Collusion-Resistant Worker Recruitment in Crowdsourcing Systems

Mingyan Xiao, *Student Member, IEEE,* Wenqiang Jin, *Student Member, IEEE,* Ming Li, *Member, IEEE,* Lei Yang, *Member, IEEE,* and Pan Li, *Member, IEEE*

Abstract—In the wake of the Web 2.0, crowdsourcing has emerged as a promising approach to maintain a flexible workforce for human intelligence tasks. To stimulate worker participation, many reverse auction-based incentive mechanisms have been proposed. Designing auctions that discourage workers from cheating and instead encouraging 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 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 paper 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 analytically prove our scheme's collusion resistance and then experimentally verify our analytical conclusion using a real-world dataset.

Index Terms—Crowdsourcing, worker recruitment auctions, collusion resistance.

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 [45] and Guru [19].

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* [48], [49], [50] have been extensively adopted, where workers compete with each other by submitting to the platform their bids, i.e., the minimum payment they accept for the task. As proved in these works, such competition can effectively bring down the platform's expense in hiring cheaper larbor and thus significantly enhance economic efficiency.

- M. Xiao, W, Jin and M. Li are with the Department of Computer Science and Engineering, The University of Texas at Arlington, TX, 76019.
- L. Yang is with the Department of Computer Science and Engineering, University of Nevada, Reno, NV, 89557.
- P. Li is with the Department of Electrical Engineering and Computer Science, Case Western Reserve University, Cleveland, OH, 44106.

Despite the appealing properties, auction-based markets are deemed vulnerable to bidder misbehaviors [12]. 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 [1], [2], [3], [4], [8], [9], [11], [51], [52], [53] 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 [13], [14], [15], [16], treasure auctions [17], [18], eBay online auctions [20], [21], [22], and auctions in P2P systems [23], [24], [25]. Empirical analysis on these auctions reveals that most collusion groups are small, less than 6 members per group [16], [22], [25]. 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 crowdsourcing marketplace.

Collusion resistance has rarely been studied in the context of crowdsourcing, except [26] which targets at a particular form of collusion, while we aim to resist a broad set of collusion attacks. In fact, designing collusion-resistant mechanisms is a nontrivial task. According to the *impos-*

sibility results proved in [27], [35], 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 [27], [28], [29], [30]. However, most of them rely on assumptions such as incomplete information sharing among colluders [28], [29] and the auctioneer having prior knowledge over bidder behaviors [30]. Neither of the above assumptions holds in practical crowdsourcing systems. To avoid these constraints, a scheme called APM is proposed by Goldberg and Hartline [27]. 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 of 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 formal analysis over the platform's extra cost caused by trading for collusion avoidance.

The main contribution of this paper 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 [19] is used to evaluate the performances of the proposed scheme.

The rest of this paper is organized as follows. Section 2 presents the problem statement. We describe our basic scheme in Section 3, which serves as the corner stone for our collusion-resistance scheme in Section 4. The performance analysis and simulation results are given in Section 5 and 6, respectively, followed by the related works in Section 7. Section 8 concludes the entire work.

2 Problem Statement

In this section, we first introduce the framework of auction-based worker recruitment in crowdsourcing, and then 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.

2.1 Worker Recruitment Auctions

The crowdsourcing system considered in this work consists of a platform and a large set of workers $\mathcal{W} = \{w_1, \cdots, w_i, \cdots, w_N\}$. Like many existing works [1], [2], [3], [4], [8], [9], [11], [53], we adopt the *reverse auction* as an incentive mechanism to recruit workers. The platform plays the role of the auctioneer and announces the to-do task. In many cases, a task acquires 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 a comprehensive demography, as people with different age, region, education level, etc. may have distinct perceptions. Thus, like [53], [54], each worker is exclusively classified into one of the following *types* $\mathcal{T} = \{T_1, \cdots, T_j, \cdots, T_M\}$. 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 $\mathcal{T}$[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 from all workers and taking account of their types, the platform selects winners for this task. Following a conventional setting, it formulates and solves the following worker recruitment problem

$$\begin{aligned} \min \quad & \sum_{i:w_i \in \mathcal{W}} P_i x_i \\ s.t. \quad & \sum_{i:w_i \in T_j} k_i x_i \geq l_j, \quad \forall j \in [1, M], && (1) \\ & \bigcup_{j:w_i \in T_j, \forall x_i = 1} T_j = \mathcal{T}, && (2) \\ & x_i \in \{0, 1\}, \quad \forall i \in [1, N] \end{aligned}$$

where 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. Besides, the worker's "reputation" should be taken into account. Typically, highly-rated workers overweight the bad-mouthed ones. Here, we use coefficient $k_i \in [0, 1]$ to represent w_i's reputation. This value can be maintained and updated by the platform from a long-term observation. For example, in Amazon Mechanical Turk [45] and Guru [19], task requesters are allowed to set the

1. We pertain the discussion over bid collusion in this paper, collusion of misreporting type information [7] and tasks' answers [5], [34] 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 $\mathcal{T}' \subseteq \mathcal{T}$ with minor modification to the scheme.

preference to workers with high ratings. Constraint (1) says that the *weighted cardinality* of the hired worker set cannot be lower than l_j, a threshold determined by the platform to provide quality-guaranteed services. Constraint (2) requires that all worker types should be covered. Moreover, any solution to the above problem should also satisfy some inherent economic properties, such as *truthfulness* and *individual rationality*. Finally, the platform calculates payment P_i 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.

