □

Proposition 2 provides an efficient way to determine winning workers and their payments. Specifically, once $\mathscr{A}_j$ is generated for each worker type T_j, instead of comparing $(r_j \cdot a) \cdot \sum_{i:\eta_i \in \Gamma_{r_j \cdot a}(\eta_j)} k_i$'s for each element in $\mathscr{A}_j$ and identifying the minimum one, the first element of $\mathscr{A}_j$ is exactly the per unit weight payment to each winner. Besides, any worker with its per unit weight bid no larger than this value is the winner. It is not difficult to tell that the computation complexity of the basic scheme is $\mathscr{O}(MN)$.

The design rationale of the basic scheme can be interpreted in the following way. Note that once per unit weight payment is determined, so is the worker's payment, as its reputation times per unit weight payment. Hence, a worker's payment is independent of

its bid, providing strategic players limited incentives to rig their bids. While collusion is still viable by manipulating the final payment of the basic scheme, the effect is largely constrained. This is because the per unit weight payment (under collusion) is always in the set $\{r_j \cdot a : r_j \in [r_j^L, r_j^H], r_j \in \mathbf{Z}^+\}$, and thus a worker's payment is limited to the set $\{k_i \cdot r_j \cdot a : r_j \in [r_j^L, r_j^H], r_j \in \mathbf{Z}^+\}$. Without introducing $\mathscr{A}$ or confining per unit weight payment to $\mathscr{A}$, the payment under collusion will be unbounded.

The choice of a needs to balance the platform's payment and the scheme robustness. Generally, a larger a causes less payment at the platform, but at a cost of weaker scheme robustness; specifically, the probability that the scheme is robust against collusion is lower. In the simulation, we provide extensive discussions regarding the choice of a.

3.3.2 A Walk-through Example

Since this example only aims to show how our scheme operates, the collusion pattern (i.e., the number of coalition group, who rise/reduce their bid and how much they coordinate their bids, etc) and the platform parameter (such as l_j) are arbitrarily set. Consider that there are thirteen workers ($N = 13$) and one task that looks for two types of workers ($M = 2$). Besides, let $l_j = 2$ ($j \in \{1,2\}$) and $\underline{k} = 0$. Assume that $w_1 - w_5, w_{11}, w_{13} \in T_1$ and $w_6 - w_{10}, w_{12} \in T_2$. The weight of $w_1 - w_3$, $w_5 - w_6$ is 0.5 and that of w_4, $w_7 - w_{13}$ is 1. Worker's bids are listed as

$$b_1 = 15, b_2 = 17, b_3 = 19, b_4 = 44, b_5 = 24, b_6 = 31, b_7 = 33,$$
$$b_8 = 36, b_9 = 38, b_{10} = 39, b_{11} = 40, b_{12} = 50, b_{13} = 60.$$

We derive two sorted worker sets

$$E_1 = \{w_1, w_2, w_3, w_{11}, w_4, w_5, w_{13}\}, E_2 = \{w_7, w_8, w_9, w_{10}, w_{12}, w_6\},$$

with the corresponding η_j as

$$\eta_1 = \{30, 34, 38, 40, 44, 48, 60\}, \eta_2 = \{33, 36, 38, 39, 50, 62\},$$

Let $a = 4$, then $\mathscr{A}_1 = \{10 \cdot 4, 11 \cdot 4, \cdots, 15 \cdot 4\}$ and $\mathscr{A}_2 = \{9 \cdot 4, 10 \cdot 4, \cdots, 16 \cdot 4\}$ since $\sum_{i:\eta_i \in \Gamma_{10 \cdot 4}(\eta_1)} k_i = 2.5 > l_1$ and $\sum_{i:\eta_i \in \Gamma_{9 \cdot 4}(\eta_2)} k_i = 2 = l_2$. According to Proposition 2, $r_1^* \cdot a = 10 \cdot 4 = 40$ and $r_2^* \cdot a = 9 \cdot 4 = 36$. Thus, winners in E_1 (of type T_1) are w_1, w_2, w_3 and w_{11}, with the first three paid at 20 each and the last one paid at 40. w_7 and w_8 are recruited in E_2 (of type T_2) and paid at 36 each. The rest workers lose and get 0. Thus, the platform's total payment is 172.

3.4 Final Collusion-Resistant Scheme

In this section, we develop a collusion-resistant worker recruitment auction based on the basic scheme.

3.4.1 Collusion Patterns

Consider a worker set $E_j \in \mathscr{E}$. Without collusion, the per unit weight payment to each winner is $r_j^* \cdot a$ according to our basic scheme. Assume that there is a collusion group $\mathscr{G}_j$ in E_j with *weighted cardinality* $t_j = \sum_{i:w_i \in \mathscr{G}_j} k_i$. When k_i's are all 1's, t_j is directly the cardinality of $\mathscr{G}_j$, i.e., the number of members in this coalition. As described in Definition 3, colluders may choose to raise or lower bids for group utility gain. We start from a special case, where they raise their bids so as to manipulate the payment, like the example shown in Section 3.2.2. More precisely, each winner's per unit weight payment can be increased *up to* $r_j^H \cdot a$, satisfying

$$\sum_{i:\eta_i \in \Gamma_{r_j^H \cdot a}(\eta_j)} k_i - t_j = \sum_{i:\eta_i \in \Gamma_{r_j^* \cdot a}(\eta_j)} k_i. \tag{3.8}$$

In another special case, where colluders decrease their bids, then each winner's per unit weight payment can be decreased *down to* $r_j^L \cdot a$, satisfying

$$\sum_{i:\eta_i \in \Gamma_{r_j^L \cdot a}(\eta_j)} k_i + t_j = \sum_{i:\eta_i \in \Gamma_{r_j^* \cdot a}(\eta_j)} k_i. \tag{3.9}$$

Although a winner's per unit weight payment has been decreased, it is possible that more colluders become winners. As a result, their group utility can still potentially be increased. It is not difficult to derive that $r_j^L \cdot a \leq r_j^* \cdot a \leq r_j^H \cdot a$.

The above discussion reveals the mechanism how a coalition receives group utility gain through manipulating the winner's payment. Ideally, if the following assumption exists

$$r_j^* \cdot a = r_j \cdot a, \quad \forall r_j \in [r_j^L, r_j^H] \tag{3.10}$$

i.e., winners are paid undifferentiated no matter a coalition colludes or not, the motivation of collusions will be diminished. Nonetheless, such a design idea is infeasible unless certain modifications are made, which will be the focus next.

3.4.2 Scheme Design

We develop a "soft" collusion-resistance approach: no coalition gain is achievable from colluding with a probability p. We formally define a (t, p)-collusion resistant auction.

Definition 5. *((t,p)-collusion resistant auction.) An auction is (t,p)-collusion resistant, if, with a probability of p or higher, no coalition with weighted cardinality t can improve its group utility by coordinating the bids. This holds even if multiple collusion groups are present, as long as each group's weighted cardinality is t or less.*

Meanwhile, we also aim to achieve truthfulness and individual rationality under the soft collusion-resistant auction framework. Recall that b_i and c_i are worker w_i's bid and true cost respectively, while c_{-i} stands for the cost set from all workers except w_i.

Definition 6. *(p-Truthfulness.) The worker recruitment auction is p-truthful, if*

$$\Pr[u_i(b_i, c_{-i}) \leq u_i(c_i, c_{-i})] \geq p, \quad \forall w_i \in \mathscr{W}$$

Definition 7. *(p-Individual Rationality.) The worker recruitment auction is p-individual rational, if*

$$\Pr[u_i \geq 0] \geq p, \quad \forall w_i \in \mathscr{W}$$

In order to defend against collusions, our idea is to carefully set winner payment, such that it will not be influenced by colluders' strategies. Before we delve into design details, we first define $[\alpha,\beta]$-*consensus estimate*.

Definition 8. *($[\alpha,\beta]$-consensus estimate.) Given $\alpha,\beta > 0$ and $v > 0$, we say that a function $h(\cdot)$ is a $[\alpha,\beta]$-consensus estimate of v if*

1. *for any w such that $\alpha \leq w \leq \beta$, we have $h(w) = h(v)$;*
2. *$h(v)$ is a nontrivial upper bound on v, i.e., $0 < v \leq h(v)$.*

$h(v)$ is called the consensus value.

Consider a function

$$h_u^{\theta}(v) = v \text{ rounded up to nearest } \theta^{s+u} \tag{3.11}$$

where s is a tunable integer and θ is a carefully chosen positive real value. The selection of θ depends on α and β. The definition of $h_u^{\theta}(\cdot)$ implies that for any v, $v \leq h_u^{\theta}(v) \leq \theta \cdot v$. Define $\mathscr{H}$ as the set of functions of the form $h_u^{\theta}(\cdot)$ with u chosen uniformly on $[0,1]$.

Definition 8 and the design of function $h_u^{\theta}(\cdot)$ are inherited from [101], but modified to accommodate our scenario. Specifically, $h_u^{\theta}(\cdot)$ is a rounded-up function here, i.e., $h_u^{\theta}(v) \geq v$ given value v, while that in [101] is a rounded-down function. One reason for such a change is to ensure non-negative worker utility in the context of crowdsourcing where the reverse auction framework is adopted. More importantly, *consensus estimate* in [101] is to achieve a high *competitive ratio*, while we leverage it to develop a soft collusion resistance approach. Therefore, the purpose and parameter design rationale of $h_u^{\theta}(\cdot)$ in these two works are different. We can induce the following corollary from [101].

Corollary 1. *For h from $\mathscr{H}$ and a given value $v > 0$, $h(v)$ is distributed identically to $\theta^U v$ where U is a random variable following uniform distribution on $[0,1]$.*

Proof. Consider a random variable $Y = \log_k h(v)$ and let $t = \log_k v$. Then $\Pr[Y \leq t + x] = \Pr[U \leq x]$ and therefore Y is uniformly distributed between t and $t+1$. Thus, $h(v)$ is identical to $k^U v$. □

Proposition 3. *For h from $\mathscr{H}$ and a given value $v > 0$, the probability that $h(v)$ is a $[\alpha,\beta]$-consensus estimate of v is $1 - \log_\theta \frac{\beta}{\alpha}$.*

Proof. According to Definition 8, h is a $[\alpha,\beta]$-consensus estimate of v if $h(\alpha) = h(v) = h(\beta) \geq \beta$. From Corollary 1, we have

$$\begin{aligned}\Pr[h(\alpha) \geq \beta] &= \Pr\left[\theta^U \alpha \geq \beta\right] = \Pr\left[\theta^U \geq \frac{\beta}{\alpha}\right] = 1 - \Pr\left[\theta^U \leq \frac{\beta}{\alpha}\right] \\ &= 1 - \Pr\left[U \leq \log_\theta \frac{\beta}{\alpha}\right] = 1 - \log_\theta \frac{\beta}{\alpha}.\end{aligned}$$

□

We are now ready to introduce our (t,p)-collusion resistant worker recruitment auction. Its pseudo-code is presented in Algorithm 3. Upon receiving bids from workers, the platform derives $\mathscr{E}$. For each $E_j \in \mathscr{E}$, the platform generates $\mathscr{A}_j$ and identifies the value $r_j^* \cdot a$ from $\mathscr{A}_j$ such that $(r_j^* \cdot a) \cdot \sum_{i:\eta_i \in \Gamma_{r_j^* \cdot a}(\eta_j)} k_i$ is minimized. According to Proposition 2, $r_j^* \cdot a$ is simply the first element of $\mathscr{A}_j$. The platform then selects a suitable function $h_u^{\theta_j}(\cdot)$. The winners in E_j are the workers of per unit weight bids at most $h_u^{\theta_j}(r_j^* \cdot a)$. A winner w_i is then paid at $k_i \cdot h_u^{\theta_j}(r_j^* \cdot a)$. The rest workers lose.

Algorithm 3 (t,p)-collusion resistant worker recruitment

Input: b_i, k_i, t_j and l_j ($i \in [1,N], j \in [1,M]$)
Output: x_i and $P_i(b)$ ($i \in [1,N]$)

1: **for** each $T_j \in \mathscr{T}$ **do**
2: Platform generates worker set E_j and $\mathscr{A}_j$;
3: Identify the first value in $\mathscr{A}_j$ and set it as $r_j^* \cdot a$;
4: Select $h_u^{\theta_j}(\cdot)$ based on t_j and η_j;
5: Winners of T_j are the ones with $\eta_i \leq h_u^{\theta_j}(r_j^* \cdot a)$;
6: Calculate each winner's payment as $P_i(b) = k_i \cdot h_u^{\theta_j}(r_j^* \cdot a)$.
7: **end for**

As long as $h_u^{\theta_j}(r_j \cdot a)$ is a $(r_j^L \cdot a, r_j^H \cdot a)$-consensus estimate of $r_j \cdot a \in [r_j^L \cdot a, r_j^H \cdot a]$, we have

$$h_u^{\theta_j}(r_j \cdot a) = h_u^{\theta_j}(r_j^* \cdot a) \quad r_j \in [r_j^L, r_j^H] \tag{3.12}$$

with the probability p_j. This is because $r_j^L \cdot a \leq r^* \cdot a \leq r_j^H \cdot a$. According to Proposition 3, p_j is calculated as

$$p_j = 1 - log_{\theta_j} \frac{r_j^H \cdot a}{r_j^L \cdot a} = 1 - log_{\theta_j} \frac{r_j^H}{r_j^L} \tag{3.13}$$

by setting $\alpha = r_j^L \cdot a$ and $\beta = r_j^H \cdot a$. It means no collusions will impact winner's per unit weight payment and thus its total payment with a probability p_j. If p_j is high, it fails the motivation of collusion at a large chance. We can set an arbitrary value of p_j from $(0,1)$ by tuning θ_j and a.

Now the remaining issue is to generate a suitable $h_u^{\theta_j}(\cdot)$ to have (3.12) hold. For this purpose, we first identify $r_j^H \cdot a$ and $r_j^L \cdot a$ via (3.8) and (3.9), respectively. Then θ_j is carefully selected such that (3.12) holds for $v \in [r_j^L \cdot a, r_j^H \cdot a]$. Specifically, we should expect

$$\begin{aligned} &r_j^L \cdot a \leq h_u^{\theta_j}(r_j^L \cdot a) \leq r_j^L \cdot a \cdot \theta_j, \quad r_j^* \cdot a \leq h_u^{\theta_j}(r_j^* \cdot a) \leq r_j^* \cdot a \cdot \theta_j, \\ &r_j^H \cdot a \leq h_u^{\theta_j}(r_j^H \cdot a) \leq r_j^H \cdot a \cdot \theta_j. \end{aligned}$$

Together with (3.12) and the fact that $r_j^L \leq r_j^* \leq r_j^H$, in order to have the above equations hold, we must have

$$r_j^H \cdot a \leq h_u^{\theta_j}(r_j \cdot a) \leq r_j^L \cdot a \cdot \theta_j, \quad \forall r_j \in [r_j^L, r_j^H]$$

which requires $r_j^L \cdot a \cdot \theta_j \geq r_j^H \cdot a$ and thus $\theta_j \geq r_j^H / r_j^L$.

Up to now, we have presented how to determine winners and their payments for E_j. The same procedure will be followed to handle the rest sets in $\mathscr{E}$. The scheme is (t,p)-collusion resistance, where $t = \min_{j \in [1,M]}\{t_j\}$ and $p = \prod_{j=1}^{M} p_j$. Its formal analysis will be given in Theorem 3.

Theorem 2. *The computation complexity of our (t,p)-collusion resistant worker recruitment algorithm is upper bounded by $\mathscr{O}(MN)$.*

Proof. The computation complexity of Algorithm 3 is dominated by the while-loop, which contains M iterations, where M is the number of worker types. For each iteration, it involves a computation of generating the worker set E_j and the corresponding $\mathscr{A}_j$ (line 2), causing N times of look-up operations and up to R times of comparisons, respectively. Besides, for the process of selecting $h_u^{\theta_j}(\cdot)$ (line 4), its main component is to identify $r_j^H \cdot a$ and $r_j^L \cdot a$, which

results in $2R$ times of comparison at most. For the process of determining winning workers (line 5), it involves N times of comparison at most. Therefore, the computation complexity of Algorithm 3 is upper bounded by $\mathscr{O}(M(2N+3R))$. Recall that R is a constant value in the algorithm. The computation complexity is thus rewritten as $\mathscr{O}(MN)$. □

3.4.3 A Walk-through Example

To better explain our scheme, we still take the example in Section 3.3.2 as an illustration. Following the same procedure as in the basic scheme, the platform first generates the worker sets $E_1 = \{w_1, w_2, w_3, w_{11}, w_4\ , w_5, w_{13}\}$ and $E_2 = \{w_7, w_8, w_9, w_{10},\ w_{12}, w_6\}$, the corresponding $\mathscr{A}_1 = \{10\cdot 4, \cdots, 15\cdot 4\}$ and $\mathscr{A}_2 = \{9\cdot 4, 10\cdot 4, \cdots, 16\cdot 4\}$ with $a = 4$, and $r_1^* \cdot a = 10\cdot 4 = 40$ and $r_2^* \cdot a = 9\cdot 4 = 36$.

Assume that the platform intends to defend against a coalition with weighted cardinality up to $t = 1.5$ in each worker type. According to (3.8) and (3.9), for E_1 we have $r_1^L \cdot a = 8\cdot 4 = 32$, $r_1^H \cdot a = 11\cdot 4 = 44$. Following the requirement $\theta_j \geq r_j^H / r_j^L$, a feasible value of θ_j is 3 is selected. In order to decide winners and their payments for E_1, let $u = 0.4$ be an instantiation. Recall that u is a random value chosen from $[0,1]$. Then $h_{0.4}^3(r_1^* \cdot a) = h_{0.4}^3(40)$ is calculated as "40 rounded up to the nearest $3^{s+0.4}$" (with s as a tunable integer), which gives us 41.9. According to the scheme, workers with per unit weight bids no larger than 41.9 are winners for E_1. Thus, $w_1 - w_3$ and w_{11} are winners paid at 21.0, 21.0, 21.0 and 41.9 respectively. $w_2 - w_5$ and w_{13} lose.

Following the similar idea, for E_2, $w_7 - w_8$ are winners, with each paid at 37.5 ($\theta_2 = 3, u = 0.3$), while others lose. The platform's total payment is thus 179.9, which is only slightly above, around 5%, than that caused in the basic scheme with no collusion resistance.

3.4.4 Addressing Inter-type Collusions

So far we have focused on the scenarios where colluders from the same coalition reside in the same worker set $E_j \in \mathscr{E}$, i.e., they belong to the same type $T_j \in \mathscr{T}$. Such kind of collusion can be viewed as *intra-type collusions*. Nonetheless, it is also possible that colluders from the same coalition are of different types, which we call *inter-type collusions*. For instance, in the example discussed in Section 3.3.2 and 3.4.3, the collusion among w_2, w_3 and w_6 is exactly an inter-type collusion.

Even though inter-type collusions seem to be more complex than intra-type collusions, each inter-type collusion can be equivalently divided into multiple independent intra-type collusions. This is because collusions under each worker type are dealt one by one in our scheme, including winner selection and payment determination. Hence, the winners'

payment for one worker type is irrelevant to that for the other type. As a result, a colluder's utility received in E_j does not impact its peers' utilities received in another $E_{j'}$ ($j \neq j'$). Therefore, for the inter-task collusion coalition $\mathscr{G} = \{w_2, w_3, w_6\}$, it can be divided into two intra-collusion coalitions, i.e., $\mathscr{G}_1 = \{w_2, w_3\}$ and $\mathscr{G}_2 = \{w_6\}$. To sum up, if we can effectively discourage the formation of intra-collusion coalitions, so for the inter-collusion coalitions.

3.4.5 Restrictions on Our Scheme

Generally, it is difficult to design a scheme that can defend against an arbitrary number of colluders. This is the same case for our scheme.

In order to have our scheme work, the condition (3.8) and (3.9) should meet, or equivalently,

$$\sum_{i:\eta_i \in \Gamma_{r_j^L \cdot a}(\eta_j)} k_i = \sum_{i:\eta_i \in \Gamma_{r_j^* \cdot a}(\eta_j)} k_i - t_j. \tag{3.14}$$

$$\sum_{i:\eta_i \in \Gamma_{r_j^H \cdot a}(\eta_j)} k_i = \sum_{i:\eta_i \in \Gamma_{r_j^* \cdot a}(\eta_j)} k_i + t_j. \tag{3.15}$$

In order to make sure $\sum_{i:\eta_i \in \Gamma_{r_j^L \cdot a}(\eta_j)} k_i$ and $\sum_{i:\eta_i \in \Gamma_{r_j^H \cdot a}(\eta_j)} k_i$ exist in any worker set E_j, then

$$\sum_{i:\eta_i \in \Gamma_{r_j^L \cdot a}(\eta_j)} k_i > 0, \tag{3.16}$$

$$\sum_{i:\eta_i \in \Gamma_{r_j^H \cdot a}(\eta_j)} k_i \leq \sum_{i:w_i \in E_j} k_i. \tag{3.17}$$

Substituting (3.14) and (3.15) into (3.16) and (3.17) respectively, we derive following restrictions on t_j:

$$t_j < \sum_{i:\eta_i \in \Gamma_{r_j^* \cdot a}(\eta_j)} k_i, \tag{3.18}$$

$$t_j \leq \sum_{i:w_i \in E_j} k_i - \sum_{i:\eta_i \in \Gamma_{r_j^* \cdot a}(\eta_j)} k_i. \tag{3.19}$$

When coalitions are of small size, then the relation $t_j < l_j$ typically exists, i.e., the weighted cardinality of a coalition is smaller than l_j. Recall that l_j is a threshold selected by the crowdsourcing platform to ensure service quality during worker recruitment. Besides, as $l_j \leq \sum_{i:\eta_i \in \Gamma_{r_j^* \cdot a}(\eta_j)} k_i$, then (3.18) holds. Moreover, there are a large population of workers in a real crowdsourcing system. Hence, we have $\sum_{i:w_i \in E_j} k_i \gg t_j$ and thus (3.19) also satisfies.

To sum up, our scheme can effectively defend against small-scale collusions in a crowdsourcing system where there are a large set of workers.

3.5 Property Analysis

In this section, we provide a formal analysis of various properties achieved by our scheme.

Recall that $\mathscr{E}$ is denoted as $\mathscr{E} = \{E_j : j \in [1,M]\}$. t_j is the weighted cardinality of the coalition which resides in E_j. p_j is the coalition's collusion success probability

Lemma 2. *For any $E_j \in \mathscr{E}$, our scheme achieves (t_j, p_j)-collusion resistance, with p_j defined by (3.13).*

Proof. It is equivalent to show that any coalition of weighted cardinality t_j cannot obtain higher group utility by rigging their bids, with a probability p_j or higher. In the following, we plan to first show the validity of the above statement for two special cases of collusions, where colluders either raise or decrease bids. Then the statement for an arbitrary collusion strategy given in Definition 3 will directly follow.

Denote by $\mathscr{T}_j' \subseteq \mathscr{G}_j$ ($\mathscr{T}_j \subseteq \mathscr{G}_j$) and u_j' (u_j) the set of winning colluders by raising bids in E_j and their corresponding utility when they collude (or not), respectively. As colluders raise their bids, some of them may lose, thus $\mathscr{T}_j' \subseteq \mathscr{T}_j$. The difference between the coalition in E_j's group utility when they collude or not is calculated as

$$\begin{aligned} u_j - u_j' &= \sum_{i:w_i \in \mathscr{T}_j} \left[k_i \cdot h_u^{\theta_j}(r_j^* \cdot a) - c_i \right] - \sum_{i:w_i \in \mathscr{T}_j'} \left[k_i \cdot h_u^{\theta_j}(r_j' \cdot a) - c_i \right] \\ &= \sum_{i:w_i \in \mathscr{T}_j \setminus \mathscr{T}_j'} \left[k_i \cdot h_u^{\theta_j}(r_j^* \cdot a) - c_i \right] \geq 0 \end{aligned}$$

with a probability p_j, where $h_u^{\theta_j}(r_j' \cdot a)$ stands for the per unit weight payment when collusions take place. Specifically, as $r_j^* \cdot a \leq r_j' \cdot a \leq r_j^H \cdot a$ and thus $h_u^{\theta_j}(r_j^* \cdot a) = h_u^{\theta_j}(r_j' \cdot a)$ holds with a probability p_j according to (3.12). Hence, the second equation above holds with a probability p_j. Besides, for a colluder $w_i \in \mathscr{T}_j \setminus \mathscr{T}_j'$, as it wins without collusion, we have $\eta_i = c_i/k_i \leq h_u^{\theta_j}(r_j^* \cdot a)$ according to Algorithm 3, which directly leads to the last inequality. Besides, p_j associates with t_j. Thus, the above expression indicates that any coalition of weighted cardinality t_j cannot achieve a higher group utility by raising bids with a probability p_j.

We further denote by $\mathscr{T}_j'' \subseteq \mathscr{G}_j$ ($\mathscr{T}_j \subseteq \mathscr{G}_j$) and u_j'' (u_j) the set of winning colluders by decreasing bids in E_j and their corresponding utility when they collude (or not), respectively. Some workers who lose when bid truthfully may win when they collude, thus

$\mathscr{T}_j \subseteq \mathscr{T}_j''$. The difference between the coalition in E_j's group utility when they collude or not is calculated as

$$u_j - u_j'' = \sum_{i:w_i \in \mathscr{T}_j} \left[k_i \cdot h_u^{\theta_j}(r_j^* \cdot a) - c_i \right] - \sum_{i:w_i \in \mathscr{T}_j''} \left[k_i \cdot h_u^{\theta_j}(r_j' \cdot a) - c_i \right]$$
$$= - \sum_{i:w_i \in \mathscr{T}_j'' \setminus \mathscr{T}_j} \left[k_i \cdot h_u^{\theta_j}(r_j^* \cdot a) - c_i \right] \geq 0$$

with a probability p_j. Specifically, as $r_j^L \cdot a \leq r_j' \cdot a \leq r_j^* \cdot a$ and thus $h_u^{\theta_j}(r_j^* \cdot a) = h_u^{\theta_j}(r_j' \cdot a)$ with a probability p_j according to (3.12). Hence, the second equation above holds at p_j. Besides, for a colluder $w_i \in \mathscr{T}_j'' \setminus \mathscr{T}_j$, as it loses without collusion, we have $h_u^{\theta_j}(r_j^* \cdot a) \leq c_i/k_i = \eta_i$ according to Algorithm 3, which thus leads to the last inequality. The above expression indicates that any coalition of weighted cardinality t_j cannot achieve a higher group utility by decreasing bids with a probability p_j.

For a coalition where members adopt arbitrary strategies given by Definition 3, it can be viewed as the combination of the above two special cases. Following a similar approach, the statement can be validated under this scenario. As its proof is similar to the above, we omit its discussion here. □

Based on Lemma 2, we are ready to give the following theorem on collusion resistance. Note that the deduction of θ_j, r_j^H and r_j^L in Theorem 3 is discussed in Section 3.4.1.

Theorem 3. *Our scheme achieves (t, p)-collusion resistance with*

$$p = \prod_{j=1}^{M} \left(1 - \log_{\theta_j} \frac{r_j^H}{r_j^L} \right)$$

and $t = \min_{j \in [1,M]} \{t_j\}$.

Proof. For intra-type collusions, a coalition forms within a single $E_j \in \mathscr{E}$. And collusions in different E_j's are independent with each other. According to Lemma 2, when colluders are of weighted cardinality up to t_j in each E_j, our scheme is collusion resistance at a probability p_j. Since there are totally M E_j's, our scheme can defend against any coalition of weighted cardinality t with probability p, where $t = \min_{j \in [1,M]} \{t_j\}$ and $p = \prod_{j=1}^{M} \left(1 - \log_{\theta_j} r_j^H / r_j^L\right)$.

For inter-type collusions, consider an arbitrary coalition $\mathscr{G}$ of weighted cardinality t' forming across M E_j's and $t' = \sum_{j=1}^{M} t_j$. From the discussion of Section 3.4.4, $\mathscr{G}$ can be equivalently divided into M intra-type coalitions, each with the weighted cardinality

t_j. Besides, we have proved in Lemma 2 that our scheme is collusion resistance to any coalition of weighted cardinality t_j in each E_j with a probability p_j. Thus, our scheme is capable of defending against any coalition of weighted cardinality t' with a probability p for inter-type collusions, where $t' = \sum_{j=1}^{M} t_j$ and $p = \prod_{j=1}^{M} \left(1 - \log_{\theta_j} r_j^H / r_j^L\right)$.

Combining the results for both intra- and inter-type collusions, we conclude that our scheme is (t, p)-collusion resistance. □

Comparing Definition 5 and Definition 6, by setting $t = \max_i\{k_i\}$, a (t, p)-collusion resistance auction is degraded to a p-truthful auction. Hence, the latter can be viewed as a special case for the former.

Corollary 2. *Our scheme is p-truthful with*

$$p = \prod_{j=1}^{M} \left(1 - \log_{\theta_j} \frac{r_j^H}{r_j^L}\right).$$

Theorem 4. *Our scheme is p-individual rational, i.e.,*

$$\Pr[u_i(b) \geq 0] \geq p \quad \forall i \in [1, N].$$

Proof. If w_i is a winner, then $P_i(b) = k_i \cdot h_u^{\theta_j}(r_j^* \cdot a) \geq b_i$. Meanwhile, according to Corollary 2, our scheme is p-truthful. Thus $\Pr[b_i = c_i] \geq p$. As a result, $\Pr[P_i(b) - c_i \geq 0] \geq p$. On the other hand, if w_i loses, then $\Pr[u_i(b) = 0] = 1$. In either case, the above statement holds. □

According to above proof, we have $P_i(b) = k_i \cdot h_u^{\theta_j}(r_j^* \cdot a) \geq b_i$, i.e., a worker's payment is always no less than its bid. For a truthful worker with $b_i = c_i$, IR is always satisfied as $P_i(b) \geq b_i = c_i$. For a colluder, as our scheme is p-truthful (i.e., $\Pr[b_i = c_i] \geq p$), IR is satisfied with probability p, i.e., $\Pr[P_i(b) \geq c_i] \geq p$. Therefore, our scheme always produces non-negative utility for truthful workers. On the other hand, it does cause negative utility at a certain probability p to workers who tend to explore the system for beneficial gain via untruthful bidding. Besides, the soft guarantee of economic properties, such as IC, IR, and collusion resistance, are commonly seen in incentive design. For example, [118] designs the par-per-click auction that is truthful with an error probability. [119] also proposes a probabilistic version of IR. Our design falls into this category.

From the scheme design, we can tell that the collusion resistance property is achieved by recruiting redundant workers and overpaying each winner, i.e., trading the platform's extra cost with collusion resistance. Hence, it is critical to examine the extra cost caused to the platform. We first evaluate the ratio between the platform's cost of our final scheme and that of the basic scheme.

Proposition 4. *The platform pays no larger than $P_b \cdot \sum_{i=1}^{N} k_i \sum_{j=1}^{M} \theta_j / l_j$ in the (t,p)-collusion resistant scheme, where P_b is the platform's payment under the basic scheme.*

Proof. Denote by P_t as the platform's total payment under the (t,p)-collusion resistant scheme.

$$\begin{aligned}\frac{P_t}{P_b} =& \frac{\sum_{j=1}^{M} h_u^{\theta_j}(r_j^* \cdot a) \cdot \sum_{i:\eta_i \in \Gamma_{h_u^{\theta}(r_j^* \cdot a)}(\eta_j)} k_i}{\sum_{j=1}^{M} r_j^* \cdot a \cdot \sum_{i:\eta_i \in \Gamma_{r_j^* \cdot a}(\eta_j)} k_i} \overset{①}{\leq} \frac{\sum_{j=1}^{M} \theta_j r_j^* \cdot a \cdot \sum_{i:\eta_i \in \Gamma_{h_u^{\theta}(r_j^* \cdot a)}(\eta_j)} k_i}{\sum_{j=1}^{M} r_j^* \cdot a \cdot l_j} \\ \leq& \frac{\sum_{j=1}^{M} \theta_j r_j^* \cdot a \cdot \sum_{i:w_i \in E_j} k_i}{\sum_{j=1}^{M} r_j^* \cdot a \cdot l_j} \overset{②}{\leq} \sum_{j=1}^{M} \frac{\theta_j \cdot \sum_{i=1}^{N} k_i}{l_j}\end{aligned}$$

where ① is derived because $h_u^{\theta_j}(r_j^* \cdot a) \leq \theta_j \cdot r_j^* \cdot a$ (due to the property of $h_u^{\theta_j}$) and $\sum_{i:\eta_i \in \Gamma_{r_j^* \cdot a}(\eta_j)} k_i \geq l_j$. ② is due to the fact that $\sum_j \alpha_j / \sum_j \beta_j \leq \sum_j \alpha_j / \beta_j$ when $\alpha_j, \beta_j > 0$. Hence, $P_t \leq P_b \cdot \sum_{i=1}^{N} k_i \sum_{j=1}^{M} \theta_j / l_j$. □

We further analyze the *frugality* of our scheme. It is defined as the ratio between the payment caused by our scheme P_t and the optimum payment P_{opt}, by solving the original worker recruitment optimization problem without considering collusion resistance, truthfulness, or individual rationality. Therefore, frugality evaluates the amount of extra payment our scheme causes in the trade of its critical properties.

Theorem 5. *The frugality of our scheme P_t / P_{opt} satisfies $\frac{P_t}{P_{opt}} \leq \sum_{j=1}^{M} \frac{\theta_j r_j^* \cdot \sum_{i=1}^{N} k_i}{l_j}$.*

Proof. We define the k-th lowest bid from workers in E_j as $b_j^{(k)}$. We have

$$\begin{aligned}\frac{P_t}{P_{opt}} \leq& \frac{\sum_{j=1}^{M} h_u^{\theta_j}(r_j^* \cdot a) \cdot \sum_{i:\eta_i \in \Gamma_{h_u^{\theta}(r_j^* \cdot a)}(\eta_j)} k_i}{\sum_{j=1}^{M} \sum_{k=1}^{l_j} b_j^{(k)}} \overset{③}{\leq} \frac{\sum_{j=1}^{M} \theta_j r_j^* \cdot a \cdot \sum_{i:\eta_i \in \Gamma_{h_u^{\theta}(r_j^* \cdot a)}(\eta_j)} k_i}{\sum_{j=1}^{M} l_j \cdot b_j^{(1)}} \\ \leq& \frac{\sum_{j=1}^{M} \theta_j r_j^* \cdot a \cdot \sum_{i:w_i \in E_j} k_i}{\sum_{j=1}^{M} l_j \cdot b_j^{(1)}} \overset{④}{\leq} \frac{\sum_{j=1}^{M} \theta_j r_j^* \cdot a \cdot \sum_{i:w_i \in E_j} k_i}{\sum_{j=1}^{M} l_j \cdot a} \overset{⑤}{\leq} \sum_{j=1}^{M} \frac{\theta_j r_j^* \cdot \sum_{i=1}^{N} k_i}{l_j}\end{aligned}$$

where ③ and ⑤ are due to same reasons for ① and ② respectively. ④ is derived due to (3.4). □

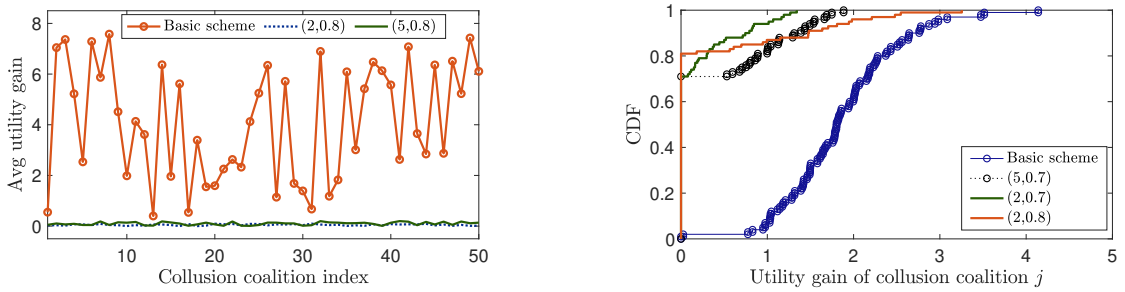
3.6 Performance Evaluation

3.6.1 Dataset

To validate the proposed scheme, we employ a real-world dataset obtained from the commercial crowdsourcing platform Guru[4]. We focus on tasks in the field of Programming & Development, which involves 894 tasks and 26904 workers. For each task, we record its required worker skills, such as experience in iOS App development and graphic design, etc. The required skill set is mapped to $\mathscr{T}$, the type set of our scheme. Thus, the cardinality of $\mathscr{T}$ is used to instantiate $M = 50$ in evaluation. For each worker, we record its offered skill, received rating (from its historical employers in the platform), and asked salary (dollars/hour), which are then mapped to the type T_j this worker belongs to, weight k_i, and exerted cost c_i, respectively, in the simulation. A total of 50 coalition groups are randomly formed among the 26904 workers. By the default setting, one coalition is formed in each type. Besides, their weighted cardinality is upper bounded by 2. Within each group, workers arbitrarily raise/decrease their bids. l_j is set to 50. All simulation results are the average over 100 trials.

3.6.2 Collusion Resistance

To examine the collusion resistance property, we analyze the utility gain, which is defined as the difference between a coalition's group utility achieved with and without the scheme. Intuitively, if a coalition's utility gain is 0, i.e., collusion does not produce a higher group utility, it eliminates the members' motivation for collusion.



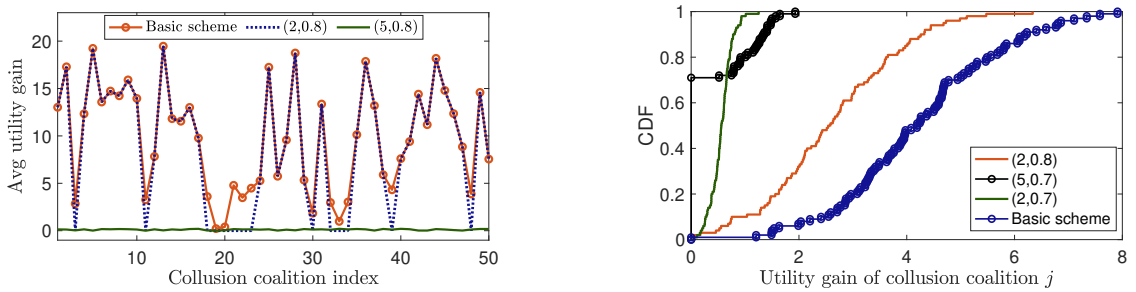
(a) Average utility gain

(b) Cumulative distribution of instance utility gain

Figure 3.2: Collusion resistance performance comparison. One coalition group exists in each type, with each coalition's weighted cardinality $k = 2$.

[4]Except Guru dataset, we are not able to identify other datasets that also contain bid information of real-world crowdsourcing systems. We would also like to point out that Guru dataset [83] has been adopted in prior works on crowdsourcing [120, 121].

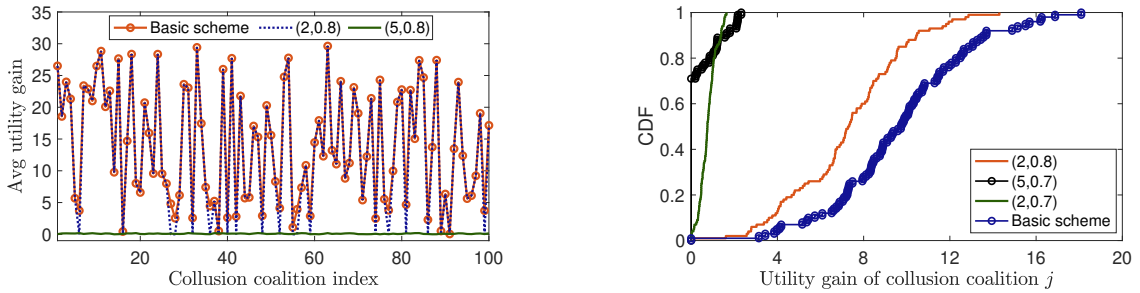
In the simulation, Fig. 3.2 shows the coalition's utility gain when the coalition cardinality is 2. We find in Fig. 3.2(a) that our scheme, under the setting of both $(2, 0.8)$ and $(5, 0.8)$, works well. Utility gain is rarely observed throughout all coalition groups. This is because the two settings can defend coalitions with the cardinality of 2 and 5, respectively. Fig. 3.2(b) depicts the cumulative distribution of instant utility gain of a randomly selected coalition. It is derived under the same setting of Fig. 3.2(a). Instead of average utility gain, Fig. 3.2(b) examines the CDF of utility gain. The coalition's maximal utility gain is as high as 4 under the basic scheme, which is significantly larger than that when our scheme is in place. Besides, the coalition's maximal utility gain under $(2, 0.7)$ is 1.5, which is smaller than that under $(5, 0.7)$, i.e., 2. It indicates that the platform can better restrict a coalition's instant group utility when setting a smaller t. Given a larger t, we are expecting a larger r^H/r^L. Thus, under the same p, i.e., 0.7 here, the larger t produces a larger θ according to (3.13). As a result, the instant payment a winner gets will be larger, which, as a consequence, brings a larger instant utility gain. By comparing with the utility gain achieved under $(2, 0.7)$ and $(2, 0.8)$, we find that the latter can prevent collusion at a higher success rate. However, it leads to a larger instant utility gain; if a coalition succeeds in colluding, it receives a higher gain. This phenomenon can be explained following the similar rationality above.



(a) Average utility gain

(b) Cumulative distribution of instance utility gain

Figure 3.3: Collusion resistance performance comparison. One coalition group exists in each type, with each coalition's weighted cardinality $k = 5$.



(a) Average utility gain

(b) Cumulative distribution of instance utility gain

Figure 3.4: Collusion resistance performance comparison. Two coalition groups exist in each type, with each coalition's weighted cardinality $k = 5$.

In Fig. 3.3, the coalition's weighted cardinality k is set to 5. Fig. 3.3(a) shows that the average utility gain can reach as high as 25 under the basic scheme (without collusion resistance). Under $(t, p) = (5, 0.8)$, the average utility gain keeps close to 0 throughout all coalitions. It means that no coalition can explore positive utility gain *on average*. Therefore, collusion is effectively prevented. We then set $(t, p) = (2, 0.8)$. However, it exhibits poor performance when implemented to resist collusions with weighted cardinality 5. The average utility of some coalitions decreases to 0, while the remaining coalitions' average utility is the same as that of the basic scheme, which denotes $(2, 0.8)$ resist parts of collusions because colluders size is larger than $t = 2$. The above result is slightly different from that of Fig. 3.2(a): our scheme with $(5, 0.8)$ can still effectively resist collusion while the scheme with $(2, 0.8)$ cannot. This is simply because the coalition with cardinality 5 is beyond the capacity of our defense scheme with $(2, 0.8)$. A result similar to Fig. 3.2(b) can be observed in Fig. 3.3(b).