Definition 1. *(A Worker's Utility.) The utility of a worker $w_i \in \mathcal{W}$ is*

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

Definition 2. *(A Coalition's Group Utility.) The group utility of a coalition $\mathcal{G}$ is*

$$u_{\mathcal{G}} = \sum_{i: w_i \in \mathcal{G}} u_i,$$

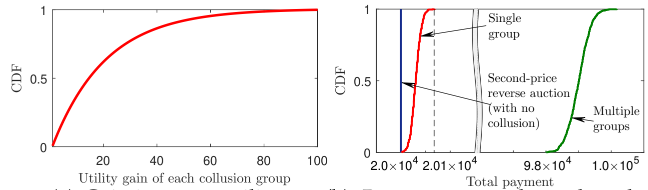
i.e., the sum of individual utility from all workers in the same coalition.

2.2 Collusions in Worker Recruitment Auctions

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 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 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* [48] 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, the winners are w_1 and w_2, each paid at 20. Now assume that w_2 and w_3 collude, i.e., $\mathcal{G} = \{w_2, w_3\}$ and w_3 raises its bid to 59. In this case, even though w_3 still loses, 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 1(a) we plot the group utility improvement of each coalition in 100 trials.



(a) Gain in group utility. (b) Payment caused to the platform.

Fig. 1. Small-size collusions have severe impact on auction-based crowdsourcing marketplace.

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 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 cost to 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.

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

$$\sum_{i: w_i \in \mathcal{G}} u_i(\boldsymbol{b}_{\mathcal{G}}, \boldsymbol{c}_{\mathcal{W} \setminus \mathcal{G}}) > \sum_{i: w_i \in \mathcal{G}} u_i(\boldsymbol{c}_{\mathcal{G}}, \boldsymbol{c}_{\mathcal{W} \setminus \mathcal{G}})$$

where $\boldsymbol{b}_{\mathcal{G}} \diamond \boldsymbol{c}_{\mathcal{G}}$, denoting that some elements of $\boldsymbol{b}_{\mathcal{G}}$ and $\boldsymbol{c}_{\mathcal{G}}$ satisfy $b_i > c_i$, some of them satisfy $b_i < c_i$, and the rest are the same.

Note that multiple coalitions may coexist in an auction. Besides, one worker can participate in different coalitions. The proposed scheme should be capable of handling all these situations. While workers may collude by misreporting other information in addition to bids, we focus the discussion over bid collusion in this paper, due to the design complexity.

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.

3.1 Basic Scheme Design

Upon receiving each worker w_i's bid b_i, the platform checks its type via accessing its profile. 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

$\mathcal{E} = \{E_j : j \in [1, M]\}$ and $\boldsymbol{\eta}_j = \{\eta_i : \forall i \; w_i \in E_j\}$. In order to recruit workers covering all types, winners are selected in each E_j.

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

$$a \leq \min_i \{\eta_i\}, \quad R \cdot a \geq \max_i \{\eta_i\}. \tag{3}$$

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

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

For each $E_j \in \mathcal{E}$, the platform first generates a set $\mathcal{A}_j = \{r_j \cdot a : \sum_{i:\eta_i \in \Gamma_{r_j \cdot a}(\boldsymbol{\eta}_j)} k_i \geq l_j, r_j \in [1, 2, \cdots, R]\}$ from $\mathcal{A}$. It then identifies the value $r_j^* \cdot a$ from $\mathcal{A}_j$ such that $(r_j^* \cdot a) \cdot \sum_{i:\eta_i \in \Gamma_{r_j^* \cdot a}(\boldsymbol{\eta}_j)} k_i$ 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 $(r_j^* \cdot a) \cdot \sum_{i:\eta_i \in \Gamma_{r_j^* \cdot a}(\boldsymbol{\eta}_j)} k_i$. Note that $r_j^* \cdot a$ can be viewed as winner's *per unit weight payment*.

Proposition 1. *For $r_j^* \cdot a$ of any $E_j \in \mathcal{E}$ that satisfies*

$$r_j^* \cdot a = \arg \min_{r_j \cdot a \in \mathcal{A}_j} \Big((r_j \cdot a) \cdot \sum_{i:\eta_i \in \Gamma_{r_j \cdot a}(\boldsymbol{\eta}_j)} k_i \Big),$$

we have

$$\sum_{i:\eta_i \in \Gamma_{r_j^* \cdot a}(\boldsymbol{\eta}_j)} k_i \geq l_j, \tag{4}$$

$$\sum_{i:\eta_i \in \Gamma_{(r_j^* - 1) \cdot a}(\boldsymbol{\eta}_j)} k_i < l_j. \tag{5}$$

Proof. First of all, we directly have (4) according to how $r_j^* \cdot a$ is derived in the basic scheme. We then prove (5) via the contradiction method. Assume that $\sum\limits_{i:\eta_i \in \Gamma_{(r_j^*-1) \cdot a}(\boldsymbol{\eta}_j)} k_i \geq l_j$. We have

$$(r_j^* \cdot a) \cdot \sum_{i:\eta_i \in \Gamma_{r_j^* \cdot a}(\boldsymbol{\eta}_j)} k_i \leq [(r_j^* - 1) \cdot a] \cdot \sum_{i:\eta_i \in \Gamma_{(r_j^*-1) \cdot a}(\boldsymbol{\eta}_j)} k_i, \tag{6}$$

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

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

and thus

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

which contradicts with (6). It implies that the assumption $\sum_{i:\eta_i \in \Gamma_{(r_j^*-1) \cdot a}(\boldsymbol{\eta}_j)} k_i \geq l_j$ is invalid. Thus, (5) holds. □

Proposition 1 provides an efficient way to determine winning workers and their payments. Specifically, once $\mathcal{A}_j$ is generated for each worker type T_j, instead of comparing $(r_j \cdot a) \cdot \sum_{i:\eta_i \in \Gamma_{r_j \cdot a}(\boldsymbol{\eta}_j)} k_i$'s for each element in $\mathcal{A}_j$ and identifying the minimum one, the first element of $\mathcal{A}_j$ 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 $\mathcal{O}(MN)$.

3.2 A Walk-through Example

Consider that there are thirteen workers ($N = 13$) and one task that looks for two types of workers ($M = 2$). Besides, let $l_j = 2$ ($j \in \{1, 2\}$). 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 $\boldsymbol{\eta}_j$ as

$$\boldsymbol{\eta}_1 = \{30, 34, 38, 40, 44, 48, 60\},$$
$$\boldsymbol{\eta}_2 = \{33, 36, 38, 39, 50, 62\},$$

Let $a = 4$, then $\mathcal{A}_1 = \{10 \cdot 4, 11 \cdot 4, \cdots, 15 \cdot 4\}$ and $\mathcal{A}_2 = \{9 \cdot 4, 10 \cdot 4, \cdots, 15 \cdot 4\}$ since $\sum_{i:\eta_i \in \Gamma_{10 \cdot 4}(\boldsymbol{\eta}_1)} k_i = 2.5 > l_1$ and $\sum_{i:\eta_i \in \Gamma_{9 \cdot 4}(\boldsymbol{\eta}_2)} k_i = 2 = l_2$. According to Proposition 1, $r_1^* \cdot a = 5 \cdot 4 = 40$ 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.

4 Final Collusion-Resistant Scheme

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

4.1 Collusion Patterns

Consider a worker set $E_j \in \mathcal{E}$. 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 $\mathcal{G}_j$ in E_j with *weighted cardinality* $t_j = \sum_{i:w_i \in \mathcal{G}_j} k_i$. When k_i's are all 1's, t_j is directly the cardinality of $\mathcal{G}_j$, 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 2.2. More precisely, each winner's per unit weight payment can be increased *up to* $r_j^H \cdot a$, satisfying

$$\sum_{i:\eta_i \in \Gamma_{r_j^H \cdot a}(\boldsymbol{\eta}_j)} k_i - t_j = \sum_{i:\eta_i \in \Gamma_{r_j^* \cdot a}(\boldsymbol{\eta}_j)} k_i. \tag{7}$$

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

$$\sum_{i:\eta_i \in \Gamma_{r_j^L \cdot a}(\boldsymbol{\eta}_j)} k_i + t_j = \sum_{i:\eta_i \in \Gamma_{r_j^* \cdot a}(\boldsymbol{\eta}_j)} k_i. \tag{8}$$

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

$$r_j^* \cdot a = r_j \cdot a, \quad \forall r_j \in [r_j^L, r_j^H] \tag{9}$$

i.e., winners are paid undifferentiated no matter a coalition colludes or not, the motivation of collusions will be diminished. Nonetheless, the above assumption does not hold unless under some modification.

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. In particular,

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

$$\Pr\left[u_i(\boldsymbol{b}_i, \boldsymbol{c}_{-i}) \leq u_i(\boldsymbol{c}_i, \boldsymbol{c}_{-i})\right] \geq p, \quad \forall w_i \in \mathcal{W}$$

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

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

In order to defend 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

$$h_u^\theta(v) = v \text{ rounded up to nearest } \theta^{s+u} \tag{10}$$

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 $\mathcal{H}$ 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 [33], 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 [33] 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 [33] 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 [33].