Fig. 3.4 shows the performance when two coalition groups are present in each type. In the setting, as there are 50 types, the total number of coalition groups is 100. For comparison purposes, we set the coalition cardinality as 5. According to the figure, collusion can still be effectively defended with our scheme under $(5, 0.8)$. This is the same case with Fig. 3.3, where only one coalition per type exists. Thus, we conclude that the number of coalitions does not impact the collusion resistance performance of our scheme. It complies with our theoretical result.

Fig. 3.5 shows the collusion resistance property of our scheme by evaluating the total payment occurred at the platform. When no coalition exists, the total payment of the basic scheme is less than that of the proposed scheme. However, the payment increases dramatically as more collusion takes place, while this value keeps almost constant in our scheme under all three settings. This is because the payment to each winner remains unchanged with a high probability as long as the coalition's weighted cardinality is no larger than 2, i.e., a default value in our simulations. On the other hand, as the basic scheme cannot resist

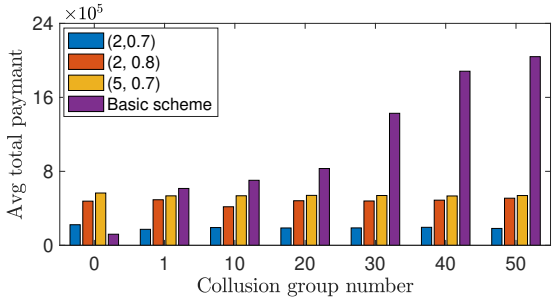


Figure 3.5: Total payment comparison under different collusion group numbers.

Table 3.2: Upper bound of colluder ratio in real-world FCC auctions [77].

PCS-C Block	Auction 35	AWS-1	700 MHz
48.7%	1.1%	1.5%	0.03%

collusion, it causes significantly increased payment as more malicious workers are present. When there is no collusion, the average total payment with the basic scheme is 1.1×10^5, while that with the proposed scheme under three settings (2, 0.7), (2, 0.8) and (5, 0.7) are $2.2 \times 10^5, 5.2 \times 10^5, 6.1 \times 10^5$, resulting in the corresponding ratio as 2, 4.7, 5.5, respectively. Although the proposed scheme incurs extract payment than the basic scheme when no collusion exists, the payment of the basic scheme increases dramatically as more collusion presents. In an extreme case where 50 collusion groups misbehave, the total payment reaches 2.0×10^6 under the basic scheme, while that of the proposed scheme under (2, 0.7) is merely 2.2×10^5 which is about 1/9 of the former.

Table 3.2 shows the upper-bound of collusion ratio in Federal Communications Commission (FCC) spectrum auctions. Four real-world FCC spectrum auctions are examined, PCS-C Block, Auction 35, AWS-1 and, 700 MHz. (Please refer to [77] for details of these four auctions.) Their collusion ratio is upper-bounded by 48.7%, 1.1%, 1.5%, and 0.03%, respectively. Recall that our scheme causes lower total payment than the basic scheme as long as one collusion group exists. Thus, our final scheme is more cost-effective than the basic scheme under all non-zero collusion ratios.

3.6.3 Payment

As mentioned, the collusion resistance property of our scheme is achieved by causing extra payment (overpayment) at the platform. Thus, in this part, we evaluate the overpayment and the effect of parameters, i.e., t and p to the platform's payment to winners.

We first check the impact of the threshold l_j to a winning worker's payment in Fig. 3.6. The payment demonstrates a "step" shape as l_j increases. This is due to the rounding

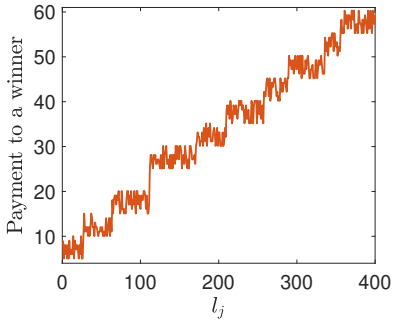
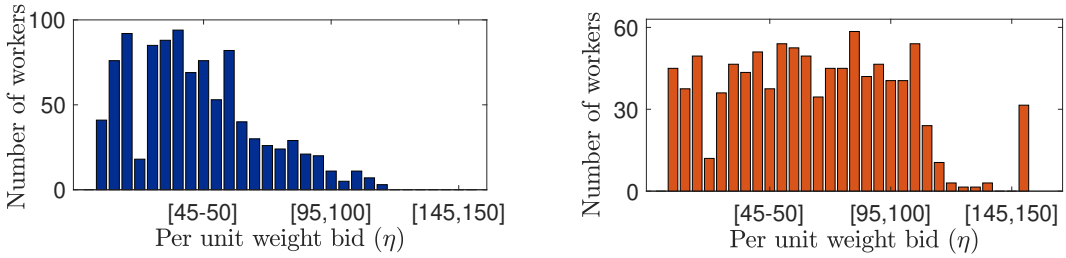


Figure 3.6: Payment to one randomly selected winner.

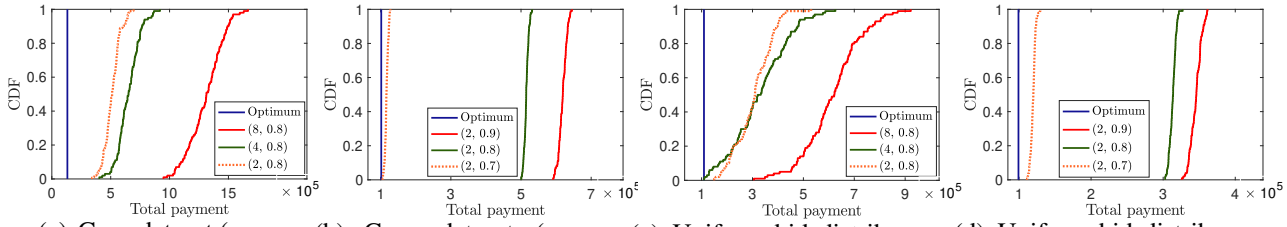


(a) Guru dataset (b) Uniform bid distribution

Figure 3.7: Distribution of *per unit weight bids* of two different datasets.

operation $h_u^{\theta}(\cdot)$ involved in the payment calculation. Besides, we observe that the payment increases as l_j grows. From the scheme design, we can infer that a larger l_j leads to a larger $r_j^* \cdot a$ and thus a larger $h_u^{\theta}(r_j^* \cdot a)$. Note that $h_u^{\theta}(\cdot)$ is a non-decreasing function. As the winner's payment is calculated as $k_i \cdot h_u^{\theta}(r_j^* \cdot a)$, it has positive correlation with l_j.

Fig. 3.8 examines the distribution of total payment of our scheme and P_{opt}. Recall that P_{opt} is obtained via optimally solving the original worker recruitment optimization problem without considering collusion resistance, truthfulness, or individual rationality. The results of Fig. 3.8(a) and 3.8(b) are derived from the Guru dataset, with its *per unit weight bid* η distribution shown in Fig. 3.7(a). We find that our scheme achieves the above-mentioned properties at the cost of higher total payment. Specifically, as shown in



(a) Guru dataset (non-uniform bid distribution) (b) Guru dataset (non-uniform bid distribution) (c) Uniform bid distribution (d) Uniform bid distribution

Figure 3.8: Total payment caused by our scheme and P_{opt}.

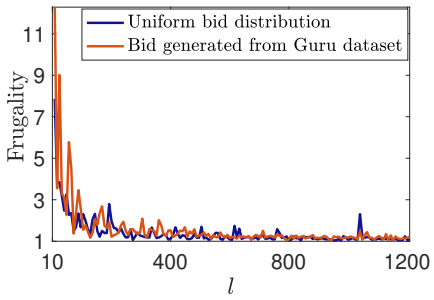


Figure 3.9: The frugality over different l's.

Fig. 3.8(a), 90-percentile of the total payment is 5.5×10^5, 8.1×10^5, and 15.8×10^5, respectively, under the settings of $(2, 0.8)$, $(4, 0.8)$, and $(8, 0.8)$, while P_{opt} is merely 1.1×10^5. We further evaluate in Fig. 3.8(c) and 3.8(d) the same metric under a synthetic worker bid dataset, with each element randomly generated following a uniform distribution. Its corresponding *per unit weight bid* distribution is plotted in Fig. 3.7(b). Apparently, the bid distribution impacts the total payment, in terms of both mean and variance. The average payment under Guru dataset is higher than that under uniform bid distribution. This is because our scheme applies a single-price scheme based on random rounding in each type segment, resulting in more winners under the Guru dataset. More specifically, when $l = 50$, the corresponding $r^* \cdot a$ falls within the range $[5, 65]$ and the per unit weight bids under Guru dataset mostly reside at the lower end of the distribution. Besides, the total payments under Guru dataset experience less variance than that under uniform bid distribution. This is because the Guru dataset generates a smaller value of r^H/r^L and thus a smaller θ to achieve the same p. Note that the possible range of payment is reduced as θ decreases.

Fig. 3.9 examines the *frugality* achieved by our scheme under different thresholds l's, the threshold determined by the platform to guarantee service quality. As discussed in Theorem 4, frugality quantifies the extra payment caused by our scheme compared with P_{opt}. We notice that frugality decreases as l grows. When l surpasses 400, frugality drops quickly approximating 1. It indicates that the platform barely overpays. On the other hand, a larger l indicates that more workers should be recruited for a given task. Therefore, the corresponding total payment will be enlarged too. Besides, when $l_j \in [10, 400]$, the frugality under Guru dataset is larger than that under uniformly distributed bids due to the same reason discussed above. Such a difference becomes negligible as l increases.

Fig. 3.10 shows the platform's average payment of a randomly selected task under different combinations of t and p. From Fig. 3.10(a), we observe that the platform's total payment increases as t grows. For example, when $p = 0.9$, the platform's payment is 8.0×10^3 at $t = 4$. This value becomes 1.2×10^4 at $t = 8$. The latter is about 1.5 times of the former. The similar trend is observed for $p = 0.7$ and $p = 0.8$. It implies that it costs

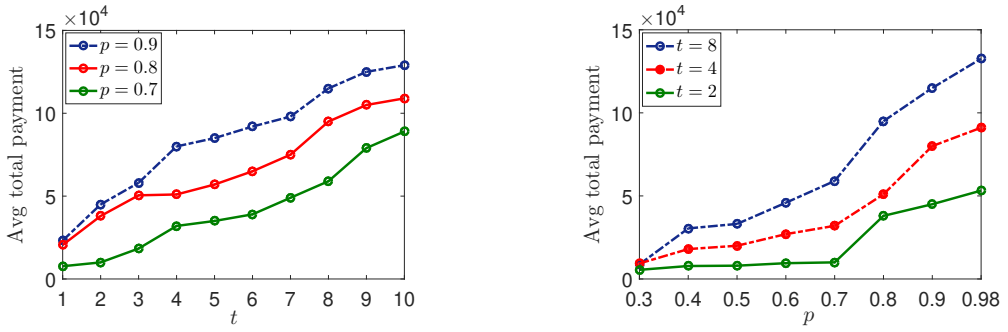


Figure 3.10: The platform's average total payment under different settings.

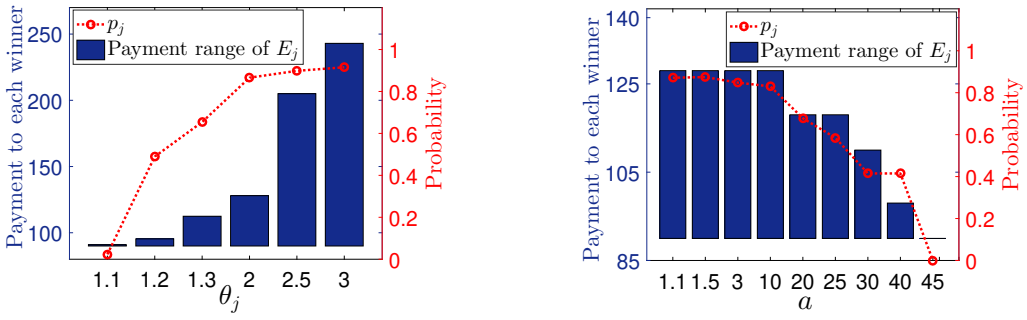


Figure 3.11: Scheme performances under different settings.

the platform more, in order to defend coalitions with larger weighted cardinality. We also notice that, under the same t, a larger p costs the platform more, as shown in Fig. 3.10(b). For example, when $t = 8$, the total payment is 5.9×10^3 with $p = 0.7$. It becomes 1.2×10^4 with $p = 0.9$. The latter is about twice the former. Hence, it costs the platform more in order to achieve a higher defense success probability. The reason can be briefly summarized as follows. For a specific $E_j \in \mathscr{E}$, in order to achieve a larger p_j, we are expecting a larger θ_j according to (3.13). Thus, from the definition of $h_u^{\theta_j}(\cdot)$ in (3.11), a winner is very likely to receive a higher payment $h_u^{\theta_j}(r_j^* \cdot a)$ for a given $r_j^* \cdot a$. We have a similar observation in Fig. 3.10(b).

Fig. 3.10 provides some insightful observations of our (t, p)-collusion resistant scheme. First, there is a tradeoff between t and the platform's total payment; to defend against coalitions of larger size, the platform has to pay more accordingly. A similar relation pertains to p and the platform's total payment; to achieve a higher defense success rate, the platform has to pay more too.

3.6.4 Impact of parameters

In this part, we evaluate the impact of different parameters on the performance of our scheme. Specifically, we concentrate on two of the most critical ones, θ, and a. A feasible worker set $E_j \in \mathscr{E}$ is randomly selected and examined with the payment range for each winner and p_j. The payment range is simply the possible value range of $h_u^{\theta_j}(r_j^* \cdot a)$ under different u and s. It provides another way to measure the average payment at the platform. Generally, a larger payment range leads to a larger average payment at the platform.

Fig. 3.11(a) depicts the impact of θ_j, where we fix $t_j = 2$ and $a = 2$. We observe that both the payment range and the probability p_j increases as θ_j grows. For example, when $\theta_j = 1.2$, the payment range to each winner in E_j is $[50, 55]$, while p_j is about 0.5. These two values become $[50, 62.5]$ and 0.9, respectively, when $\theta_j = 2$. It implies that in order to construct a more robust collusion-resistant scheme, i.e., a higher p_j, a larger θ_j is desirable, which, however, will result in a higher winner payment. On the other hand, a smaller θ_j can cost the platform less, but also renders the system more vulnerable to collusions. Hence, a suitable θ_j should be selected by balancing these two aspects. A similar tradeoff exists for parameter a in Fig. 3.11(b), where we set $t_j = 2$ and $\theta_j = 2$. Recall that a should also meet the requirement (3.4). When $a = 3$, the payment range for each winner and the probability p_j is $[90, 128]$ and 0.85, respectively. They are decreased to $[90, 90]$ and 0.2 when $a = 45$. We notice that a larger a leads to a lower payment range to each winner and thus a lower total payment at the platform, but also a lower collusion defense success rate; oppositely, a smaller a achieves a higher defense success rate, but a higher cost to the platform as well. The above results tell that suitable values of θ_j and a should be selected by balancing the aspects of the platform's payment and the scheme robustness.

3.7 Related Work

Collusion resistance in crowdsourcing. Collusion resistance has rarely been investigated in crowdsourcing. An initial research is conducted by Ji and Chen [95] with the focus on achieving *group strategy-proofness*, whereby a member that benefits from the coalition strategy will not pay off another member that suffers a loss [100]. Nonetheless, the scheme design for collusion resistance should further take into account the scenarios that members from the same coalition can exchange side-payment. Therefore, group strategy-proofness aims to prevent a particular form of collusions. Torshiz et. al [102] studied how to avoid worker collusions in reporting falsified results without being detected. Alternatively, we defend worker collusions during their economic interactions with the platform, so as to protect the platform and other benign workers from economic loss. Thus, we are working on a totally different problem.

Collusion resistance in spectrum auctions. Since collusions also happen in spectrum auctions [74, 75], collusion resistance is also investigated therein. Ji and Liu [125] proposed a collusion-resistant dynamic pricing approach to maximize the users' utilities while combating their collusive behaviors using the derived optimal reserve prices. [126, 127] also fall into the same line of research. However, these works are lack of formal proofs over their collusion resistance property. Besides, they only tackle a specific subset of collusions. Zhou et al. [128] then developed a general collusion-resistant framework for dynamic spectrum auctions.

Collusion resistance in general auctions. Since Robinson [73] gave theoretical evidence that auctions are vulnerable to collusions, there only have been a handful of works on collusion-resistant mechanism design [96, 97, 98, 99]. Che and Kim [97, 98] considered weaker colluders where members from the same coalition only have incomplete information regarding strategies adopted by each other. Nonetheless, in crowdsourcing, since all transactions are made online, it is pretty easy for colluders to share with their strategies offline without being detected. Penna and Ventre [99] developed a collusion-resistant mechanism assuming that the auctioneer has prior knowledge over bidders' behaviors. The assumption is questionable in real crowdsourcing systems where a large number of workers are involved.

Among the existing works, the one that is closest to ours is APM [96], which also resorts to the consensus estimate technique to derive a soft defense approach. Specifically, consensus estimate is applied to the winner number; collusions are discouraged for failing to change (the number of) winners in an auction. However, APM only works for generic auctions where bidders are homogeneous and only differ in bids. Winner selection is straightforward, e.g. picking the winners that offer top-l bids. In our problem, workers are heterogeneous for associated with various reputations. Winner selection further takes into account worker reputation as the quality constraint. By simply applying consensus estimate to the winner number may violate this constraint. Besides, APM relies on selecting the optimum set of winners, which is easy in generic auctions. Since worker recruitment is modeled as a binary integer programming problem, it is computationally intractable to derive its optimum result. Thus, the key ingredient of APM does not exist here. Alternatively, we novelly employ the consensus estimate over winner payments, which avoids the limitations of APM.

It is also worth mentioning some other related works on collusions [113, 114, 122, 123, 124, 115]. Notice that they focus on theoretical understanding of collusion performances under different auction settings. For example, [115] studies the impact of bidder collaboration in all-pay auctions. Instead, we aim to design a collusion-resistant scheme that prevents coalitions to rig auction outcomes in crowdsourcing.

Collusion detection in general auctions. As claimed by [47, 48], collusion might never be detected unless restrictive assumptions are imposed. The current existing works on collusion detection share a similar idea: They first derive important features in auctions without collusion and then show that one or more of these features are absent in collusive bids [48]. The fundamental assumption is that collusion strategies (e.g., who conducts collusion and how) and auction setting (e.g., bidder numbers, auction type, etc.) are consistent throughout auctions. Based on this, some works [49, 53] use various statistical tests to compare the bidding patterns of collusive and competitive bidders using bid price data of auctions that had been proven collusive in the past. Nonetheless, the information of collusive auctions, called a reference, may not be readily available. Another category of papers [48, 54] make improvement on these tests; they are able to identify collusive bidding pattern from the knowledge of bid prices of historical auctions without the reference. In addition to bidding price data, winning bid price [55], bid price-to-reserve price ratio [56], winning bid price to reserve price ratio with non-price attributes [57], etc., can also be used to derive important features that serve as the basis for collusion detection by showing features absence in collusive bids. Still, all the above works assume that colluders adopt consistent collusion strategies under a fixed auction setting. In practice, auctions are highly dynamic; besides, colluders can arbitrarily form coalitions and rig their bids without sticking to any pattern. In contrast, this work does not impose such assumptions; to be specific, we have no requirement on the availability of prior knowledge of collusion or the consistency of collusion strategies. Instead of passively detecting collusive behaviors, our defense scheme proactively prevents collusion via proper incentive design.

CHAPTER 4

ELICITING JOINT TRUTHFUL ANSWERS AND PROFILES FROM STRATEGIC WORKERS IN CROWDSOURCING SYSTEMS

4.1 Introduction

Crowdsourcing facilitates individuals and businesses to outsource their processes and jobs to a large pool of workers. Crowdsourcing has a wide spectrum of potential applications. For example, quite a few research propose to harness the sensing power of distributed devices for spectrum monitoring/sensing of a large geographic area [157, 32, 30, 31]. Specifically, under the framework of crowdsourcing, devices are hired to sense the spectrum occupancy/vacancy of their present locations. The aggregated sensing results can produce a real-time fine-grained spectrum usage map over a large geographic area. Crowdsourcing has also gained great interest in the field of wireless signal fingerprinting based indoor/outdoor localization [156, 155, 42, 44]. To reduce the effort of a manual calibration for the site survey, especially in a multi-floor building or a large geographic area, various kinds of crowdsourcing-based indoor localization methodologies have been successfully applied. In addition, there are plenty of commercial crowdsourcing platforms. For example, in Clickworker [43] some tasks hire workers with devices to carry out geolocation-aware image collection, image tagging, road traffic monitoring, etc. In Taskrabbit [85], the platform publishes spatial tasks such as cleaning a house or walking a dog. Typically, these tasks are only accessible by workers nearby.

In most crowdsourcing systems, the platform assigns tasks to suitable workers based on their self-reported profiles[1], such as locations and expertise.

A typical workflow can be divided into four stages: task assignment, task execution, answer collection, and answer aggregation/analysis. In general, task assignment problems are formulated to achieve certain optimization goals, e.g., maximizing the number of assigned tasks [152, 153] or minimizing overall cost (time, effort, and computing resources, etc.) incurred to workers [61, 60]. For practical considerations, the problem may further take into account various constraints, such as the maximum distance a worker is willing to travel to perform tasks, a worker's available time duration, and her expected work quality.

[1] We use the terms profiles and parameters interchangeably in the rest of the paper.