Corollary 1. *For h from $\mathcal{H}$, $h(v)$ is distributed identically to $\theta^U v$ for U uniform on $[0, 1]$.*

Proof. Consider a random variable $Y = log_k h(v)$ and let $t = log_k v$. Then $P[P \leq t + x] = P[U' \leq x]$ and therefore Y is uniformly distributed between t and $t + 1$. Thus, $h(v)$ is identical to $k^{U'} v$. □

Proposition 2. *For h from $\mathcal{H}$, the probability $h(v)$ is a $[\alpha, \beta]$-consensus estimate of v ($v > 0$) 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

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

□

We are now ready to introduce our (t, p)-collusion resistant worker recruitment auction. Its pseudo-code is presented in Algorithm 1. Upon receiving bids from workers, the platform derives $\mathcal{E}$. For each $E_j \in \mathcal{E}$, the platform generates $\mathcal{A}_j$ and identifies the value $r_j^* \cdot a$ from $\mathcal{A}_j$ such that $(r_j^* \cdot a) \cdot \sum_{i:\eta_i \in \Gamma_{r_j^* \cdot a}(\boldsymbol{\eta}_j)} k_i$ is minimized. According to Proposition 1, $r_j^* \cdot a$ is simply the first element of $\mathcal{A}_j$. 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 1 (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 ($i \in [1, N]$)

1: **for** each $T_j \in \mathcal{T}$ **do**
2: Platform generates worker set E_j and $\mathcal{A}_j$;
3: Identify the first value in $\mathcal{A}_j$ and set it as $r_j^* \cdot a$;
4: Select $h_u^{\theta_j}(\cdot)$ based on t_j and $\boldsymbol{\eta}_j$;
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 $P_i = k_i \cdot h_u^{\theta_j}(r_j^* \cdot a)$.
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

$$h_u^{\theta_j}(r_j \cdot a) = h_u^{\theta_j}(r_j^* \cdot a) \quad r_j \in [r_j^L, r_j^H] \tag{11}$$

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 2, p_j is calculated as

$$p_j = 1 - log_{\theta_j} \frac{r_j^H \cdot a}{r_j^L \cdot a} = 1 - log_{\theta_j} \frac{r_j^H}{r_j^L} \quad (12)$$

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 (11) hold. For this purpose, we first identify $r_j^H \cdot a$ and $r_j^L \cdot a$ via (7) and (8), respectively. Then θ_j is carefully selected such that (11) holds for $v \in [r_j^L \cdot a, r_j^H \cdot a]$. Specifically, we should expect

$$r_j^L \cdot a \leq h_u^{\theta_j}(r_j^L \cdot a) \leq r_j^L \cdot a \cdot \theta_j,$$

$$r_j^* \cdot a \leq h_u^{\theta_j}(r_j^* \cdot a) \leq r_j^* \cdot a \cdot \theta_j,$$

$$r_j^H \cdot a \leq h_u^{\theta_j}(r_j^H \cdot a) \leq r_j^H \cdot a \cdot \theta_j.$$

Together with (11) 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 $\mathcal{E}$. 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 2.

Theorem 1. *The computation complexity of our (t, p)-collusion resistant worker recruitment algorithm is upper bounded by $\mathcal{O}(MN)$.*

Proof. The computation complexity of Algorithm 1 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 $\mathcal{A}_j$ (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 1 is upper bounded by $\mathcal{O}(M(2N+3R))$. Recall that R is a constant value in the algorithm. The computation complexity is thus rewritten as $\mathcal{O}(MN)$. □

4.3 A Walk-through Example

To better explain our scheme, we still take the example in Section 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 $\mathcal{A}_1 = \{10 \cdot 4, \cdots, 15 \cdot 4\}$ and $\mathcal{A}_2 = \{9 \cdot 4, 10 \cdot 4, \cdots, 15 \cdot 4\}$ 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 coalition with weighted cardinality up to $t = 1.5$ in each worker type. According to (7) and (8), 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.

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 $E_j \in \mathcal{E}$, i.e., they belong to the same type $T_j \in \mathcal{T}$. Such kind of collusions 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 3.2 and 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 $\mathcal{G} = \{w_2, w_3, w_6\}$, it can be divided into two intra-collusion coalitions, i.e., $\mathcal{G}_1 = \{w_2, w_3\}$ and $\mathcal{G}_2 = \{w_6\}$. To sum up, if we can effectively discourage the formation of intra-collusion coalitions, so for the inter-collusion coalitions.

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 (7) and (8) should meet, or equivalently,

$$\sum\nolimits_{i:\eta_i \in \Gamma_{r_j^L \cdot a}(\boldsymbol{\eta}_j)} k_i = \sum\nolimits_{i:\eta_i \in \Gamma_{r_j^* \cdot a}(\boldsymbol{\eta}_j)} k_i - t_j. \quad (13)$$

$$\sum\nolimits_{i:\eta_i \in \Gamma_{r_j^H \cdot a}(\boldsymbol{\eta}_j)} k_i = \sum\nolimits_{i:\eta_i \in \Gamma_{r_j^* \cdot a}(\boldsymbol{\eta}_j)} k_i + t_j. \quad (14)$$

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

$$\sum\nolimits_{i:\eta_i \in \Gamma_{r_j^L \cdot a}(\boldsymbol{\eta}_j)} k_i > 0, \quad (15)$$

$$\sum\nolimits_{i:\eta_i \in \Gamma_{r_j^H \cdot a}(\boldsymbol{\eta}_j)} k_i \leq \sum\nolimits_{i:w_i \in E_j} k_i. \quad (16)$$

Substituting (13) and (14) into (15) and (16) respectively, we derive following restrictions on t_j:

$$t_j < \sum_{i:\eta_i \in \Gamma_{r_j^* \cdot a}(\boldsymbol{\eta}_j)} k_i, \tag{17}$$

$$t_j \leq \sum_{i:w_i \in E_j} k_i - \sum_{i:\eta_i \in \Gamma_{r_j^* \cdot a}(\boldsymbol{\eta}_j)} k_i. \tag{18}$$

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 $l_j \leq \sum_{i:\eta_i \in \Gamma_{r_j^* \cdot a}(\boldsymbol{\eta}_j)} k_i$, then (17) 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 (18) 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.

5 Performance Analysis

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

Lemma 1. *For any $E_j \in \mathcal{E}$, our scheme achieves (t_j, p_j)-collusion resistance, with p_j defined by (12).*

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 $\mathcal{T}_j' \subseteq \mathcal{G}_j$ ($\mathcal{T}_j \subseteq \mathcal{G}_j$) 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 may lose, thus $\mathcal{T}_j' \subseteq \mathcal{T}_j$. The difference between the coalition in E_j's group utility when they collude or not is calculated as

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

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 (11). Hence, the second equation above holds with a probability p_j. Besides, for a colluder $w_i \in \mathcal{T}_j \setminus \mathcal{T}_j'$, 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 1, 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 $\mathcal{T}_j'' \subseteq \mathcal{G}_j$ ($\mathcal{T}_j \subseteq \mathcal{G}_j$) 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 $\mathcal{T}_j \subseteq \mathcal{T}_j''$. The difference between the coalition in E_j's group utility when they collude or not is calculated as

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

with a probability p_j. 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 (11). Hence, the second equation above holds at p_j. Besides, for a colluder $w_i \in \mathcal{T}_j'' \setminus \mathcal{T}_j$, 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 1, 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 as above, we omit its discussion here. □

Based on Lemma 1, we are ready to give the following theorem on collusion resistance.

Theorem 2. *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 $E_j \in \mathcal{E}$. And collusions in different E_j's are independent with each other. According to Lemma 1, 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 $\mathcal{G}$ of weighted cardinality t' forming across M E_j's and $t' = \sum_{j=1}^{M} t_j$. From the discussion of Section 4.4, $\mathcal{G}$ can be equivalently divided into M intra-type coalitions, each with the weighted cardinality t_j. Besides, we have proved in Lemma 1 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 3. *Our scheme is p-individual rational, i.e.,*

$$\Pr\left[u_i \geq 0\right] \geq p \quad \forall i \in [1, N].$$

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

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 3. *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.

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

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 $\sum_{i:\eta_i \in \Gamma_{r_j^* \cdot a}(\boldsymbol{\eta}_j)} k_i \geq l_j$. 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 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 trade of its critical properties.

Theorem 4. *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 $\boldsymbol{b}_j^{(k)}$. We have

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

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

6 Performance Evaluation

6.1 Dataset

To validate the proposed scheme, we employ a real-world dataset obtained from the commercial crowdsourcing platform Guru [19]. We focus on tasks in the field of Programming & Development, which involve 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 $\mathcal{T}$, the type set of our scheme. Thus, the cardinality of $\boldsymbol{T}$ is used to instantiate M. 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. We assume that at most 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 by default. All simulation results are the average over 100 trials.

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.

In the simulation, the coalition's weighted cardinality is first set to 5. We observe from Fig. 2(a) 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 remaining coalitions' average

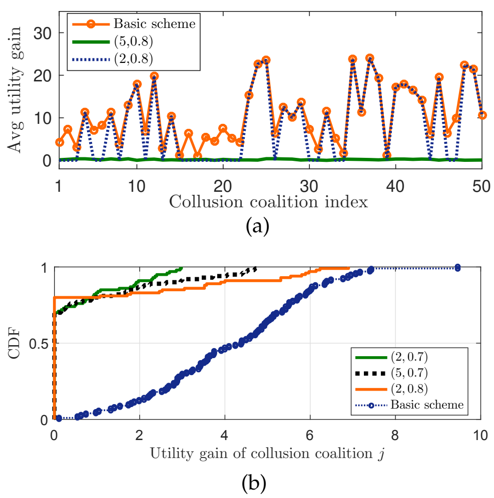


Fig. 2. Collusion resistance performance comparison. (a) Average utility gain. (b) Cumulative distribution of instance utility gain.