Work quality is of essential importance to the success of crowdsourcing tasks because low-quality answers from the crowd would easily deteriorate the accuracy of tasks via aggregation. Low quality is often attributed to workers' deliberate mis-reporting, lack of effort exertion, or free-riders copying results from peers. There have been some prior studies tackling the latter two cases [154, 158]. In this paper, we are interested in defending against more intelligent workers who may game the system through strategically reporting task answers for higher beneficial gain. Our discussion focuses on one of the most typical crowdsourcing tasks–*binary-answer*[2], e.g., if a specific spectrum band is vacant or not in the current location. A task is associated with a ground truth of a binary value. Each worker exerts her effort to derive an answer. To elicit truthful answers from the crowd, some existing solutions resort to economic mechanisms [104, 106, 105, 108, 109, 110, 111, 108]. Incentives are rewarded to truthful workers such that truth-telling is a *Nash equilibrium* [80]: no worker receives higher gain by lying, when others respond honestly. Therefore, no one would unilaterally deviate from honest reporting. It is worth noting that workers' truthful answers do not necessarily always coincide with the task ground truth, as workers may be hindered from, for example, the insufficient accuracy of smartphones' built-in GPS module.

Workers may also be deceitful about their self-reported profiles. These profiles are indispensable in task assignments. A strategic worker may fabricate her profile to manipulate task assignment outcomes for their own gain. For instance, a worker in location-based tasks is more likely to be selected if she misreports her position as being closer to the location of interest. To prevent workers from manipulating task assignment outcomes, one of the primary goals of this work is to achieve profile truthfulness. Some prior studies aim at improving *cost truthfulness* [134, 136, 137, 138, 60, 139, 157, 140], i.e., motivating workers to reveal their genuine costs. As cost can be deemed as part of self-reported profiles, cost truthfulness is a special case of profile truthfulness. Our approaches to achieve profile truthfulness need to cover a much wider spectrum of strategic behaviors.

Rather than treating answer misreporting and profile misreporting separately, this paper aims to develop a unified framework that protects two different stages from workers' strategic manipulation simultaneously. To the best of our knowledge, this is the first study to tackle such a combined challenge of misreporting in crowdsourcing. Under the framework of incentive design, there are existing solutions to elicit answer truthfulness [104, 106, 105, 108, 109, 110, 111, 108] and cost truthfulness [134, 136, 137, 138, 60, 139, 157, 140], respectively. However, they are not directly applicable here due to their neglect of the other aspect. We cannot simply apply the above schemes in different stages of crowdsourcing ei-

[2]Our scheme can be easily extended to tasks with multiple answers.

ther, i.e., a worker is first paid for answer truthfulness and then paid for cost truthfulness, as the total payment would violate the conditions for each of the two objects. Thus, our goal is to design a unified payment scheme that guarantees both answer and profile truthfulness.

Since a worker's true answers and profile are only known to herself, uncovering the worker's untruthful behavior is hard. Hence, instead of directly detecting if a worker lies or not, we take a proactive approach that focuses on prevention instead of passive detection. To be specific, we propose an incentive mechanism such that honestly providing both answers and profile is a *Nash equilibrium*. While there are various types of incentives to adopt, such as payment, reputation, and social recognition, our design assumes incentives in the form of monetary payment. The salient challenge in this approach is to design one payment to reward a worker for truth elicitation in two kinds of submissions. Our idea is to first derive the sufficient and necessary condition for answer truthfulness and profile truthfulness, separately. We then construct an incentive optimization problem that incorporates these conditions as constraints. Its optimal solution lists the payment to each worker. Since the solution must satisfy the constraints and thus the conditions for truth-telling, the workers are well motivated to behave honestly.

To derive each worker's sufficient and necessary condition for answer truthfulness, we use reference answers, i.e., reported answers from each worker's peers. As a worker's true observation toward a task is only known to herself, workers have "incomplete information". For example, workers are unaware of the platform's payment and each other's best strategies, i.e., which answers everyone else could report in order to maximize their own benefits (payments). To take this characteristic into account, this paper resorts to the model of *Bayesian game* [141] instead of the standard game model. A worker's payment is evaluated in its expectation with respect to her entire (binary) observation space, as a function of the worker's payments given the various reference reports instances and her *posterior belief*. To be specific, the posterior belief is the probability of the worker having a particular observation, given the observations of other workers. Then, by setting a worker's expected payment while truth-telling no less than that while lying, the worker has little incentive to lie, which is so-called a *Bayesian Nash equilibrium* [141]. By applying Bayesian inference, the posterior belief can be converted into an expression of the ground truth's prior probability and the conditional probability of a worker's answer given the ground truth. Both probabilities are practically known to the platform (Section 4.3.1). The idea of utilizing reference answers to derive the condition for answer truthfulness was also used in the *peer prediction* approach [104, 106, 105]. However, in their work, only one reference answer is randomly picked from a worker's peers to evaluate the worker's truthfulness. Such a simplistic method will misjudge a worker when the selected peer reports incorrectly. Instead, our approach is more robust as answers from all peers are taken into account.

To derive the sufficient and necessary condition for profile truthfulness, we design a randomized worker selection and worker payment approach. We first formulate an optimization problem for worker selection, i.e., assigning suitable workers to each task. Since this problem is NP-hard, we first relax the integrality constraint of each variable to its fractional domain and optimally solve the relaxed problem. Given that the fractional optimal solution is inapplicable to practical worker selection, it is then decomposed into a weighted sum of a set of feasible integer solutions. All weights are real-valued and range from 0 to 1, with their sum equal to 1. Then, we come up with a randomized worker selection; each feasible integer solution is randomly picked at a probability equal to its associated weight. To ensure the randomized worker selection is feasible to apply, the factional optimal solution needs to scale up by a factor η. According to [143], given any α-approximate algorithm that proves an *integrality gap* of at most η for the "natural" linear relaxation, one can use η as the scaling factor. Thus, an α-approximate algorithm to the worker selection problem is further developed. Once the worker selection outcome is determined, we set a worker's payment as η times the fractional payment derived from fractional VCG (Vickrey-Clarke-Groves) [144]. We note that the fractional VCG was originally developed to achieve *bid truthfulness* in generic auctions via proper payment design. In this work, we tailor it to tackle profile misreporting. The joint randomized worker selection and worker payment ensure profile truthfulness.

The contribution of this work is summarized as follows.

- We develop an incentive mechanism that aims to achieve comprehensive joint answer and profile truthfulness in crowdsourcing. The proposed mechanism ensures that truth-telling is a Bayesian Nash equilibrium.
- We propose a randomized worker selection algorithm and formally prove that the proposed algorithm produces an approximation ratio upper bounded by 2.
- To investigate the efficacy of our mechanism, a prototype consisting of a worker-side app and a platform-side program is implemented. We recruited 30 volunteers to conduct a series of in-field experiments. The full stack of code for the prototype implementation is open-sourced at `https://sites.google.com/site/reportingtruthful/`.

4.2 System model and Problem Statement

4.2.1 System Model

We consider a crowdsourcing system that consists of a set of workers and a platform. The platform has a set of tasks to obtain answers from a crowd $\mathscr{W} = \{w_1, \cdots, w_i, \cdots, w_K\}$ of K candidate workers. We consider binary-answer tasks and denote the answer space of each task as $\mathscr{A} = \{0, 1\}$. Besides, each task is associated with a ground truth $G \in \mathscr{A}$,

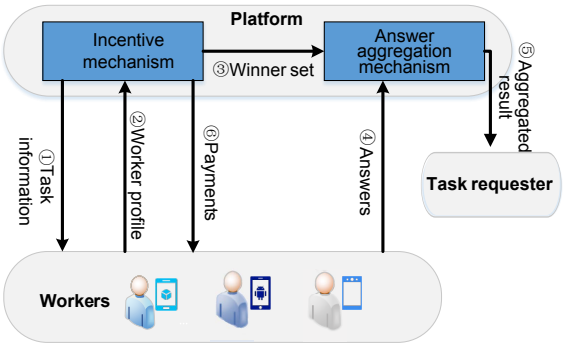


Figure 4.1: A typical workflow of a crowdsourcing system.

which is unknown prior to the accomplishment of tasks. A typical workflow of the system is illustrated in Figure 4.1.

- Step ①: The platform publishes a task.
- Step ②: Each worker w_i submits her profile.
- Step ③: The platform selects the winner set $\mathscr{W}^*$ for this task.
- Step ④: Each winning worker $w_i \in \mathscr{W}^*$ submits her answer r_i.
- Step ⑤: The platform derives the task's final result by aggregating collected answers using methods such as *majority voting* [145] or *maximum a posterior probability estimate* (MAP) [146].
- Step ⑥: The platform determines payments to winning workers.

In order to select a proper set of workers for a task, conventional approaches formulate an optimization problem that aims to maximize or minimize a certain objective while satisfying a set of constraints [136, 137, 138, 60, 139]. In this paper, we consider a linear optimization problem in a generalized form, as follows.

$$\begin{aligned} \mathrm{P}_1: \quad \min \quad & \pi_1 = \sum_{w_i \in \mathscr{W}} f(b_i) x_i \\ \text{s.t.} \quad & \sum_{w_i \in \mathscr{W}} d_i x_i \geq \gamma, \quad x_i \in \{0,1\} \quad \forall w_i \in \mathscr{W}. \end{aligned}$$

The platform solves P_1 to choose a proper set of workers $\mathscr{W}^*$ for the task. In P_1, x_i is a binary variable: 1 if worker w_i is to be selected, and 0 otherwise. The self-reported profile from w_i, denoted b_i, is a vector of task-dependent parameters. Take crowdsourcing-based spectrum monitoring/sensing as an illustration. Each device is associated with its specific sensing capabilities, such as the sensing range and operational frequency band. Besides, each device is at a different distance away from the task location. $f(\cdot)$ can be any weighted aggregate function over all elements in b_i. It is selected by the platform and unknown to

the workers. For example, suppose a worker's profile $b_i = \{s_i, t_i\}$ consists of her sensing capability t_i and its distance to the task location s_i, where s_i and t_i are normalized values. The function can be $f(b_i) = 0.8s_i + 0.2(1 - t_i)$. In this example, the worker's distance to the location of interest outweighs her sensing capability for being selected. Since b_i is both task-dependent and worker-dependent and thus *a prior* unknown to the platform, it is collected before the formulation of P_1. The scalar value of task-independent parameter d_i is available at the platform via long-term observation of w_i's performance, e.g., task accomplishment rate or average rating from task requesters. γ is a task-specific threshold value chosen by the platform.

4.2.2 Problem Statement and Design Objectives

Workers are modeled as rational and self-interested. They choose their strategies in a way to maximize their own benefits. Thus, workers may intentionally game the system. As discussed earlier, the platform collects data from workers for both worker selection and answer aggregation. Both procedures are thus vulnerable to manipulation. According to the formulation of P_1, w_i is more likely to be selected if she submits a fabricated profile b_i that produces a lower value of $f(b_i)$. Workers are also interested in reporting falsified answers, if it generates higher gains than truth-telling. Therefore, the primal goal of this work is to elicit worker truthfulness in reporting both their profiles and answers.

We propose to leverage incentives to motivate workers to behave honestly. The idea is simple: a worker receives the highest payment for truth-telling. A winning worker w_i's payment is expressed as $p_i(r_i, r_{-i}, b_i, b_{-i})$, where b_i and b_{-i} represent the worker profile reported by w_i and its peers, respectively, and r_i and r_{-i} are the answers reported by w_i and its peers, respectively. b_i, b_{-i} and r_{-i} are vectors. r_{-i} is the *reference answers* from peers. Intuitively, payment p_i is dependent of r_i and b_i. Besides, since the platform is unaware of a worker's true answer to the task, we propose to utilize reference answers to measure the worker's answer truthfulness. Hence, p_i also depends on r_{-i}. Besides, to achieve profile truthfulness, p_i is also dependent of b_{-i}. More details can be found in Section 4.3.2.

We now define truthfulness that is achieved by a proper incentive mechanism.

Definition 9. (Truthfulness.) *Given true reference answers o_{-i} and profiles c_{-i}, the mechanism achieves truthfulness if and only if every worker's best strategy is truth-telling, i.e.,*

$$\mathbb{E}_{\mathscr{A},\mathscr{I}}\left[p_i(r_i = o_i, b_i = c_i | o_{-i}, c_{-i})\right] \geq \mathbb{E}_{\mathscr{A},\mathscr{I}}\left[p_i(r_i = \overline{o}_i, b_i = \overline{c}_i | o_{-i}, c_{-i})\right] \tag{4.1}$$

where o_i and c_i denotes w_i's true answer and profile, $\overline{o}_i \neq o_i$ and $\overline{c}_i \neq c_i$.

The above definition forms a *Bayesian Nash equilibrium*, where no individual can gain higher payment on average by lying when others behave honestly. Therefore, no worker has the incentive to alter her reported answer or profile unilaterally, because such a strategy would harm her interest. Since a worker's true observation toward a task is unknown to the platform, her payment is evaluated in the expectation with respect to $\mathscr{A}$ in (4.1). Besides, our proposed mechanism relies on a randomized worker selection, i.e., winning workers are selected in a probabilistic manner. Thus, the expected payment further takes into account all possible worker selection outcomes $\mathscr{I}$. The details of the randomized worker selection and $\mathscr{I}$ are discussed in Section 4.3.2.

Here we make a brief clarification why a Bayesian game is considered. Bayesian games are games with incomplete information, which are, informally, games where players may not know all aspects of the game, such as sequence, strategies, and payoffs of other players. In contrast, standard games refer to games of complete information. In our paper, each worker, i.e., player, is unaware of its payoff function because the platform's payment and other workers' submitted information (i.e., profiles and answers) are unknown. Hence, the interactions among workers in our case should be formulated as a Bayesian game.

4.3 Mechanism Design

Our design first derives the sufficient and necessary conditions for answer and profile truthfulness separately. In the framework of incentive design, such a condition is expressed in the form of worker payments. Specifically, for the condition of answer truthfulness, a worker's payment is set higher for reporting her true observation toward a task than lying. Given that workers' true observations are only known to themselves, we utilize reference answers to evaluate workers' answer truthfulness (Section 4.3.1). Motivated by the fractional VCG, the profile truthfulness is achieved via the design of a randomized worker selection and worker payments (Section 4.3.2). Since the proposed randomized worker selection requires a *scaling factor* to ensure its feasibility, we further develop an α-approximate algorithm to obtain this value (Section 4.3.3). All the derived conditions are finally incorporated into an incentive optimization problem as constraints. Its solution provides the payment for each worker that motivates truth-telling in two different kinds of submissions (Section 4.3.4).

Table 4.1: Notation

$\mathscr{W}$	entire worker set	$\mathscr{W}^*$	winner set
G	ground truth	p_i	w_i's payment
c_i	w_i's true profile	b_i	w_i's reported profile
o_i	w_i's true answer	r_i	w_i's reported answer
$\mathscr{A}$	task answer set	x_i	recruitment variable
d_i	w_i's coefficient	γ	recruitment threshold
π_1^*, π_2^*, π_3^*, π_4^*	optimum result of P_1, P_2, P_3, P_4		
$\pi_1'^*$	optimum result of the linear relaxed P_1		
$\overline{o}_i$	opposite of true observation o_i		
c_{-i}	true profile set from w_i's peers		
b_{-i}	reported profile set from w_i's peers		
o_{-i}	true answer set from w_i's peers		
r_{-i}	reported answer set from w_i's peers		
$\mathscr{I}$	set of all possible task assignment outcomes		
θ	an instance of the true reference report		
$x(I)$	a feasible solution to P_1 under index I		
$\beta(I)$	selection probability of $x(I)$		
$\mathscr{I}$	the set of feasible solution to P_1		
x^F	optimum solution to the linear relaxed P_1		
α	approximation ratio of Algorithm 4		
δ	truth-telling payment margin		
η	integrality gap between π_1^* and π_2^*; upper bound of the approximation ratio of our mechanism		

4.3.1 Eliciting Truthful answers

This part derives a sufficient and necessary condition for answer truthfulness. For a given worker selection outcome $I \in \mathscr{I}$, (4.1) is then degenerated to $\mathbb{E}_{\mathscr{A}}[p_i(o_i|o_{-i}=\theta)] \geq \mathbb{E}_{\mathscr{A}}[p_i(\overline{o}_i|o_{-i}=\theta)]$[3]. θ is an instance of the true reference answer, which is a vector.

A belief regarding the prior probability of task ground truth $\Pr[G=a_t]$ ($a_t \in \mathscr{A}$) is deemed available at the platform. It also knows the likelihood that a worker honestly reports its genuine answer given the ground truth, i.e., $\Pr[o_i=a_i|G=a_t]$ ($a_i \in \mathscr{A}$). This knowledge can be obtained by the platform via long-term observation. Specifically, if the proposed mechanism achieves truthfulness, then $r_i=o_i$ and thus $\Pr[o_i=a_i|G=a_t] = \Pr[r_i=a_i|G=a_t]$. Since r_i is observable at the platform and the ground truth G would be eventually derived, $\Pr[r_i=a_i|G=a_t]$ can be obtained.

[3]Since we focus on analyzing the relation between answer reporting and incentives, b_i and c_{-i} are temporarily dropped for expression simplicity. p_i is still dependent on them.

We have

$$\mathbb{E}_{\mathscr{A}}\left[p_i(o_i|o_{-i}=\theta)\right]=\sum_{a_i\in\mathscr{A}}\Pr[o_i=a_i|o_{-i}=\theta]\cdot p_i(a_i,o_{-i}=\theta)$$

where $\Pr[o_i=a_i|o_{-i}=\theta]$ is the likelihood that given w_i's peers reporting honestly worker w_i does the same. We have

$$\Pr[o_i=a_i|o_{-i}=\theta]=\frac{\Pr[o_{-i}=\theta|o_i=a_i]\Pr[o_i=a_i]}{\Pr[o_{-i}=\theta]}. \tag{4.2}$$

Here, workers are assumed to report independently. Hence, $\Pr[o_{-i}=\theta|o_i=a_i]$ can be expressed by

$$\Pr[o_{-i}=\theta|o_i=a_i]=\prod_{w_k\in\mathscr{W}^*\setminus w_i}\Pr[o_k=a_k|o_i=a_i]$$

where
$$\Pr[o_k=a_k|o_i=a_i]=\sum_{a_t\in\mathscr{A}}\Pr[o_k=a_k|G=a_t]\Pr[G=a_t|o_i=a_i], \tag{4.3}$$

and

$$\Pr[G=a_t|o_i=a_i]=\frac{Pr[o_i=a_i|G=a_t]\Pr[G=a_t]}{\Pr[o_i=a_i]}=\frac{Pr[o_i=a_i|G=a_t]\Pr[G=a_t]}{\sum_{a_{t'}\in\mathscr{A}}\Pr[o_i=a_i|G=a_{t'}]\Pr[G=a_{t'}]}.$$

Recall that $\Pr[o_i=a_i|G=a_t]$ and $\Pr[G=a_t]$ are common knowledge at the platform. Thus $\Pr[o_k=a_k|o_i=a_i]$ (4.3) is derived. As $\Pr[o_i=a_i]=\sum_{a_t\in\mathscr{A}}\Pr[G=a_t]\cdot\Pr[o_i=a_i|G=a_t]$ and $\Pr[o_{-i}=\theta]=\prod_{w_k\in\mathscr{W}^*\setminus w_i}\Pr[o_k=a_k]$, then $\Pr[o_i=a_i|o_{-i}=\theta]$ (4.2) is also derived. Finally, $\mathbb{E}_{\mathscr{A}}\left[p_i(o_i|o_{-i}=\theta)\right]$ is obtained.

Similarly,

$$\mathbb{E}_{\mathscr{A}}\left[p_i(\overline{o}_i|o_{-i}=\theta)\right]=\sum_{a_i\in\mathscr{A}}\Pr[o_i=a_i|o_{-i}=\theta]\cdot p_i(\overline{a}_i,o_{-i}=\theta).$$

Therefore, the sufficient and necessary condition for answer truthfulness is

$$\sum_{a_i\in\mathscr{A}}\Pr[o_i=a_i|o_{-i}=\theta]\cdot p_i(a_i,o_{-i}=\theta)\geq\sum_{a_i\in\mathscr{A}}\Pr[o_i=a_i|o_{-i}=\theta]\cdot p_i(\overline{a}_i,o_{-i}=\theta). \tag{4.4}$$

Its instantiation is

$$\begin{aligned}&\Pr[o_i=1|o_{-i}]\cdot p_i(1,o_{-i})+\Pr[o_i=0|o_{-i}]\cdot p_i(0,o_{-i})\\ \geq&\Pr[o_i=1|o_{-i}]\cdot p_i(0,o_{-i})+\Pr[o_i=0|o_{-i}]\cdot p_i(1,o_{-i}).\end{aligned}$$

where $p_i(0, o_{-i})$ and $p_i(1, o_{-i})$ stand for w_i's payment when reporting "0" and "1", respectively, when o_{-i} is observed with an instance θ. If w_i's payment $p_i(0, o_{-i})$ and $p_i(1, o_{-i})$ satisfy the above constraint, a worker is more willing to report genuine observation for a higher payment in expectation. (4.5) will be integrated into the final mechanism design, which will be clear soon.

4.3.2 Eliciting Truthful Profiles

To elicit truthful profiles, inspired by a fractional version of the VCG mechanism [144], our design consists of randomized worker selection and worker payment.

Randomized worker selection. The objective of this part is to select a proper set of winning workers to carry out the task by solving P_1. Since P_1 is a *0-1 knapsack problem* [147], it is NP-hard to solve. We first consider its linear relaxed form by converting the binary variable $x_i \in \{0,1\}$ to $0 \leq x_i \leq 1$. The relaxed problem transforms P_1 into a fractional domain and returns a fractional optimum solution, denoted by x^F. Let $\pi_1'^*$ be the optimum result of the relaxed P_1. While x^F is inapplicable to task assignment, we are able to decompose it into a randomized format. Specifically, identify β_I and $x(I) = \{x_i(I)|\forall i\}$ such that $x^F = \sum_{I\in\mathscr{I}} \beta_I x(I)$, where $\mathscr{I} = \{x(I)|\forall I \in \mathscr{I}\}$ is the set of feasible integer solutions to P_1 and $\beta_I \geq 0$ ($\sum_{I\in\mathscr{I}} \beta_I = 1$). Then a randomized task assignment chooses the I-th integer solution $x(I)$ with probability β_I.