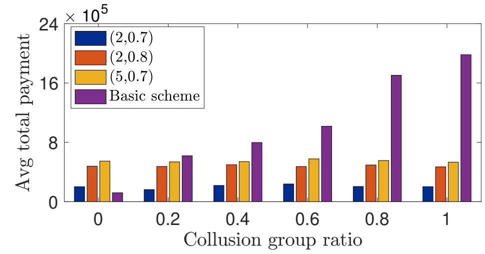


Fig. 3. Total payment comparison under different collusion group ratios.

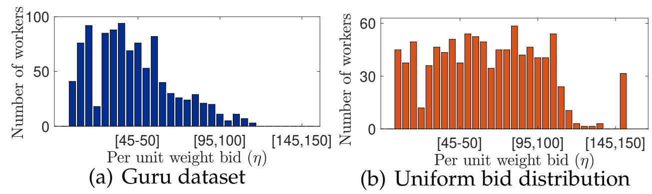


Fig. 4. Distribution of *per unit weight bids* of two different datasets.

utility is same as that of the basic scheme, which denotes $(2, 0.8)$ resist parts of collusions because colluders size is larger than $t = 2$.

Fig. 2(b) depicts the cumulative distribution of instant utility gain of a randomly selected coalition. The coalition's maximal utility gain is as high as 9.5 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 2.8, which is smaller than that under $(5, 0.7)$, i.e., 4.8. 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 (12). 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.

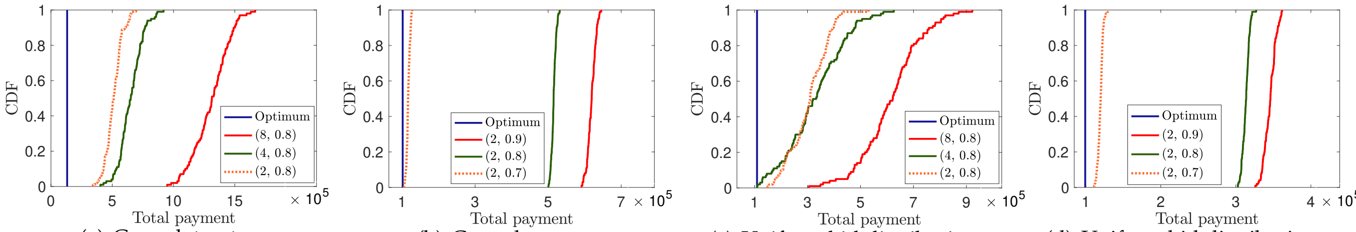
Fig. 3 shows the collusion resistance property of our scheme by evaluating the total payment occurred at the platform. When no coalition exists, the total payment of the basic scheme is less than that of the proposed scheme. However, the payment increases dramatically as more collusion takes place, while 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 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 collusion, its causes significantly increased payment as more malicious workers are present.

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.

Fig. 5 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. 5(a) and 5(b) are derived from the Guru dataset, with its *per unit weight bid* η distribution shown in Fig. 4(a). We find that our scheme achieves the above-mentioned properties at the cost of higher total payment. Specifically, as shown in Fig. 5(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. 5(c) and 5(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. 4(a). 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. 6 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 quantify 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



(a) Guru dataset (b) Guru dataset (c) Uniform bid distribution (d) Uniform bid distribution

Fig. 5. Total payment caused by our scheme and P_{opt}.

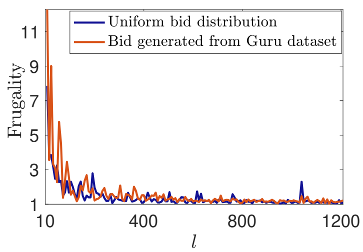
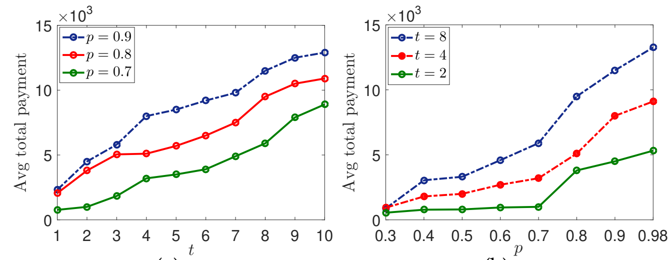
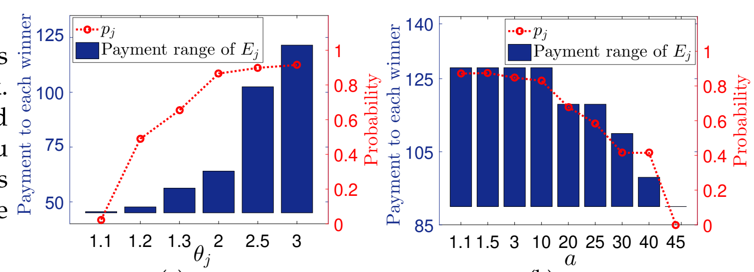


Fig. 6. The frugality over different l's.



(a) (b)

Fig. 7. The platform's average total payment under different settings.



(a) (b)

Fig. 8. Scheme performances under different settings.

barely over pays. 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. 7 shows the platform's average payment of a randomly selected task under different combinations of t and p. From Fig. 7(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 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. 7(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 $E_j \in \mathcal{E}$, in order to achieve a larger p_j, we are expecting a larger θ_j according to (12). Thus, from the definition of $h_u^{\theta_j}(\cdot)$ in (10), 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. 7(b).

Fig. 7 provides some insightful observations of out (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.

6.4 Impact of parameters

In this part, we evaluate the impact of different parameters to the performance of our scheme. Specifically, we concentrate on two of the most critical ones, θ and a. A feasible worker set $E_j \in \mathcal{E}$ 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. 8(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. 8(b), where we set $t_j = 2$ and $\theta_j = 2$. Recall that a should also meet the requirement (3). 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.

7 Related Work

Collusion resistance in crowdsourcing. Collusion resistance has rarely been investigated in crowdsourcing. An initial research is conducted by Ji and Chen [26] with the focus on achieving *group strategy-proofness*, whereby a member that benefits from the coalition strategy will not payoff another member that suffers a loss [31]. 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 [34] 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 [13], [14], collusion resistance is also investigated therein. Ji and Liu [41] proposed a collusion-resistant dynamic pricing approach to maximize the users' utilities while combating their collusive behaviors using the derived optimal reserve prices. [42], [43] 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. [44] then developed a general collusion-resistant framework for dynamic spectrum auctions.

Collusion resistance in general auctions. Since Robinson [12] gave theoretical evidence that auctions are vulnerable to collusions, there only have been a handful of works on collusion-resistant mechanism design [27], [28], [29], [30]. Che and Kim [28], [29] 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 [30] 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 [27], 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 reputation. 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 [36], [37], [38], [39], [40]. Notice that they focus on theoretical understanding of collusion performances under different auction settings, rather than collusion-resistant mechanism design.

8 Conclusion

In this paper we develop a (t, p)-collusion resistant scheme for worker recruitment auctions in crowdsourcing. 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 formal analysis over the tradeoff. Simulation results demonstrate the effectiveness of our scheme.

References

[1] 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.

[2] 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.

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

[4] 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.

[5] Jurca, Radu and Faltings, Boi, "Minimum payments that reward honest reputation feedback," in *Proceedings of the ACM conference on Electronic commerce*, 2006.

[6] D. Yang, G. Xue, X. Fang, and J. Tang, "Incentive mechanisms for crowdsensing: Crowdsourcing with smartphones," *IEEE/ACM Transactions on Networking*, vol. 24, no. 3, pp. 1732–1744, June 2016.

[7] Hu, Jia and Lin, Hui and Guo, Xuancheng and Yang, Ji, "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.

[8] 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. PP, no. 99, pp. 1–1, December 2017.

[9] 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.

[10] J. Xu, J. Xiang, and D. Yang, "Incentive mechanisms for time window dependent tasks in mobile crowdsensing," *IEEE Transactions on Wireless Communications*, vol. 14, no. 11, pp. 6353–6364, July 2015.

[11] 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, December 2014.

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

[13] 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.

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

[15] P. Bajari and J. T. Fox, *Complementarities and collusion in an FCC spectrum auction*. National Bureau of Economic Research, 2005.

[16] 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.

[17] 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.

[18] 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.

[19] Guru. [Online]. Available: https://www.guru.com/

[20] 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

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

[22] 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.

[23] Y. Liu, Y. Yang, and Y. L. Sun, "Detection of collusion behaviors in online reputation systems," in *Proceedings of Asilomar Conference on Signals, Systems and Computers*, 2008.

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

[25] Q. Lian, 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.

[26] S. Ji and T. Chen, "On designing collusion-resistant incentive mechanisms for mobile crowdsensing systems," in *Proceedings of the IEEE International Conference on Trust, Security and Privacy/ Big Data Service and Engineering/ Education and Social Science*, 2017.

[27] A. V. Goldberg and J. D. Hartline, "Collusion-resistant mechanisms for single-parameter agents," in *Proceedings of the ACM-SIAM International Symposium on Discrete Algorithms*, 2005.

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

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

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

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

[32] Technical report. [Online]. Available: https://www.dropbox.com/s/1puqmuekzn7772r/techreport.pdf?dl=0

[33] A. V. Goldberg and J. D. Hartline, "Competitiveness via consensus," in *Proceedings of the ACM-SIAM International Symposium on Discrete Algorithms*, 2003.