On the other hand, there does not exist a convex combination of integer solution $\sum_{I\in\mathscr{I}} \beta_I x_i(I)$ that equals x_i^F, because otherwise, the expected objective value generated by these integer solutions equals to that generated by the fractional solution, which is apparently a contradiction to the fact that the fractional solution achieves lower objective value than any possible integer solution. Therefore, to derive a feasible decomposition, we need to scale up the optimum fractional solution by a certain factor. According to [143], given any α-approximate algorithm that proves an *integrality gap* of at most η for the "natural" linear relaxation, one can use η as the scaling factor. We leave the job of finding such an approximation algorithm in Section 4.3.3.

The solution of β_I's is obtained via solving the following liner maximization problem

$$\begin{aligned} \max \quad & \sum_{I\in\mathscr{I}} \beta_I \\ \text{s.t.} \quad & \sum_{I\in\mathscr{I}} \beta_I x_i(I) \leq \eta x_i^F, \quad \forall i \\ & \sum_{I\in\mathscr{I}} \beta_I \leq 1,\ \beta_I \geq 0, \quad \forall I \in \mathscr{I}. \end{aligned} \tag{4.5}$$

Note that $x(I)$ is obtained by enumeration. Since it has an exponential number of elements, the enumeration process is time-consuming. Motivated by the *ellipsoid method* [148], we resort to its dual problem and propose an algorithm that solves it within polynomial time. Note that the α-approximate algorithm providing the scaling factor η is also essential to solve (4.5). Details are given in Appendix. Besides, we are able to prove in Lemma 4 that the optimum value of (4.5) is 1.

A toy example. Here we provide a toy example to better illustrate how the proposed random worker selection works. Consider that there are two workers and one task. We further assume the result from the relaxed P_1 and the approximation ratio as $x^F = (0.5, 0.5)$ and $\eta = 2$, respectively. Let $\mathscr{I}$, the set of feasible solution to P_1, as $\{(1,0),(1,1)\}$. We aim to find $\beta_{(1,0)}$ and $\beta_{(1,1)}$ such that $\beta_{(1,0)} \cdot 1 + \beta_{(1,1)} \cdot 1 = 2 \cdot 0.5$, $\beta_{(1,0)} \cdot 0 + \beta_{(1,1)} \cdot 1 = 2 \cdot 0.5$, and $\beta_{(1,0)} + \beta_{(1,1)} = 1$. Through simple calculation, we derive $\beta_{(1,0)} = 0$ and $\beta_{(1,1)} = 1$, which means our randomized worker selection chooses $x_1 = 1, x_2 = 1$ (resp., $x_1 = 1, x_2 = 0$) with probability 1 (resp., 0).

Worker payment. Recall that $\pi_1'^*$ is the optimum result of the linear relaxed P_1. Consider a factional payment $p_i^F = \pi_{1,-i}'^* - (\pi_1'^* - f_i(b_i)x_i^F)$, where $\pi_{1,-i}'^*$ stands for the optimum result of the linear relaxed P_1 when w_i is excluded from the formulation. The winner w_i's payment is then set to $p_i(I) = \eta p_i^F$. w_i is a winner if $x_i(I) = 1$ given a randomly picked worker selection outcome $I \in \mathscr{I}$. The calculation of p_i^F quantifies the *externality* each winner causes to others under fractional worker selection. As we will show in Theorem 8, paying a winner with its externality and the scale-up factor η are essential to ensure profile truthfulness.

Justification of applying fractional VCG. The worker selection problem P_1 is NP-hard and thus computationally expensive to find its optimum solution. We thus propose a randomized worker selection to achieve a polynomial complexity. Here we give a definition of fractional VCG.

Definition 10. (Fractional VCG.) *In a fractional VCG, the allocation rule is given by* $x^F = \{x_i^F | \forall i\}$*, the optimum allocation solution in the fractional domain; the pricing rule is given by* $p_i^F = \pi_{1,-i}'^* - (\pi_1'^* - f_i(b_i)x_i^F)$*, where* $\pi_1'^*$ *is the optimum allocation result in the fractional domain, and* $\pi_{1,-i}'^*$ *stands for the optimum result when* w_i *is excluded from the auction.*

Compared with conventional VCG, the allocation and pricing are determined through solving LP problems in fractional VCG. Therefore, it is more efficient to execute at the crowdsourcing platform. Apparently, our computation efficiency is achieved by compromising allocation optimum. Fortunately, as we prove in Theorem 3 in the paper, the optimality gap, quantified in approximation ratio here, is upper bounded by 2. We would also like to mention some other existing approaches in tackling computation complexity of VCG auctions. They typically develop heuristic algorithms in allocation and pricing while

achieving truthfulness and individual rationality simultaneously. Those approaches, however, are hard to bound the optimality gap especially when scenarios become complicated.

4.3.3 An α-Approximate Algorithm

This part develops an α-approximate algorithm that provides the scaling factor η for our randomized worker selection. It is also an indispensable component to the proposed algorithm to solve (4.5). We first transform P_1 into its equivalent form P_2.

$$\begin{aligned} \mathrm{P}_2: \quad \min \quad & \pi_2 = \sum_{w_i \in \mathscr{W}} f_i(b_i) x_i \\ \text{s.t.} \quad & \sum_{w_i \in \mathscr{W} \setminus S} d_i(S) x_i \geq \gamma(S), \forall S \subseteq \mathscr{W} : \gamma(S) > 0, \\ & x_i \in \{0,1\}, \forall w_i \in \mathscr{W} \end{aligned} \tag{4.6}$$

where S is an arbitrary subset of workers, $\gamma(S) = \gamma - \sum_{w_i \in S} d_i$, and $d_i(S) = \min\{d_i, \gamma(S)\}$. (4.6) involves a series of constraints with respect to S satisfying $S \subseteq \mathscr{W}$ and $\gamma(S) > 0$. For a given S, its corresponding x_i's are set to 1's.

Lemma 3. *P_2 is equivalent to P_1.*

Proof. Denote by $\mathscr{S}_1$ and $\mathscr{S}_2$ the feasible solution sets of P_1 and P_2, respectively. Since P_1 and P_2 have the same objective function, it is equivalent to show $\mathscr{S}_1 = \mathscr{S}_2$.

First, we show $\mathscr{S}_1 \subseteq \mathscr{S}_2$. For an arbitrary feasible solution $x \in \mathscr{S}_1$, denote its corresponding winner set as $\mathscr{W}^* = \{w_i | x_i = 1\}$. We have $\sum_{w_i \in \mathscr{W}^*} d_i \geq \gamma$, which can be transformed to

$$\sum_{w_i \in \mathscr{W}^* \setminus S} d_i \geq \gamma - \sum_{w_i \in S} d_i = \gamma(S), \tag{4.7}$$

with $S \subset \mathscr{W}^*$. Then we discuss through the following two cases that $x \in \mathscr{S}_1$ is also a feasible solution to P_2.

Case 1: There exist some workers $w_i \in \mathscr{W}^* \setminus S$ with $d_i \geq \gamma(S)$. Denote these workers as S'. Then $\sum_{w_i \in \mathscr{W} \setminus S} d_i(S) x_i = \sum_{w_i \in \mathscr{W}^* \setminus S} d_i(S) x_i = \sum_{w_i \in \mathscr{W}^* \setminus (S \cup S')} d_i + |S'| \gamma(S) \geq \gamma(S)$, which implies that (4.6) holds.

Case 2: No worker in $\mathscr{W}^* \setminus S$ has $d_i \geq \gamma(S)$. In another word, every worker in $\mathscr{W}^* \setminus S$ has $d_i < \gamma(S)$. We have $\sum_{w_i \in \mathscr{W} \setminus S} d_i(S) x_i = \sum_{w_i \in \mathscr{W}^* \setminus S} d_i(S) x_i = \sum_{w_i \in \mathscr{W}^* \setminus S} d_i \geq \gamma(S)$. The last inequality is due to (4.7). Therefore, (4.6) holds as well.

Second, we show $\mathscr{S}_2 \subseteq \mathscr{S}_1$. For an arbitrary feasible solution $x \in \mathscr{S}_2$, we have $\sum_{w_i \in \mathscr{W} \setminus S} d_i(S) x_i \geq \gamma(S) = \gamma - \sum_{w_i \in S} d_i$. It can be written as $\sum_{w_i \in \mathscr{W} \setminus S} d_i(S) x_i + \sum_{w_i \in S} d_i x_i \geq \gamma$, because $x_i = 1 \ \forall w_i \in S$. Besides, $d_i \geq d_i(S)$ according to the definition of $d_i(S)$. There-

fore, $\sum_{w_i \in \mathscr{W}} d_i x_i = \sum_{w_i \in \mathscr{W} \setminus S} d_i x_i + \sum_{w_i \in S} d_i x_i \geq \sum_{w_i \in \mathscr{W} \setminus S} d_i(S) x_i + \sum_{w_i \in S} d_i x_i \geq \gamma$, which implies that x is also a feasible solution to P_1.

From the discussion above, we have $\mathscr{S}_1 = \mathscr{S}_2$. Therefore, P_2 is equivalent to P_1. □

Since P_1 and P_2 are equivalent, if we can find an α-approximate algorithm for P_2, so it is for P_1.

Now consider a linear program P_3 by relaxing P_2's binary variable $x_i \in \{0,1\}$ to $0 \leq x_i \leq 1$. We further formulate a dual problem of P_3 but with the dual variables associated with constraints $0 \leq x_i \leq 1 (\forall i)$ dropped

$$\begin{aligned} \mathrm{P}_4: \quad \max \quad & \pi_4 = \sum_{S \subseteq \mathscr{W}: \gamma(S) > 0} \gamma(S) y(S) \\ \text{s.t.} \quad & \sum_{S \subseteq \mathscr{W}: w_i \in \mathscr{W} \setminus S, \gamma(S) > 0} d_i(S) y(S) \leq f_i(b_i), \forall w_i \in \mathscr{W} \\ & y(S) \geq 0, \forall S \subseteq \mathscr{W} \end{aligned} \tag{4.8}$$

Algorithm 4 outlines the steps for the proposed α-approximate algorithm. It leverages the formulation of P_4 to derive a feasible solution to P_1. Its idea is to gradually grow the winning worker set $S^{(t)}$ by selecting the worker which produces the smallest $(f_i(b_i) - q_i^{(t)})/d_i(S^{(t)})$ in each iteration (line 4). For each $S^{(t)}$, it then calculates the corresponding dual variable solution $y(S^{(t)})$. It continues until the termination condition $\gamma(S^{(t)}) \leq 0$ reaches.

Lemma 4. *Algorithm 4 provides a feasible solution to P_1 and P_4.*

Proof. We first examine if Algorithm 4 provides a feasible solution to P_1. According to the algorithm, x_i is either 0 or 1. Thus, the binary constraint is satisfied. Besides, the iteration of the algorithm stops when $\gamma(S) \leq 0$. Together with the definition of $\gamma(S)$, we have $\sum_{w_i \in \mathscr{W}} d_i \geq \sum_{w_i \in S} d_i \geq \gamma$. Then the other constraint of P_1 is also satisfied. Since the solution meets both constraints of P_1, it is feasible to P_1.

We next examine if Algorithm 4 provides a feasible solution to P_4. Suppose the while-loop consists of $T+1$ iterations. We first verify if $y(S^{(t)})$ identified in an arbitrary iteration $t+1, \forall t \in [0,T]$ is non-negative. Particularly, when $t = 0$, $f_i(b_i) \geq q_i^{(0)} = 0$, and thus $y(S^{(0)})$ is positive; when $t \in [1,T]$

$$\begin{aligned} q_i^{(t)} &= q_i^{(t-1)} + d_i(S^{(t-1)}) y(S^{(t-1)}) = q_i^{(t-1)} + d_i(S^{(t-1)}) \frac{f_{i^{*(t)}}(b_{i^{*(t)}}) - q_{i^{*(t-1)}}^{(t-1)}}{d_{i^{*(t-1)}}(S^{(t-1)})} \\ &\leq q_i^{(t-1)} + d_i(S^{(t-1)}) \frac{f_i(b_i) - q_i^{(t-1)}}{d_i(S^{(t-1)})} = f_i(b_i). \end{aligned}$$

Algorithm 4 The α-approximate algorithm

Input: $\{b_i\}$, $\{d_i\}$, γ
Output: $\{x_i\}$, $\{y(S)\}$, π_1, $\mathscr{W}^*$

$x_i \leftarrow 0$, $q_i^{(0)} \leftarrow 0$, $\forall i$, $y(S) \leftarrow 0$, $\forall S$, $S^{(0)} \leftarrow \emptyset$, $\gamma(S^{(0)}) \leftarrow \gamma - \sum_{w_i \in S^{(0)}} d_i$, $\pi_1^{(0)} \leftarrow 0$, $t \leftarrow 0$;
while $\gamma(S^{(t)}) > 0$ **do**
  $d_i(S^{(t)}) \leftarrow \min\{d_i, \gamma(S^{(t)})\}$, $\forall i$;
  $i^{*(t)} \leftarrow \arg\min_{i \in \mathscr{W} \setminus S^{(t)}} (f_i(b_i) - q_i^{(t)})/d_i(S^{(t)})$;
  $y(S^{(t)}) \leftarrow \left(f_{i^{*(t)}}(b_{i^{*(t)}}) - q_{i^{*(t)}}^{(t)}\right)/d_{i^{*(t)}}(S^{(t)})$;
  $x_{i^{*(t)}} \leftarrow 1$;
  $\pi_1^{(t+1)} \leftarrow \pi_1^{(t)} + f_{i^{*(t)}}(b_{i^{*(t)}})$;
  $S^{(t+1)} \leftarrow S^{(t)} \cup w_{i^{*(t)}}$;
  $q_i^{(t+1)} \leftarrow q_i^{(t)} + d_i(S^{(t)})y(S^{(t)})$, $\forall i \in \mathscr{W} \setminus S^{(t)}$;
  $\gamma(S^{(t+1)}) = \gamma - \sum_{w_i \in S^{(t+1)}} d_i$;
  $t \leftarrow t+1$;
end while
$\mathscr{W}^* \leftarrow S^{(t)}$, $\pi_1 \leftarrow \pi_1^{(t)}$.

Thus, $q_{i^{*(t)}}^{(t)} \leq f_{i^{*(t)}}(b_{i^{*(t)}})$. As $d_{i^{*(t)}}(S^{(t)}) > 0$, then we have $y(S^{(t)}) = (f_{i^{*(t)}}(b_{i^{*(t)}}) - q_{i^{*(t)}}^{(t)})/d_{i^{*(t)}}(S^{(t)}) \geq 0$.

The remaining task is to verify if $\{y(S) : S \subseteq \mathscr{W}\}$ has the constraint (4.8) hold for all $w_i \in \mathscr{W}$. For this purpose, we divide $\mathscr{W}$ into two non-overlapping subsets: $\mathscr{W}^*$ and $\mathscr{W} \setminus \mathscr{W}^*$, i.e., the winning worker set and the losing worker set.

Case 1: $w_{i^{*(t)}} \in \mathscr{W}^*$, a worker selected in the arbitrary $(t+1)$-th ($t \in [0,T]$) iteration. We have

$$\sum_{S \subseteq \mathscr{W}: w_{i^{*(t)}} \in \mathscr{W} \setminus S, \gamma(S) > 0} d_{i^{*(t)}}(S) y(S) = \sum_{\tau=0}^{t} d_{i^{*(t)}}(S^{(\tau)}) y(S^{(\tau)})$$

$$= \sum_{\tau=0}^{t-1} d_{i^{*(t)}}(S^{(\tau)}) y(S^{(\tau)}) + d_{i^{*(t)}}(S^{(t)}) \frac{(f_{i^{*(t)}}(b_{i^{*(t)}}) - q_{i^{*(t)}}^{(t)})}{d_{i^{*(t)}}(S^{(t)})} = q_{i^{*(t)}}^{(t)} + f_{i^{*(t)}}(b_{i^{*(t)}}) - q_{i^{*(t)}}^{(t)} = f_{i^{*(t)}}(b_{i^{*(t)}}).$$

Thus, (4.8) is satisfied for all winning workers.

Case 2: $w_i \in \mathscr{W} \setminus \mathscr{W}^*$, any losing worker. We have $(f_i(b_i) - q_i^{(T)})/d_i(S^{(T)}) \geq (f_{i^*(T)}(b_{i^*(T)}) - q_{i^*(T)}^{(T)})/d_{i^*(T)}(S^{(T)})$. Therefore,

$$\sum_{S \subseteq \mathscr{W}: i \in \mathscr{W} \setminus S, \gamma(S) > 0} d_i(S) y(S) = \sum_{\tau=0}^{T} d_i(S^{(\tau)}) y(S^{(\tau)})$$

$$= \sum_{\tau=0}^{T-1} d_i(S^{(\tau)}) y(S^{(\tau)}) + d_i(S^{(T)}) \frac{(f_{i^*(T)}(b_{i^*(T)}) - q_{i^*(T)}^{(T)})}{d_{i^*(T)}(S^{(T)})} \leq q_i^{(T)} + d_i(S^{(T)}) \frac{(f_i(b_i) - q_i^{(T)})}{d_i(S^{(T)})} = f_i(b_i)$$

which implies that (4.8) holds for workers from $\mathscr{W} \setminus \mathscr{W}^*$ as well.

According to the analysis above, Algorithm 4 provides a feasible solution to P_4.

□

Proposition 5. *Algorithm 4 provides an α-approximation solution to P_1 where $\alpha = 2$, i.e., $\pi_1/\pi_1^* \leq 2$.*

Proof. Let w_{i^*} denote the worker selected in the last iteration by Algorithm 4. The while loop continues as long as $\gamma(S) > 0$. Then $\gamma(\mathscr{W}^* \setminus w_{i^*}) = \gamma - \sum_{w_i \in \mathscr{W}^* \setminus w_{i^*}} d_i > 0$ and thus

$$\sum_{w_i \in \mathscr{W}^* \setminus w_{i^*}} d_i < \gamma. \tag{4.9}$$

Besides,

$$\pi_1 = \sum_{w_i \in \mathscr{W}^*} f_i(b_i) = \sum_{w_i \in \mathscr{W}^*} \sum_{S \subseteq \mathscr{W}: w_i \in \mathscr{W} \setminus S, \gamma(S) > 0} d_i(S) y(S) = \sum_{S \subseteq \mathscr{W}: \gamma(S) > 0} \sum_{w_i \in \mathscr{W}^* \setminus S} d_i(S) y(S). \tag{4.10}$$

The second equality can be easily inferred from the proof of Lemma 2. The third equality is obtained by switching the order of the two sum operations. Note that

$$\sum_{w_i \in \mathscr{W}^* \setminus S} d_i(S) \leq \sum_{w_i \in \mathscr{W}^* \setminus w_{i^*}} d_i - \sum_{w_i \in S} d_i + d_{i^*}(S) \leq \gamma - \sum_{w_i \in S} d_i + d_{i^*}(S) = \gamma(S) + d_{i^*}(S) \leq 2\gamma(S)$$

where the second inequality is due to (4.9). Let π_4^* be the optimum value of P_4. Due to the strong duality, $\pi_4^* \leq \pi_1^*$. Combining all the results above, we have

$$\pi_1 \leq \sum_{S \subseteq \mathscr{W}: \gamma(S) > 0} 2\gamma(S) y(S) \leq 2\pi_4^* \leq 2\pi_1^*.$$

□

Based on Proposition 5 and its proof, we have π_1 derived by Algorithm 4 and π_3^* have an integrality gap of at most η, i.e., $\pi_1/\pi_3^* \leq \eta$, where $\eta = 2$.

Lemma 5. *The computation complexity of Algorithm 4 is $\mathscr{O}(K^2)$.*

The complexity of Algorithm 4 is dominated by the while-loop, which contains at most K iterations. Recall that K stands for the number of workers. In each iteration, the most time-consuming calculation is ranking (line 4), which is upper-bounded by K operations. Thus, the complexity is upper bounded by $\mathscr{O}(K^2)$.

While there have been some prior works [149] developing approximation algorithms to solve 0-1 knapsack problems, they all face a trade-off between iteration convergence and computation efficiency. Particularly, when updating the dual variable, i.e., line 5 in our algorithm, their solution does not specify the exact step size. As a result, if the step size is too small, then it takes excessively long rounds for the algorithm to stop; if the step size is too large, then the algorithm may fail to converge. Instead, Algorithm 4 identifies a proper step size, $\left(f_{i^*(t)}(b_{i^*(t)}) - q_{i^*(t)}^{(t)}\right)/d_{i^*(t)}(S^{(t)})$, for each iteration that ensures the convergence within K iterations.

4.3.4 Piecing All Components Together

We are now ready to integrate all ingredients into a unified payment mechanism that elicit joint answer and profile truthfulness from strategic workers.

For the first objective, as discussed in Section 4.3.1, its sufficient and necessary condition is that each winning worker receives no less payment when reporting honestly than lying, as specified in (4.4). Practical mechanisms require certain margins for truth-telling [63]; honest reporting is better than lying by at least some margin δ, chosen by the platform to offset the external benefits a worker might obtain by lying. Thus, we twist a little bit over (4.4) and obtain (4.11) to account for the margin δ.