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

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

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

[37] 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.

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

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

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

[41] Z. Ji and K. J. 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, January 2008.

[42] Y. Wu, B. Wang, K. J. 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, December 2009.

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

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

[45] Amazon Mechanical Turk. [Online]. Available: https://www.mturk.com/

[46] Figure Eight. [Online]. Available: https://www.figure-eight.com/

[47] Figure Eight. [Online]. Available: https://cran.r\-project.org/web/packages/fitdistrplus/fitdistrplus.pdf

[48] N. Nisan and T. Roughgarden and E. Tardos and V. Vazirani, "Algorithmic game theory," in *Cambridge University Press Cambridge*, 2007.

[49] 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.

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

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

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

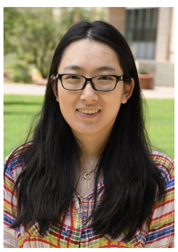
[53] L. Huang and Y. Zhu, J. Yu and M. Wu "Group Buying Based Incentive Mechanism for Mobile Crowd Sensing," in *Proceedings of the IEEE SECON*, 2016.

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

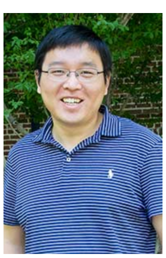
[55] G. Arjun K and N. Saralees "Handbook of beta distribution and its applications," in *CRC press*, 2004.

Mingyan Xiao received her M.S. degree from National University of Defense Technology in 2017, and B.E. from Nanjing University of Aeronautics and Astronautics in 2014, respectively.She is currently a Ph.D. student in the Department of Computer Science at The University of Texas, at Arlington. Her recent research interests are in the area of resource allocation and management in wireless networks, data-driven security and privacy.

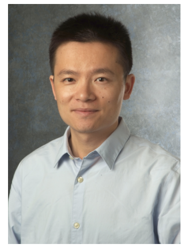
Wenqiang Jin received he B.E. and M.E. in Electronic Engineering from Chongqing University of Posts and Telecommunications, China, in 2011 and 2017, respectively. He is currently an Ph.D. student in the Department of Computer Science and Engineering, The University of Texas at Arlington. His research interests include mobile crowd sensing and IoT security.



Ming Li received the B.E. degree in Electrical Engineering from Sun Yat-sen University, China, in 2007, the M.E. degree in Electrical Engineering from Beijing University of Posts and Communications, China, in 2010, and the Ph.D. degree in Electrical and Computer Engineering from Mississippi State University, Starkville, in 2014, respectively. She is currently an assistant professor in the Department of Computer Science and Engineering, The University of Texas at Arlington. Her research interests include mobile computing, internet of things, security, and privacy-preserving computing. Her work won Best Paper Awards in Globecom 2015 and DASC 2017, respectively. She received the NSF CAREER Award in 2020 and is a member of the IEEE.



Lei Yang received the B.S. and M.S. degrees in electrical engineering from Southeast University, Nanjing, China, in 2005 and 2008, respectively, and the Ph.D. degree from the School of Electrical Computer and Energy Engineering, Arizona State University, Tempe, AZ, USA, in 2012. He was a Postdoctoral Scholar with Princeton University, Princeton, NJ, USA, and an Assistant Research Professor with the School of Electrical Computer and Energy Engineering, Arizona State University. He is currently an Assistant Professor with the Department of Computer Science and Engineering, University of Nevada, Reno, NV, USA. His research interests include big data analytics, edge computing and its applications in IoT and 5G, stochastic optimization and modeling in smart cities and cyber-physical systems, data privacy and security in crowdsensing, and optimization and control in mobile social networks. He was a recipient of the Best Paper Award Runner-up at the IEEE INFOCOM 2014. He is currently associate editor for IEEE Access.



Pan Li received the BE degree in electrical engineering from the Huazhong University of Science and Technology, Wuhan, China, in 2005, and the PhD degree in electrical and computer engineering from the University of Florida, Gainesville, in 2009, respectively. He is currently an associate professor with the Department of Electrical Engineering and Computer Science at Case Western Reserve University. He was an assistant professor in the Department of Electrical and Computer Engineering at Mississippi State University between August 2009 and August 2015. His research interests include network science and economics, energy systems, security and privacy, and big data. He has been serving as an editor for IEEE Wireless Communications Letters, IEEE Journal on Selected Areas in Communications—Cognitive Radio Series and IEEE Communications Surveys and Tutorials, a feature editor for IEEE Wireless Communications, a general chair for IEEE SmartData 2017, and a technical program committee (TPC) co-chair for the Ad-hoc, Mesh, Machine-to-Machine and Sensor Networks Track, IEEE VTC 2014, Physical Layer Track, Wireless Communications Symposium, WTS 2014, and Wireless Networking Symposium, IEEE ICC 2013. He received the NSF CAREER Award in 2012 and is a member of the IEEE and the ACM.