For the second objective, for a given worker selection outcome I, a winning worker w_i receives ηp_i^F. On the other hand, since w_i's true observation is unknown to the platform, we thus let w_i's expected payment with respect to $\mathscr{A}$ equal to ηp_i^F (4.12).

Combining all discussions above, we arrive at the following formulation in calculating winner w_i's payment, given a randomly picked worker selection outcome $I \in \mathscr{I}$

$$
\begin{aligned}
\mathrm{P}_5: \quad & \max \quad \delta \\
\text{s.t.} \quad & \sum_{a_i \in \mathscr{A}} \Pr[o_i = a_i | o_{-i}] p_i(a_i, o_{-i}) - \sum_{a_i \in \mathscr{A}} \Pr[o_i = a_i | o_{-i}] p_i(\overline{a}_i, o_{-i}) \geq \delta \quad (4.11) \\
& \sum_{a_i \in \mathscr{A}} \Pr[o_i = a_i | o_{-i}] p_i(a_i, o_{-i}) = \eta p_i^F \quad (4.12) \\
& p_i(a_i, o_{-i}) \geq 0, \quad \forall a_i \in \mathscr{A}.
\end{aligned}
$$

The above optimization is a linear programming problem that can be efficiently solved via the conventional *simplex method* [64].

Algorithm 5 summarizes our final design.

Algorithm 5 The final design

Input: $\{b_i\}$, $\{r_i\}$
Output: $\mathscr{W}^*$, $\{\beta_I\}$, $\{p_i\}$

1: Compute the optimal fractional worker selection x^F by solving the relaxed P_1;
2: Compute the scale-up factor η using Algorithm 4;
3: Derive β_I's by formulating and solving (4.5);
4: Select each integer solution $x(I)$ of P_1 randomly with probability β_I, thus $\mathscr{W}^*$ is derived;
5: Calculate winner's payment $p_i(a_i, o_{-i})$ ($a_i \in \mathscr{A}$) by solving P_5.

Theorem 6. *The computation complexity of Algorithm 5 is* $\mathscr{O}(K^2)$.

Proof. The computation of Algorithm 5 mainly consists of the following components, solving the relaxed P_1 (line 1), obtaining η using Algorithm 4 (line 2), solving (4.5) (line 3), and solving P_5 (line 5). In the following, we provide an analysis of these four components. We employ the simplex method to solve the relaxed P_1, an LP problem. According to [65], the computation complexity of simplex method is $\mathscr{O}(nd)$, where n and d are the number of variables and constraints, respectively. Thus, the computation complexity for solving the relaxed P_1 is $\mathscr{O}(K)$. Recall that K is the number of workers. Similarly, the complexity for solving P_5 is $\mathscr{O}(K)$. For Algorithm 4, its complexity is upper bounded by $\mathscr{O}(K^2)$. While (4.5) is an LP problem, it involves an exponential number of variables. Thus, we first convert it to its dual problem and then solve it via the ellipsoid method, whose complexity is at most $\mathscr{O}(n^2)$, where n is the number of variables. Thus, solving (4.5) causes $\mathscr{O}(K^2)$. To sum up, the computation complexity of the proposed mechanism is $\mathscr{O}(K^2)$. □

4.3.5 Extension to Multi-choice Tasks

When considering multiple-choice tasks, the change is applied to the instantiation of the sufficient and necessary condition for answer truthfulness, the form is still the same though

$$\sum_{a_i \in \mathscr{A}} \Pr[o_i = a_i | o_{-i} = \theta] \cdot p_i(a_i, o_{-i} = \theta) \geq \sum_{a_i \in \mathscr{A}} \Pr[o_i = a_i | o_{-i} = \theta] \cdot p_i(\overline{a}_i, o_{-i} = \theta),$$

which denotes that truth-telling achieves higher payment than lying. Multiple-choice tasks introduces more choices of lying than binary-choice tasks. Specifically, for each $o_i = a_i$, there are $k-1$ lying cases. Thus, more cases are generated in its instantiation. All the instantiations will serve as constraints in P_5 to get final payments. Recall that P_5 is an LP problem, which can be efficiently solved by the simplex method with the computation complexity $\mathscr{O}(nd)$, where n and d are the number of variables and constraints, respectively. Hence, although multi-choice tasks would increase the number of constraints in P_5, it would not cause any change to our scheme design.

Theorem 7. *Given K workers and M k-choice tasks, the computation complexity of Algorithm 2 is* $\mathscr{O}(MK^2 + MKk(k-1)^k)$

Proof. The proof follows the main idea of Theorem 1. Here we mainly highlight the changes due to multi-choice tasks. Under the k-choice setting, P_5 becomes the linear programming problem with $(k-1)^k$ constraints and k variables, where k is the number of choices in each task. Therefore, the complexity of P_5 for K workers is $\mathscr{O}(Kk(k-1)^k)$. Hence the computation complexity of the proposed mechanism for one k-choice task is $\mathscr{O}(K+K^2+K^2+Kk(k-1)^k) = \mathscr{O}(K^2+Kk(k-1)^k)$. □

Although Theorem 1 seems to indicate an exponential computation complexity, the exponential part k is typically a small value no larger than 6. Hence, the overall computation is still practical to implement on a regular server.

4.4 Performance Analysis

In this section, we analyze the properties achieved by our mechanism, including joint answer and profile truthfulness and its approximation ratio.

Theorem 8. *The proposed mechanism guarantees joint answer and profile truthfulness.*

Proof. According to Definition 9, we need to prove $\mathbb{E}_{\mathscr{A},\mathscr{I}}\left[p_i(o_i,c_i)\right] \geq \mathbb{E}_{\mathscr{A},\mathscr{I}}\left[p_i(\overline{o}_i,\overline{c}_i)\right]$[4]. For this purpose, we first show $\mathbb{E}_{\mathscr{A},\mathscr{I}}\left[p_i(o_i,c_i)\right] \geq \mathbb{E}_{\mathscr{A},\mathscr{I}}\left[p_i(o_i,\overline{c}_i)\right]$ and then $\mathbb{E}_{\mathscr{A},\mathscr{I}}\left[p_i(o_i,\overline{c}_i)\right] \geq \mathbb{E}_{\mathscr{A},\mathscr{I}}\left[p_i(\overline{o}_i,\overline{c}_i)\right]$.

Specifically,

$$\begin{aligned}\mathbb{E}_{\mathscr{A},\mathscr{I}}[p_i(o_i,c_i)] &= \sum_{I\in\mathscr{I}} \beta_I \mathbb{E}_{\mathscr{A}}[p_i(o_i,c_i)] = \sum_{I\in\mathscr{I}} \beta_I \sum_{a_i\in\mathscr{A}} \Pr[o_i=a_i]p_i(a_i) = \sum_{I\in\mathscr{I}} \beta_I \eta\, p_i^F(c_i) \\ &= \eta\, p_i^F(c_i)\end{aligned}$$

[4]For expression simplicity, we omit o_{-i} and c_{-i} from p_i in the following discussion.

which is exactly η times w_i's payment under the fractional VCG mechanism when reporting c_i. Similarly, we have $\mathbb{E}_{\mathscr{A},\mathscr{I}}[p_i(o_i,\overline{c}_i)] = \eta\, p_i^F(\overline{c}_i)$, i.e., η times w_i's payment under the fractional VCG mechanism when reporting untruthful $\overline{c}_i$. On the other hand, as proved in [144], the fractional VCG guarantees $p_i^F(c_i) \geq p_i^F(\overline{c}_i)$, and thus $\mathbb{E}_{\mathscr{A},\mathscr{I}}[p_i(o_i,c_i)] \geq \mathbb{E}_{\mathscr{A},\mathscr{I}}[p_i(o_i,\overline{c}_i)]$.

When w_i submits $\overline{c}_i$, under a specific feasible worker recruitment profile $I \in \mathscr{I}$, w_i either wins or loses. The following discussion is conducted for these two cases, separately.

For the first case, w_i loses with $\overline{c}_i$. Then $\mathbb{E}_{\mathscr{A}}[p_i(o_i,\overline{c}_i)] = \mathbb{E}_{\mathscr{A}}[p_i(r_i,\overline{c}_i)] = 0$. Since w_i loses, it will not be selected. Hence, its payment is 0.

For the second case, w_i wins with $\overline{c}_i$. Then

$$\mathbb{E}_{\mathscr{A}}[p_i(o_i)] - \mathbb{E}_{\mathscr{A}}[p_i(\overline{o}_i)] = \sum_{a_i \in \mathscr{A}} \Pr[o_i = a_i] \cdot (p_i(a_i) - p_i(\overline{a}_i)) \geq \delta.$$

Combining these two cases, we have $\mathbb{E}_{\mathscr{A}}[p_i(o_i,\overline{c}_i)] \geq \mathbb{E}_{\mathscr{A}}[p_i(\overline{o}_i,\overline{c}_i)]$. Therefore, $\mathbb{E}_{\mathscr{A},\mathscr{I}}[p_i(o_i,\overline{c}_i)] = \sum_{i\in\mathscr{I}} \beta_I \cdot \mathbb{E}_{\mathscr{A}}[p_i(o_i,\overline{c}_i)] \geq \sum_{i\in\mathscr{I}} \beta_I \cdot \mathbb{E}_{\mathscr{A}}[p_i(\overline{o}_i,\overline{c}_i)] = \mathbb{E}_{\mathscr{A},\mathscr{I}}[p_i(\overline{o}_i,\overline{c}_i)]$. □

It is desirable to analyze the approximation ratio of the proposed mechanism to the optimum result of P_1, where truthfulness is not guaranteed. It evaluates the optimality tradeoff for truthfulness.

Theorem 9. *The proposed mechanism achieves the approximation ratio upper bounded by 2.*

Proof. The expected overall objective value achieved by the proposed mechanism is formulated by $\sum\limits_{I\in\mathscr{I}} \beta_I \sum\limits_{w_i\in\mathscr{W}} f_i(b_i)x_i(I)$. Thus, the approximation ratio is calculated as

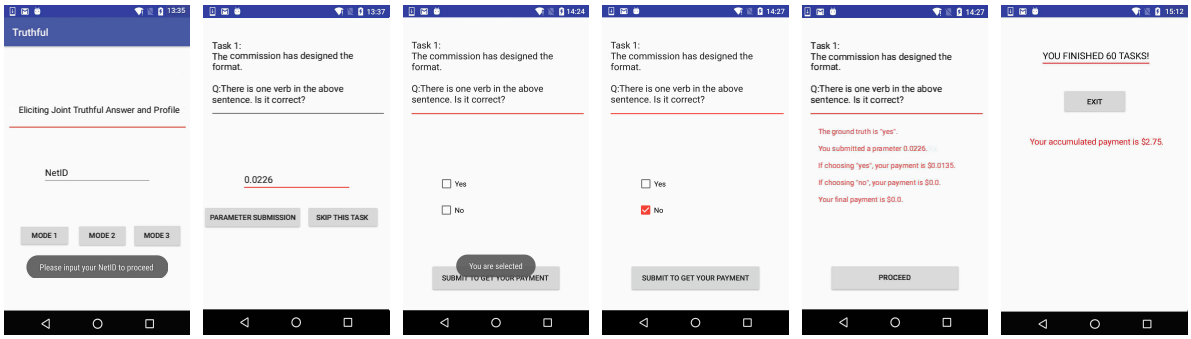
$$\frac{\sum\limits_{I\in\mathscr{I}} \beta_I \sum\limits_{w_i\in\mathscr{W}} f_i(b_i)x_i(I)}{\pi_1^*} = \frac{\sum\limits_{w_i\in\mathscr{W}} f_i(b_i)(\sum\limits_{I\in\mathscr{I}} \beta_I x_i(I))}{\pi_1^*} = \eta \frac{\sum\limits_{w_i\in\mathscr{W}} f_i(b_i)x_i^F}{\pi_1^*} = \eta \frac{\pi_3^*}{\pi_1^*} \leq \eta$$

where $\eta = 2$. □

4.5 Experimental Evaluation

4.5.1 Experimental Setup

As a proof-of-concept implementation, we develop a prototype of the proposed mechanism. The prototype mainly consists of the worker-side app and the platform-side program. Specifically, the app is developed in Android. The platform program runs on a Dell laptop with a 1.6GHz processor and 16GB RAM. To facilitate the task publication and



(a) Login page (b) Reporting profile (c) Selecting winners (d) Reporting answer (e) Payment for one task (f) Accumulated payment

Figure 4.2: Screenshots of worker-side app.

data reporting, the web server, named HTTP File Server (HFS) [66], is utilized. To investigate the performance of our mechanism, in-field experiments involving 30 volunteers have been conducted. For each task, the worker-side app also generates a random number from $[0.005, 0.075]$ to represent the volunteer's self-reported profile[5], which is unknown by the platform. There are 60 tasks in total. All of them are binary-answer English grammar questions from commonly used real-world crowdsourcing datasets [67]. An on-site training workshop on app usage was provided to volunteers before the experiment. The experiment procedure is briefly summarized as follows. Once the platform publishes a task, workers submit their reported profiles (Figure 4.2(b)). The volunteer decides if to submit the same profile as generated by the app or a different one. Then the platform determines which workers to recruit. The selected workers then solve the question and send back the answers, i.e., answers (Figure 4.2(c) and 4.2(d)). The platform determines payment to each worker for this task (Figure 4.2(e)). A worker's final payment is the accumulated amount it receives in all tasks (Figure 4.2(f)).

We developed our own prototype instead of using existing crowdsourcing platforms, such as Amazon Mechanical Turk or Flower Eight, due to the complex nature of our payment rule and the centralized worker selection. In these platforms, a worker's payment is predeclared and generally fixed. Thus, dynamic incentives cannot be implemented. All of our source codes are available online[6].

For comparison purposes, the experiment is also conducted with another two incentive mechanisms. The first one, called *random VCG*, employs the random VCG auction framework for worker selection and payment calculation so as to achieve profile truthfulness. The second one is called *truth serum* [158] that was designed for answer truthfulness.

[5]In the experiments, there is only one parameter in a worker's profile for simplicity.

[6]https://sites.google.com/site/reportingtruthful/

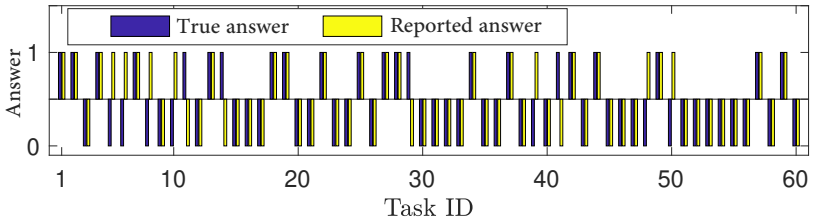
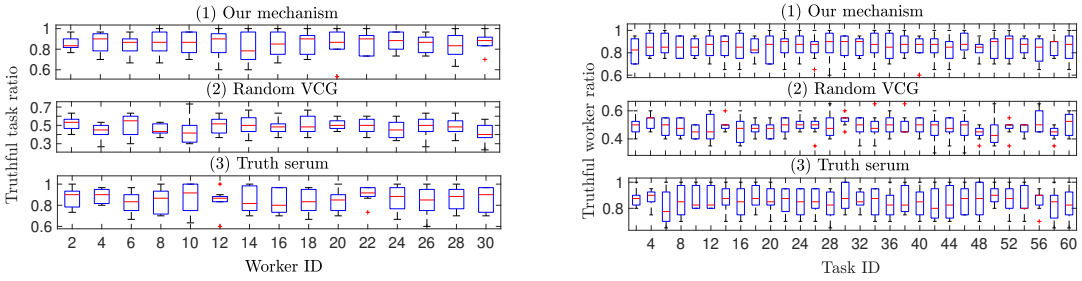


Figure 4.3: A randomly selected worker's reported and true answer for 60 tasks in our mechanism.

Specifically, truth serum extracts a worker's posterior belief from its reported answer and scores it using reference answers. The scoring rule is carefully designed such that truth-telling is a Bayesian Nash equilibrium.

4.5.2 Analysis of Answer Truthfulness

This section evaluates answer truthfulness. Since all tasks are relatively easy grammar questions for college students, we assume that their genuine answer to a task is the same as its ground truth.



(a) Truthful task ratio

(b) Truthful worker ratio

Figure 4.4: Answer elicitation performance comparison among our mechanism, random VCG, and truth serum.

Figure 4.3 shows a randomly selected worker's reported answer and the ground truth across the entire 60 tasks. We use "1" and "0" to denote the answer "yes" and "no", respectively. This worker misreports more often at the beginning but tends to be honest later. Particularly, 7 tasks are misreported among the first half batch, while this number drops to 4 for the second half batch. Since the worker gets the best-expected payoff when behaving honestly, it gradually adjusts strategies to truth-telling.

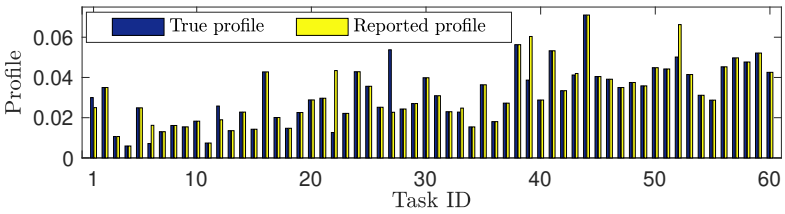
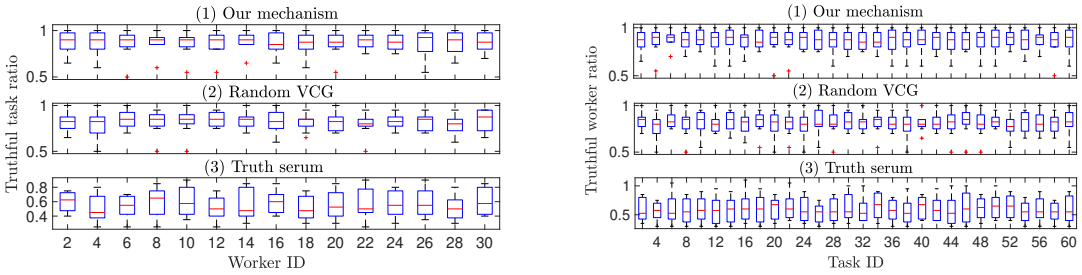


Figure 4.5: A randomly selected worker's reported and true profile for 60 tasks in our mechanism.



(a) Truthful task ratio (b) Truthful worker ratio

Figure 4.6: Profile elicitation performance comparison among our mechanism, random VCG and truth serum.

Figure 4.4(a) examines the *truthful task ratio*, which is defined as the percentage of tasks that a worker honestly reports among all the tasks. We observe that the ratio achieved by our mechanism and truth serum is around 0.9, while that for the random VCG is as low as 0.5. Thus, workers demonstrate no preference in answer reporting when truth elicitation is not enforced. Our mechanism performs as good as truth serum in motivating truth-telling. However, the latter does not consider profile truthfulness, which will be discussed in the next section. In addition, we compare in Figure 4.4(b) the *truthful worker ratio*, which is defined as the percentage of honest-reporting workers for a given task, among the three mechanisms. Similarly, ours has a similar performance as truth serum. Both of them outperform the random VCG. It is worth mentioning that neither our mechanism nor truth serum can guarantee perfect truth-telling in real-world experiments. This is because workers are modeled as idealized "rational individuals" with perfect knowledge to act in the paper, which may not be reflective of their actual status in real-world scenarios.

4.5.3 Analysis of Profile Truthfulness

Figure 4.5 depicts a randomly selected worker's reported profile and its genuine value for the 60 tasks. We observe a similar trend as in Figure 4.4: the worker is more likely to misreport at the beginning but tends to behave honestly after a few rounds of task executions. This is also because our mechanism effectively encourages workers to report true

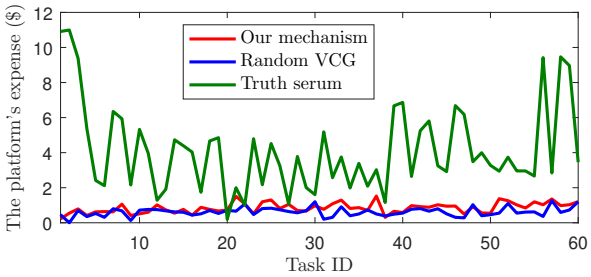


Figure 4.7: The platform's expense for different tasks.

Table 4.2: Accuracy performance comparison.

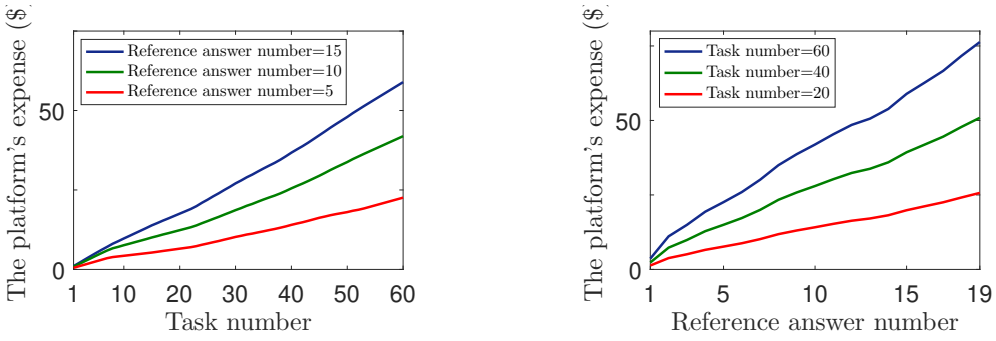
	Accuracy	Error	Accuracy ratio
Our mechanism	55	5	91.7%
Truth serum	54	6	90.0%
Random VCG	32	28	53.3%

profiles. Figure 4.6(a) compares the truthful task ratio among the three mechanisms. We find that our mechanism and the random VCG have a similar performance in truthful profile elicitation, which outperforms truth serum. This is because the former two apply random VCG for worker selection where profile truthfulness is guaranteed while truth serum does not consider profile truthfulness but merely answer truthfulness. A similar observation is obtained in Figure 4.6(b).

We further examine the platform's expense incurred by the three mechanisms for each task in Figure 4.7. The expense is the sum of all worker payments. Truth serum causes the highest expense among the three. Since it fails to consider profile truthfulness, workers can manipulate reported profiles and thus incur extra payment to the platform. We also observe that our mechanism brings a slightly higher expense than random VCG on average. This extra expense ensures the answer truthfulness that random VCG fails to achieve.

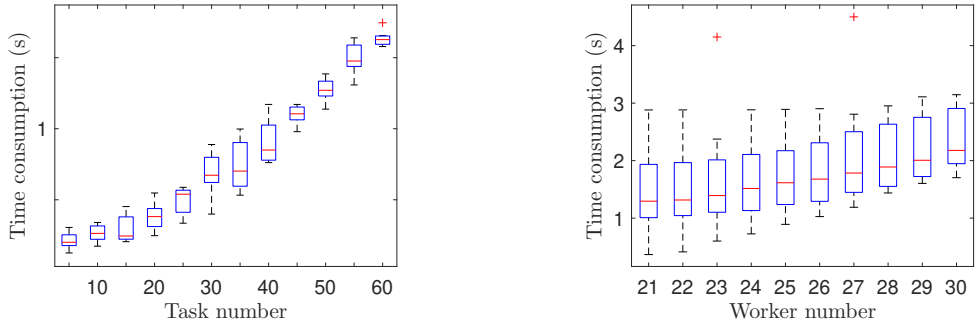
4.5.4 Accuracy Performance

Once the platform collects answers from workers, it aggregates them and derives the final result. We employ the majority voting as the aggregation method, i.e., if more than half answers are "yes", then the final result of the task is deemed "yes"; otherwise, it is "no". Accuracy is different from answer truthfulness. The former denotes that the aggregated answer is the same as the ground truth, while the latter means a worker reports her true observation. Table 4.2 compares the accuracy performance among the three mechanisms. Our mechanism and truth serum have a similar performance. Specifically, we get correct answers in 55 and 54 tasks out of 60 via our mechanism and truth serum, respectively. The



(a) (b)

Figure 4.8: The platform's expense under different crowdsourcing sizes.



(a) (b)

Figure 4.9: Time consumption of our mechanism under different system sizes.

random VCG has the lowest accuracy ratio, 53.3%. Together with the answer truthfulness analysis Figure 4.4, we find that answer truthfulness and accuracy demonstrate a strong positive correlation in our mechanism. This property is useful, especially for the tasks whose ground truth is hard to derive.

4.5.5 Impact of System Size

We also analyze the impact of system size on the mechanism performance. Figure 4.8 shows the platform's expense under different system sizes. We find in Figure 4.8(a) that the expense increases linearly as the number of tasks grows. Specifically, the total amount is \$26.8 when there are 30 tasks and 15 reference answers. This value increases to \$60.5 when the task number becomes 60. The observation meets our expectation: more workers need to be recruited to execute more tasks, thus incurring higher expenses to the platform. Here, reference answers correspond to r_{-i} in the mechanism. Figure 4.8(b) shows that the platform's average expense also increases linearly with respect to the number of reference answers. Specifically, the platform's average payment is \$25.5 when there are 40 tasks and 10 reference answers. It increases to \$50.3 when the answer number becomes 15.

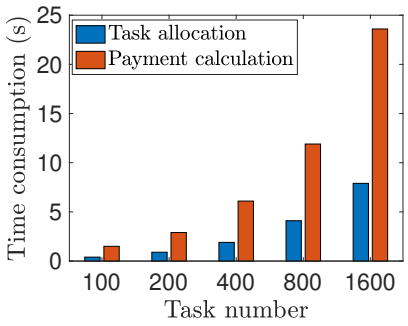
Figure 4.9 illustrates the time consumption of our mechanism under different settings. It mainly comes from three processes: winner selection, payment calculation, and the data communication between the platform program and the worker-side app. Notice that the task execution time, i.e., the duration for workers to conduct tasks and derive results, is not included, as this part depends on individual intelligence that varies from worker to worker. We observe from Figure 4.9(a) that the time slightly increases, from about 0.7s to 2.2s, when the task number changes from 5 to 60. The average time consumption for each task is as low as 0.05s. It is worth mentioning that tasks are conducted sequentially in the current experiment. The time consumption can be further reduced when they are processed in parallel, which we will implement and examine in our future work. A similar trend is observed in Figure 4.9(b). The time consumption slightly increases when more workers participate. Specifically, the value is 1.1s when there are 21 workers, and it becomes 2.1s when the worker size is 60.

To sum up, our mechanism can not only effectively elicit truthful answers and profiles, but is also feasible to implement for practical crowdsourcing systems due to its moderate expense caused to the platform and high computation efficiency.

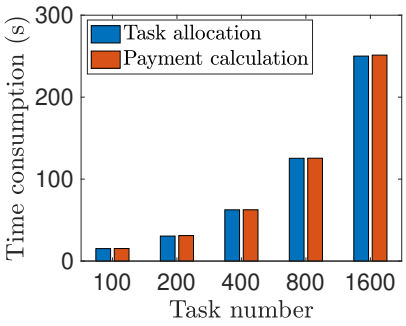
4.5.6 Analysis of Scalability

To evaluate the scalability of the proposed scheme, we further conduct a series of simulations. Besides, its performance is also compared with a baseline approach, i.e., the conventional VCG. To implement the baseline approach, we utilize CPLEX, a commercial optimization software package [68], to optimally solve the integer programming (IP) problem P_1 for task allocation and VCG for payment determination. For each task and worker, we randomly generate task-independent parameters d_i and workers' profile b_i from the normal distribution $N(0.5, 0.2)$. γ is set to 2 by default. The maximum worker size and task size is set to 1000 and 1600, respectively. All results are averaged over 100 trials. The evaluations run on a Dell laptop with a 1.6GHz processor and 16GB RAM.

Figure 4.10 compares the time consumption between our scheme and the baseline approach given different amounts of tasks. As a note, the baseline approach stands for the conventional VCG approach that directly applies CPLEX to find the optimum solution of task allocation and pricing. We observe in Figure 4.10 that our scheme spends much less time than the baseline approach given the same number of tasks. For example, when there are 400 tasks, it takes our scheme 2.3 s and 6.9 s to derive task allocation and pricing outcomes, respectively. However, the values become 62.5 s and 63.2 s for the baseline approach. This is because the latter solves computationally expensive IP problems, i.e., P_1, for both task allocation and payment determination; instead, our scheme follows the

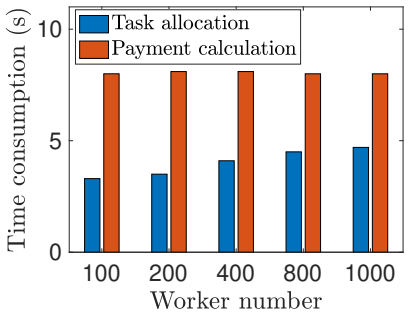


(a) Our scheme

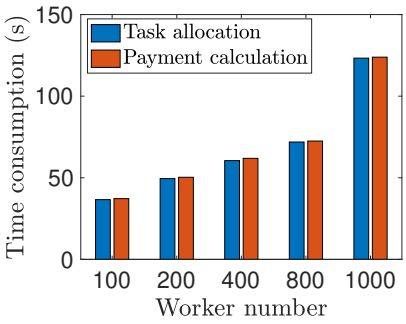


(b) The baseline approach

Figure 4.10: Time consumption of our scheme and the baseline approach over different task numbers (Worker number=500).



(a) Our scheme



(b) The baseline approach

Figure 4.11: Time consumption of our scheme and the baseline approach over different worker numbers (Task number=500).

framework of fractional VCG that only involves solving LP problems with polynomial complexity.

Figure 4.11 compares the time consumption between our scheme and the baseline approach given different numbers of workers. We observe that our scheme's time for task allocation increases slower than that of the baseline approach as the worker number grows. This is because more variables introduce a lower time complexity to LP problems (or our scheme) than IP problems (or the baseline approach). We also notice in Figure 4.11(a) that our payment determination takes relatively stable time with the increase of worker numbers. Combining the results from Figure 4.10 and Figure 4.11, we conclude that our scheme is more practical for implementation than the conventional VCG in terms of computation efficiency, especially when the number of workers and/or tasks is large.

4.6 Related Work

Answer truthfulness. Mechanism design to elicit truthful answers/data, in binary-answer tasks, is an extensively studied topic [104, 106, 105, 108, 109, 110, 111, 108] The idea is to devise payment rules such that truth-telling is a Nash equilibrium. Since the ground truth for each task is unknown to the system, a natural solution is to reward workers based on other workers' reports, i.e., reference answers [104, 106, 105] Another solution is to utilize a truth detection technology that gives a signal indicating if a worker is truthful or lying based on factors, e.g., physiological measures (e.g., pupil dilation) [108, 109]. Realizing that workers' efforts also determine the accuracy of crowdsourcing services, a couple of works [110, 111, 108] further utilize incentives to motivate workers to exert efforts in task execution. For example, Dasgupta et al. [110] incentivized maximum effort followed by truthful reports of answers in an equilibrium that achieves maximum payoffs for workers. Gong et al. [111] developed mechanisms to incentivize strategic workers to truthfully reveal their private quality and data, and make truthful efforts as desired by the crowdsourcing requester. The above two works assume that workers have the same cost of effort exertion and/or the same solution accuracy. In practice, for example, students with a better academic background are likely better at homework assessment. They may spend smaller costs (e.g., time) generating homework assessments with higher quality (i.e., higher solution accuracy). Huang et al. [108] then accommodated such worker heterogeneity in their incentive mechanism design. None of the above works considers profile truthfulness. As discussed, this property is of equal importance for the platform to deliver high-quality crowdsourcing services.

Cost truthfulness. Minimizing the overall worker cost, in terms of energy consumption, travel distance, or computing resources, is a prevailing decision criterion to generate suitable worker-task pairs during task assignments. Since cost is private information and workers are strategic in reporting this value for favorable outcomes, the main challenge lies in how to stimulate workers to disclose their costs truthfully. Incentive mechanisms have attracted most attention for truthful cost elicitation due to their ability to deal with workers' strategic behaviors [134, 136, 137, 138, 60, 139, 157, 140] For example, Yang et al. [134] were among the first to discuss cost truthfulness during task assignment in crowdsourcing. Auction-based incentive mechanisms have been developed. Along this line of research, [137, 139, 157] then study the impact of budget constraints at the platform. Since the monetary provision is limited, the platform's strategy space is thus confined. Noticing that existing mechanisms assume the existence of only one task requester, [140] considers multiple requesters. The framework of double-auction is thus applied. Cost truthfulness is guaranteed at both requesters and workers. Since cost is merely one kind of self-reported profile, cost truthfulness is thus a special case of profile truthfulness that we aim to achieve.

Aside from the cost, strategic workers are able to manipulate a much wider spectrum of self-reported profiles, which conventional cost truthfulness schemes cannot resist.

Summary. To our knowledge, this is the first study that protects two different stages, i.e., task assignment and answer aggregation, in crowdsourcing from workers' strategic misreporting simultaneously. In the task assignment stage, workers report their profiles, such as locations, expertise, and cost of task execution, to the platform, who then decides task assignment based on the collected profiles. Hence, strategic workers may manipulate their reports to gain benefit. This is the same case in the answer aggregation stage where workers lie about their reported answers. Prior works utilize incentive design to tackle either one of the above two kinds of misbehaviors. Instead, we aim to develop a unified framework to address them at the same time. It is infeasible to directly apply existing schemes in each stage, i.e., a worker is first paid for answer truthfulness and then paid for cost truthfulness. This is because the worker's total payment received from both stages, if not carefully calibrated jointly, would violate conditions for both profile truthfulness and answer truthfulness.

Another limitation of the prior works is that they mostly focus on cost truthfulness, i.e., motivating workers to reveal their genuine costs in task execution. In fact, in the stage of task assignment, workers are required to report many other information, such as location and expertise, in addition to the cost. We call them "profile" in this paper. Cost truthfulness cannot guarantee workers truth-telling over other information. To address this issue, our approach achieves profile truthfulness that covers a much wider spectrum of strategic behaviors.

CHAPTER 5

CONCLUSIONS

This dissertation secures crowdsourcing from workers' strategic misreporting behaviors. All schemes are developed under the framework of incentive mechanism design. Our core idea is to pay strategic workers properly such that their best strategies are to behave honestly or in a way that is desired.

In Chapter 2, to resist parameter manipulation performed by individual workers, we leverage the incentive mechanism to develop an effective and novel defense scheme. Our idea is to stimulate workers to report desirable indicators, such that the new formulated problem shares the identical optimal solution with the original one. To implement the defense scheme, we further develop an iterative algorithm. The backtracking based approach is adopted to accelerate the convergence speed. We then formally prove its convergence property and individual rationality. Extensive simulation results show that the algorithm only takes a few iterations to converge, and thus to find the accurate solution under moderate system sizes.

In Chapter 3, we propose a (t, p)-collusion resistant scheme that resists parameter manipulation carried by worker coalitions. No coalition of weighted cardinality t can improve its group utility by coordinating the bids at a probability p. In addition, some desirable economic properties, including p-truthfulness and p-individual rationality, are also guaranteed via our scheme. Since the existence of these properties is in the trade of extra cost at the platform, we also provide a formal analysis of the tradeoff. Simulation results demonstrate the effectiveness of our scheme.

In Chapter 4, we develop an incentive mechanism to jointly elicit truthful answers and parameters from strategic workers in crowdsourcing. Our design first derives the sufficient and necessary conditions for these two goals separately. We then formulate a worker selection optimization problem. Due to its NP-hardness, we resort to solving its relaxed version in a fractional domain. The factional optimal solution is then decomposed into a randomized format. An α-approximate algorithm is further developed. The upper bound of its integrality gap then serves as a scaling factor 2, which is applied to the randomized worker selection to ensure its feasibility. As a final step, the conditions for answer and parameter truthfulness are integrated as constraints of the payment optimization problem, whose solution is the incentive paid to each worker to motivate honest behaviors. As a proof-of-concept implementation, we prototype the proposed mechanism. A series of

experiments that involve 30 volunteers have been conducted. Results show that our mechanism is effective and efficient for practical implementation.

REFERENCES

[1] M. Karaliopoulos, et al., "User recruitment for mobile crowdsensing over opportunistic networks," in *Proceedings of IEEE INFOCOM*, 2015.

[2] Z. Duan, et al., "Distributed auctions for task assignment and scheduling in mobile crowdsensing systems," in *Proceedings of IEEE ICDCS*, 2017.

[3] L. Pournajaf, et al., "Spatial task assignment for crowd sensing with cloaked locations," in *Proceedings of IEEE MDM*, July 2014.

[4] K. L. Huang, et al., "Are you contributing trustworthy data?: The case for a reputation system in participatory sensing," in *Proceedings of the ACM MSWiM*, October 2010.

[5] P. Gilbert, et al., "Toward trustworthy mobile sensing," in *Proceedings of the HotMobile*, February 2010.

[6] K. L. Huang, et al., "A privacy-preserving reputation system for participatory sensing," in *Proceedings of IEEE LCN*, October 2012.

[7] D. Christin, et al., "Incognisense: An anonymity-preserving reputation framework for participatory sensing applications," in *Proceedings of IEEE PerCom*, March 2012.

[8] D. He, et al., "User privacy and data trustworthiness in mobile crowd sensing," *IEEE Wireless Communications*, vol. 22, no. 1, pp. 28–34, February 2015.

[9] A. Dua, et al., "Towards trustworthy participatory sensing," in *Proceedings of the USENIX HotSec*, 2009.

[10] Tschantz, Michael Carl and Sen, Shayak and Datta, Anupam. SoK: Differential Privacy as a Causal Property. In *IEEE Symposium on Security and Privacy*, 2020.

[11] Kifer, Daniel and Machanavajjhala, Ashwin. No free lunch in data privacy. In *Proceedings of the ACM International Conference on Management of Data*, 2011.

[12] W. Jin, et al., "DPDA: A Differentially Private Double Auction Scheme for Mobile Crowd Sensing," in *Proceedings of IEEE CNS*, 2018.

[13] W. Jin, et al., "If You Do Not Care About It, Sell It: Trading Location Privacy in Mobile Crowd Sensing," in *Proceedings of IEEE INFOCOM*, 2019.

[14] H. Xie, et al., "Incentive mechanism and protocol design for crowdsourcing systems," in *Proceedings of Allerton*, September 2014.

[15] H. Jin, et al., "Quality of information aware incentive mechanisms for mobile crowd sensing systems," in *Proceedings of the ACM MobiHoc*, June 2015.

[16] Z. Feng, et al., "Trac: Truthful auction for location-aware collaborative sensing in mobile crowdsourcing," in *Proceedings of IEEE INFOCOM*, April 2014.

[17] P. Micholia, et al., "Mobile crowdsensing incentives under participation uncertainty," in *Proceedings of the ACM Workshop on MSCC*, July 2016.

[18] J. Jormakka and J. Molsa, "Modelling information warfare as a game," *Journal of information warfare*, vol. 4, no. 2, pp. 12–25, 2005.

[19] L. Jiang, et al., "How bad are selfish investments in network security?" *IEEE/ACM Transactions on Networking*, vol. 19, no. 2, pp. 549–560, 2011.

[20] D. P. Bertsekas, *Nonlinear programming*. Belmont, MA, USA: Athena Scientific, 1999.

[21] G. Iosifidis, et al., "A double-auction mechanism for mobile data-offloading markets," *IEEE/ACM Transactions on Networking*, vol. 23, no. 5, pp. 1634–1647, October 2015.

[22] C. Liu, et al., "Network optimization and control," *Foundations and Trends in Networking*, vol. 2, no. 3, pp. 271–379, 2007.

[23] S. Boyd and L. Vandenberghe, *Convex optimization*. Stanford, CA, USA: Cambridge university press, 2004.

[24] Yelp dataset challenge. [Online]. Available: https://www.yelp.com/dataset/challenge

[25] H. To, et al., "A server-assigned spatial crowdsourcing framework," *ACM Transactions on Spatial Algorithms and Systems*, vol. 1, no. 1, p. 2, 2015.

[26] B. Liu, et al., "Protecting location privacy in spatial crowdsourcing using encrypted data," in *Proceedings of EDBT*, 2017.

[27] C. Liu, et al., "Mechanism design games for thwarting malicious behavior in crowdsourcing application," in *Proceedings of IEEE INFOCOM*, 2017.

[28] [Online]. Available: https://www.dropbox.com/s/mqnsutw7q6e47m2/TechnicalReport.pdf?dl=0

[29] OpenStreeMap [Online]. Available: https://www.openstreetmap.org/#map=4/38.01/-95.84

[30] P. Smith, A. Luong, S. Sarkar, H. Singh, A. Singh, N. Patwari, S. K. Kasera, and K. Derr, "A novel software defined radio for practical, mobile crowd-sourced spectrum sensing," *IEEE Transactions on Mobile Computing*, 2021.

[31] Y. Hu and R. Zhang, "A spatiotemporal approach for secure crowdsourced radio environment map construction," *IEEE/ACM Transactions on Networking*, vol. 28, no. 4, pp. 1790–1803, Aug 2020.

[32] X. Li, and Q. Zhu, " Social incentive mechanism based multi-user sensing time optimization in co-operative spectrum sensing with mobile crowd sensing," *Sensors*, vol. 18, no. 1, pp. 250–271, 2018.

[33] A. Chakraborty, M.S. Rahman, H. Gupta, and S.R. Das, "Specsense: Crowdsensing for efficient querying of spectrum occupancy," in *Proceedings of the IEEE INFOCOM*, 2017.

[34] Y. Tong, J. She, B. Ding, L. Wang, and L. Chen, "Online mobile micro-task allocation in spatial crowdsourcing," in *Proceedings of the IEEE ICDE*, 2016.

[35] Z. Chen, R. Fu, Z. Zhao, Z. Liu, L. Xia, L. Chen, P. Cheng, C.C. Cao, Y. Tong, and C.J. Zhang, "gmission: A general spatial crowdsourcing platform," in *Proceedings of the VLDB*, 2014.

[36] L. Wang, Z. Yu, Q. Han, B. Guo, and H. Xiong, "Multi-objective optimization based allocation of heterogeneous spatial crowdsourcing tasks," *IEEE Transactions on Mobile Computing*, vol. 17, no. 7, pp. 1637–1650, Jul 2017.

[37] P. Cheng, X. Lian, L. Chen, and C. Shahabi, "Prediction-based task assignment in spatial crowdsourcing," in *Proceedings of the IEEE ICDE*, 2017.

[38] P. Cheng, X. Jian, and L. Chen, "An experimental evaluation of task assignment in spatial crowdsourcing," in *Proceedings of the VLDB*, 2018.

[39] S. He, and S-H.G. Chan, "Wi-Fi fingerprint-based indoor positioning: Recent advances and comparisons," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 1, pp. 466–490, Aug 2015.

[40] X. Teng, D. Guo, Y. Guo, X. Zhou, Z. Ding, and Z. Liu, "IONavi: An indoor-outdoor navigation service via mobile crowdsensing," *ACM Transactions on Sensor Networks* , vol. 13, no. 2, pp. 1–28, Apr 2017.

[41] Z. Li, X. Zhao, F. Hu, Z. Zhao, J.L.C. Villacrés, and T. Braun, "SoiCP: A Seamless Outdoor–Indoor Crowdsensing Positioning System," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 8626–8644, Jun 2019.

[42] C-L. Leca, I. Nicolaescu, and P. Ciotirnae, "Crowdsensing Influences and Error Sources in Urban Outdoor Wi-Fi Fingerprinting Positioning," *Sensors*, vol. 20, no. 2, pp. 427, Jan 2020.

[43] "Clickworker," https://www.clickworker.com.

[44] S. Li, Z. Qin, H. Song, C. Si, B. Sun, X. Yang, and R. Zhang, "A lightweight and aggregated system for indoor/outdoor detection using smart devices," *Future Generation Computer Systems*, vol. 107, pp. 988–997, June 2020.

[45] J. Wang, N. Tan, J. Luo, and S.J. Pan, "WOLoc: WiFi-only outdoor localization using crowdsensed hotspot labels," in *Proceedings of IEEE INFOCOM*, 2017.

[46] C. Zhang, J. Luo, and J. Wu, "A dual-sensor enabled indoor localization system with crowdsensing spot survey," in *Proceedings of IEEE DCOSS*, 2017.

[47] K. Schurter, "Identification and inference in first-price auctions with collusion," *Working Paper: University of Chicago*, 2017.

[48] S.S. Padhi, and P.J. Mohapatra, "Detection of collusion in government procurement auctions," *Journal of Purchasing and Supply Management*, vol. 17, no. 4, pp. 207–221, Dec 2011.

[49] Y. Bolotova, and J.M. Connor, and D.J. Miller, "The impact of collusion on pricebehavior: empirical results from two recent cases," *International Journal of Industrial Organization*, vol. 26, pp. 1290–1307, Nov 2008.

[50] Howe, Jeff, "Crowdsourcing: How the power of the crowd is driving the future of business," *Wired Magazine*, 2008.

[51] Chang, Joseph Chee and Amershi, Saleema and Kamar, Ece, "Revolt: Collaborative crowdsourcing for labeling machine learning datasets," *Proceedings of the CHI Conference on Human Factors in Computing Systems*, 2017.

[52] Bachrach, Yoram and Kash, Ian A and Key, Peter and Oren, Joel, "Strategic behavior and learning in all-pay auctions: an empirical study using crowdsourced data," *Autonomous Agents and Multi-Agent Systems*, vol. 33, pp. 192–215, Nov 2019.

[53] J.E. Harrington Jr, and J. Chen, "Cartel pricing dynamics with cost variability and endogenous buyer detection," *International Journal of Industrial Organization*, vol. 24, pp. 1185–1212, Nov 2006.

[54] I. Morozov, and E.A. Podkolzina, "Collusion detection in procurement auctions," *National Research University Higher School of Economics*, No. WP BPR, vol. 25, Mar 2013.

[55] S. Lundberg, "Restrictions on competition in municipal competitive procurement in Sweden," *International Advances in Economic Research*, vol. 11, pp. 329–342, Mar 2005.

[56] J.H. Kagel, *Auctions: a survey of experimental research*, University of Pittsburgh, 1990.

[57] P. Bajari, and L. Ye, "Deciding between competition and collusion," *The Review of Economics and Statistics*, vol. 85, no. 4, pp. 971–989, Nov 2003.

[58] Z. Duan, W. Li, and Z. Cai, "Distributed auctions for task assignment and scheduling in mobile crowdsensing systems," in *Proceedings of the IEEE ICDCS*, 2017.

[59] H. Jin, L. Su, D. Chen, K. Nahrstedt, and J. Xu, "Quality of information aware incentive mechanisms for mobile crowd sensing systems," in *Proceedings of the ACM MobiHoc*, 2015.

[60] M. Xiao, M. Li, L. Guo, M. Pan, Z. Han, and P. Li, "Securing task allocation in mobile crowd sensing: An incentive design approach," in *Proceedings of IEEE CNS*, 2019.

[61] H. Jin, L. Su, D. Chen, K. Nahrstedt, and J. Xu, "Inception: Incentivizing privacy-preserving data aggregation for mobile crowd sensing systems," in *Proceedings of the ACM MobiHoc*, 2016.

[62] S. He, D-H. Shin, J. Zhang, and J. Chen, "Toward optimal allocation of location dependent tasks in crowdsensing," in *Proceedings of the IEEE INFOCOM*, 2014.

[63] R. Jurca, and B. Faltings, "Minimum payments that reward honest reputation feedback," in *Proceedings of the ACM EC*, 2006.

[64] W. Shi, L. Zhang, C. Wu, Z. Li, and F. C. Lau, "An online auction framework for dynamic resource provisioning in cloud computing," in *Proceedings of the ACM SIGMETRICS*, 2014.

[65] D. A. Spielman and S.-H. Teng, "Smoothed analysis of algorithms: Why the simplex algorithm usually takes polynomial time," *Journal of the ACM*, vol. 51, no. 3, pp. 385–463, May 2004.

[66] "Http file server," http://www.rejetto.com/hfs/.

[67] "Crowdsourcing datasets," http://dbgroup.cs.tsinghua.edu.cn/ligl/crowddata/.

[68] "Cplex," https://www.ibm.com/docs/en/icos/12.8.0.0?topic=programming-where-find-cplex-examples.

[69] J. Hu, H. Lin, X. Guo, and J. Yang, "Dtcs: An integrated strategy for enhancing data trustworthiness in mobile crowdsourcing," *IEEE Internet of Things Journal*, vol. 5, no. 6, pp. 4663–4671, Feb 2018.

[70] J. Lin, D. Yang, M. Li, J. Xu, and G. Xue, "Frameworks for privacy-preserving mobile crowdsensing incentive mechanisms," *IEEE Transactions on Mobile Computing*, vol. 17, no. 8, pp. 1851–1864, Dec 2017.

[71] Z. Feng, Y. Zhu, Q. Zhang, L.M. Ni, and A.V. Vasilakos, "Trac: Truthful auction for location-aware collaborative sensing in mobile crowdsourcing," in *Proceedings of the IEEE INFOCOM*, 2014.

[72] X. Zhang, Z. Yang, Z. Zhou, H. Cai, L. Chen, and X. Li, "Free market of crowdsourcing: Incentive mechanism design for mobile sensing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 12, pp. 3190–3200, Dec 2014.

[73] M.S. Robinson, "Collusion and the choice of auction," *The RAND Journal of Economics*, vol. 16, no. 1, pp. 141–145, Apr 1985.

[74] P. Cramton, and J.A. Schwartz, "Collusive bidding: Lessons from the fcc spectrum auctions," *Journal of regulatory Economics*, vol. 17, no. 3, pp. 229–252, May 2000.

[75] P. Cramton, "The fcc spectrum auctions: An early assessment," *Journal of Economics & Management Strategy*, vol. 6, no. 3, pp. 431–495, Fall 1997.

[76] P. Bajari, and J.T. Fox, "Complementarities and collusion in an FCC spectrum auction," *Working paper: University of Chicage*, 2009.

[77] P. Bajari, and J. Yeo, "Auction design and tacit collusion in fcc spectrum auctions," *Information Economics and Policy*, vol. 21, no. 2, pp. 90–100, Jun 2009.

[78] M. Friedman, "Comment on collusion in the auction market for treasury bills," *Journal of Political Economy*, vol. 72, no. 5, pp. 513–514, Oct 1964.

[79] G. Goswami, T.H. Noe, and M.J. Rebello, "Collusion in uniform-price auctions: Experimental evidence and implications for treasury auctions," *The Review of Financial Studies*, vol. 9, no. 3, pp. 757–785, Autumn 1996.

[80] Mailath, George J, "Do people play Nash equilibrium? Lessons from evolutionary game theory," *Journal of Economic Literature*, vol. 36, no. 3, pp. 1347–1374, Sep 1998.

[81] Michael Hay, Chao Li, Gerome Miklau, and David Jensen. Accurate estimation of the degree distribution of private networks. In *Proceedings of the International Conference on Data Mining*, 2009.

[82] Yue Wang, Xintao Wu, and Leting Wu. Differential privacy preserving spectral graph analysis. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, 2013.

[83] Guru. [Online]. Available: https://www.guru.com, 2021.

[84] Waze. [Online]. Available: https://www.waze.com/, 2021.

[85] Taskrabbit. [Online]. Available: https://www.taskrabbit.com, 2021.

[86] GrubHub. [Online]. Available: https://www.grubhub.com/, 2021.

[87] Attn ebay users: Beware of 'shill bidders' driving up prices. [Online]. Available: https://www.cio.com/article/2369962/internet/attn-ebay-users--beware-of--shill-bidders--driving-up-prices.html, 2021.

[88] Shill bidding policy. [Online]. Available: https://www.ebay.com/help/policies/selling-policies/selling-practices-policy/shill-bidding-policy?id=4353, 2021.

[89] Uber. [Online]. Available: https://www.uber.com/, 2021.

[90] A.E. Roth, and A. Ockenfels, "Last-minute bidding and the rules for ending second-price auctions: Evidence from ebay and amazon auctions on the internet," *American Economic Review*, vol. 92, no. 4, pp. 1093–1103, Sep 2002.

[91] Y. Liu, Y. Yang, and Y.L. Sun, "Detection of collusion behaviors in online reputation systems," in *Proceedings of the ACSSC*, 2008.

[92] KhudaBukhsh, Ashiqur R and Carbonell, Jaime G and Jansen, Peter J,"Detecting non-adversarial collusion in crowdsourcing" in *Proceedings of the AAAI*, 2014.

[93] G. Ciccarelli, and R.L. Cigno, "Collusion in peer-to-peer systems," *Computer Networks*, vol. 55, no. 15, pp. 3517–3532, Oct 2011.

[94] Q. Lian, and Z. Zhang, M. Yang, B.Y. Zhao, Y. Dai, and X. Li, "An empirical study of collusion behavior in the maze p2p file-sharing system," in *Proceedings of the IEEE ICDCS*, 2007.

[95] S. Ji, and T. Chen, "On designing collusion-resistant incentive mechanisms for mobile crowdsensing systems," in *Proceedings of the IEEE Trustcom/BigDataSE/ICESS*, 2017.

[96] A.V. Goldberg, and J.D. Hartline, "Collusion-resistant mechanisms for single-parameter agents," in *Proceedings of the SODA*, 2005.

[97] Y-K. Che, and J. Kim, "Optimal collusion-proof auctions," *Journal of Economic Theory*, vol. 144, no. 2, pp. 565–603, Mar 2009.

[98] ——, "Robustly collusion-proof implementation," *Econometrica*, vol. 74, no. 4, pp. 1063–1107, Jul 2006.

[99] P. Penna, and C. Ventre, "Optimal collusion-resistant mechanisms with verification," in *Proceedings of the ACM EC*, 2009.

[100] S. Li, "Obviously strategy-proof mechanisms," *American Economic Review*, vol. 107, no. 11, pp. 3257–3287, Nov 2017.

[101] A.V. Goldberg, and J.D. Hartline, "Competitiveness via consensus," in *Proceedings of the SODA*, 2003.

[102] M. Niazi Torshiz, and H. Amintoosi, "Collusion-resistant worker selection in social crowdsensing systems," *Computer and Knowledge Engineering*, vol. 1, no. 1, pp. 9–20, Jan 2017.

[103] Zhang, Xinglin and Yang, Zheng and Liu, Yunhao and Tang, Shaohua, "On reliable task assignment for spatial crowdsourcing," *IEEE Transactions on Emerging Topics in Computing*, vol. 7, no. 1, pp. 174–186, Jan 2016.

[104] Huang, Chao and Yu, Haoran and Huang, Jianwei and Berry, Randall, "Eliciting Information from Heterogeneous Mobile Crowdsourced Workers Without Verification," *IEEE Transactions on Mobile Computing*, Early Access, Jan 2021.

[105] Kong, Yuqing and Schoenebeck, Grant and Tao, Biaoshuai and Yu, Fang-Yi, "Information elicitation mechanisms for statistical estimation," in *Proceedings of AAAI*, 2020.

[106] Kong, Yuqing, "Dominantly truthful multi-task peer prediction with a constant number of tasks," in *Proceedings of the SODA*, 2020.

[107] Wang, Dong and Abdelzaher, Tarek and Kaplan, Lance and Aggarwal, Charu C, "Recursive fact-finding: A streaming approach to truth estimation in crowdsourcing applications," in *Proceedings of the ICDCS*, 2013.

[108] Huang, Chao and Yu, Haoran and Berry, Randall A and Huang, Jianwei, "Using Truth Detection to Incentivize Workers in Mobile Crowdsourcing," in *IEEE Transactions on Mobile Computing*, early access, 2020.

[109] ——, "Using truth detection to incentivize workers in mobile crowdsourcing," *IEEE Transactions on Mobile Computing*, pp. 1–1, 2020.

[110] A. Dasgupta and A. Ghosh, "Crowdsourced judgement elicitation with endogenous proficiency," in *Proceedings of the WWW*, 2013.

[111] X. Gong and N. Shroff, "Incentivizing truthful data quality for quality-aware mobile data crowdsourcing," in *Proceedings of the ACM MobiHoc*, 2018.

[112] J. Schummer, "Manipulation through bribes," *Journal of Economic Theory*, vol. 91, no. 2, pp. 180–198, Apr 2000.

[113] R.C. Marshall, and L.M. Marx, "Bidder collusion," *Journal of Economic Theory*, vol. 133, no. 1, pp. 374–402, Mar 2007.

[114] D.A. Graham, and R.C. Marshall, "Collusive bidder behavior at single-object second-price and english auctions," *Journal of Political economy*, vol. 95, no. 6, pp. 1217–1239, Dec 1987.

[115] O. Lev, M. Polukarov, Y. Bachrach, and J.S. Rosenschein, " Mergers and collusion in all-pay auctions and crowdsourcing contest," in *Proceedings of the AAMAS*, 2013

[116] J. Silvertown, M. Harvey, R. Greenwood, M. Dodd, J. Rosewell, T. Rebelo, J. Ansine, and K. McConway, "Crowdsourcing the identification of organisms: A case-study of iSpot," *ZooKeys*, vol. 5, no. 480, pp. 125–146, Feb 2015.

[117] iSpot. [Online]. Available: https://www.ispotnature.org, 2021.

[118] N.R. Devanur, and S.M. Kakade, "The price of truthfulness for pay-per-click auctions" in *Proceedings of ACM EC*, 2009.

[119] A. Ronen, "On approximating optimal auctions" in *Proceedings of ACM EC*, 2001.

[120] A. Doan, M.J. Franklin, D. Kossmann, and T. Kraska, "Crowdsourcing applications and platforms: A data management perspective" in *Proceedings of VLDB*, 2011.

[121] W. Wang, Z. He, P. Shi, W. Wu, Y. Jiang, B. An, Z. Hao, and B. Chen, "Strategic social team crowdsourcing: Forming a team of truthful workers for crowdsourcing in social networks" *IEEE Transactions on Mobile Computing*, vol. 18, no. 6, pp. 1419–1432, Jun 2018.

[122] R.P. McAfee, and J. McMillan, "Bidding rings," *The American Economic Review*, vol. 82, no. 3, pp. 579–599, Jun 1992.

[123] S. Athey and K. Bagwell, "Optimal collusion with private information," *RAND Journal of Economics*, vol. 32, no. 3, pp. 428–465, Oct 2001.

[124] A. Abdulkadiroglu, and K-S. Chung, "Auction design with tacit collusion," *Working Paper: Columbia and Northwestern University*, 2003.

[125] Z. Ji, and K.R. Liu, "Multi-stage pricing game for collusion-resistant dynamic spectrum allocation," *IEEE Journal on Selected Areas in Communications*, vol. 26, no. 1, pp. 182–191, Jan 2008.

[126] Y. Wu, B. Wang, K.R. Liu, and T.C. Clancy, "A scalable collusion-resistant multi-winner cognitive spectrum auction game," *IEEE Transactions on Communications*, vol. 57, no. 12, pp. 3805–3816, Dec 2009.

[127] ——, "Collusion-resistant multi-winner spectrum auction for cognitive radio networks," in *Proceedings of the IEEE GLOBECOM*, 2008.

[128] X. Zhou, and H. Zheng, "Breaking bidder collusion in large-scale spectrum auctions," in *Proceedings of the ACM MobiHoc*, 2010.

[129] Amazon Mechanical Turk. [Online]. Available: https://www.mturk.com, 2021.

[130] T. Roughgarden, "Algorithmic game theory," *Communications of the ACM*, vol. 53, no. 7, pp. 78–86, Jul 2010.

[131] S.D. Jap, "Online reverse auctions: Issues, themes, and prospects for the future," *Journal of the Academy of Marketing Science*, vol. 30, no. 4, pp. 506–525, Oct 2002.

[132] S.D. Jap, "An exploratory study of the introduction of online reverse auctions," *Journal of Marketing*, vol. 67, no. 3, pp. 96–107, Jul 2003.

[133] B. Satzger, H. Psaier, D. Schall, and S. Dustdar, "Auction-based crowdsourcing supporting skill management," *Information Systems*, vol. 38, no. 4, pp. 547–560, Jun 2013.

[134] D. Yang, G. Xue, X. Fang, and J. Tang, "Crowdsourcing to smartphones: incentive mechanism design for mobile phone sensing," in *Proceedings of the ACM MobiHoc*, 2012.

[135] U. Gadiraju, R. Kawase, S. Dietze, and G. Demartini, "Understanding malicious behavior in crowdsourcing platforms: The case of online surveys," in *Proceedings of the ACM CHI*, 2015.

[136] W. Liu, Y. Yang, E. Wang, and J. Wu, "Dynamic user recruitment with truthful pricing for mobile crowdsensing," in *Proceedings of IEEE INFOCOM*, 2020.

[137] W. Jin, M. Xiao, L. Guo, L. Yang, and M. Li, "Ulpt: A user-centric location privacy trading framework for mobile crowd sensing," *IEEE Transactions on Mobile Computing*, 2021.

[138] M. Xiao, W. Jin, M. Li, L. Yang, A. Thapa, and P. Li, "Collusion-resistant worker recruitment in crowdsourcing systems," *IEEE Transactions on Mobile Computing*, 2021.

[139] W. Jin, M. Xiao, M. Li, and L. Guo, "If you do not care about it, sell it: Trading location privacy in mobile crowd sensing," in *Proceedings of IEEE INFOCOM*, 2019.

[140] Y. Wei, Y. Zhu, H. Zhu, Q. Zhang, and G. Xue, "Truthful online double auctions for dynamic mobile crowdsourcing," in *Proceedings of the IEEE INFOCOM*, 2015.

[141] V. Conitzer and T. Sandholm, "Complexity results about nash equilibria," in *Proceedings of the IJCAI*, 2002.

[142] L. Huang, Y. Zhu, J. Yu, and M-Y. Wu, "Group buying based incentive mechanism for mobile crowd sensing," in *Proceedings of the IEEE SECON*, 2016.

[143] R. Lavi and C. Swamy, "Truthful and near-optimal mechanism design via linear programming," *Journal of the ACM*, vol. 58, no. 6, p. 25, December 2011.

[144] R. Carr and S. Vempala, "Randomized metarounding," in *Proceedings of the ACM STOC*, 2000.

[145] Y. Zheng, G. Li, and R. Cheng, "Docs: a domain-aware crowdsourcing system using knowledge bases," *Proceedings of the VLDB*, vol. 10, no. 4, pp. 361–372, November 2016.

[146] X. Liu, M. Lu, B. C. Ooi, Y. Shen, S. Wu, and M. Zhang, "Cdas: a crowdsourcing data analytics system," *Proceedings of the VLDB Endowment*, vol. 5, no. 10, pp. 1040–1051, June 2012.

[147] H. Kellerer, U. Pferschy, and D. Pisinger, *Knapsack problems*, 2003.

[148] M. Grötschel, L. Lovász, and A. Schrijver, *Geometric algorithms and combinatorial optimization*, 1988.

[149] T. Carnes and D. Shmoys, "Primal-dual schema for capacitated covering problems," in *Proceedings of IPCO*, 2008.

[150] G. Kazai, J. Kamps, and N. Milic-Frayling, "Worker types and personality traits in crowdsourcing relevance labels," in *Proceedings of the ACM CIKM*, 2011.

[151] A.K. Gupta, and S. Nadarajah, *Handbook of beta distribution and its applications*, CRC press, 2004.

[152] Ye, Guanyu and Zhao, Yan and Chen, Xuanhao and Zheng, Kai, "Task Allocation with Geographic Partition in Spatial Crowdsourcing," in *Proceedings of ACM CIKM*, 2021.

[153] Zhao, Yan and Zheng, Kai and Cui, Yue and Su, Han and Zhu, Feida and Zhou, Xiaofang, "Predictive task assignment in spatial crowdsourcing: a data-driven approach," in *Proceedings of IEEE ICDE*, 2020.

[154] Eliminating Spammers and Ranking Annotators for Crowdsourced Labeling Tasks," *The Journal of Machine Learning Research*, vol. 13, no. Feb, pp. 491–518, Feb 2012.

[155] J. Purohit, X. Wang, S. Mao, X. Sun, and C. Yang, "Fingerprinting-based indoor and outdoor localization with lora and deep learning," in *Proceedings of IEEE GLOBECOM*, 2020.

[156] Y. Yang, Y. Ding, D. Yuan, G. Wang, X. Xie, Y. Liu, T. He, and D. Zhang, "Transloc: transparent indoor localization with uncertain human participation for instant delivery," in *Proceedings of MobiCom*, 2020.

[157] Y. Hu and R. Zhang, "Differentially-private incentive mechanism for crowdsourced radio environment map construction," in *Proceedings of IEEE INFOCOM*, 2019.

[158] Incentives for effort in crowdsourcing using the peer truth serum," *ACM Transactions on Intelligent Systems and Technology*, vol. 7, no. 4, pp. 48, Jul 2016.