Incentive Mechanism for Uncertain Tasks under Differential Privacy

Jiang, Xikun; Ying, Chenhao; Li, Lei; Wu, Haiqin; Luo, Yuan; Düdder, Boris

Publication date:
2023

Document version
Publisher's PDF, also known as Version of record

Document license:
Unspecified

Citation for published version (APA):
Abstract—Mobile crowd sensing (MCS) has emerged as an increasingly popular sensing paradigm due to its cost-effectiveness. This approach relies on platforms to outsource tasks to participating workers when prompted by task publishers. Although incentive mechanisms have been devised to foster widespread participation in MCS, most of them focus only on static tasks (i.e., tasks for which the timing and type are known in advance) and do not protect the privacy of worker bids. In a dynamic and resource-constrained environment, tasks are often uncertain (i.e., the platform lacks a priori knowledge about the tasks) and worker bids may be vulnerable to inference attacks. This paper presents HERALD*, an incentive mechanism that addresses these issues through the use of uncertainty and hidden bids. Theoretical analysis reveals that HERALD* satisfies a range of critical criteria, including truthfulness, individual rationality, differential privacy, low computational complexity, and low social cost. These properties are then corroborated through a series of evaluations.

Index Terms—Uncertain Sensing Tasks, Differential Privacy, Mobile Crowd Sensing, Incentive Mechanism.

I. INTRODUCTION

The proliferation of mobile devices equipped with advanced processors and numerous sensors (like GPS and microphones), has been a key driver behind the rise of mobile crowd sensing (MCS) as a prominent sensing paradigm. MCS relies on a pool of workers equipped with mobile devices to gather sensory data and has fueled the development of various applications, including smart transportation and traffic management. Numerous MCS systems have been developed and implemented accordingly in these and other areas [1]–[5].

Fig. 1 provides an illustration of a typical MCS system, in which platforms enlist participating workers to execute tasks upon request from demanders (also referred to as task publishers). The success of the majority of MCS applications hinges on the willingness of an adequate number of mobile workers to partake in the process, thereby ensuring the provision of high-quality services. However, workers may exhibit hesitance towards participating in MCS, since task execution may drain their battery power, storage, computation, and communication resources. Furthermore, participating in MCS may render workers’ private information (including their location and bid details) vulnerable to exposure during data collection and exchange. To counterbalance these costs and safeguard their privacy, it is crucial to furnish workers with suitable incentives that do not jeopardize their confidentiality.

The importance of incentives has led to the development of numerous mechanisms [6]–[24] to encourage participation in MCS. However, a considerable number of these mechanisms are grounded on the presumption that tasks are static (i.e., the platform knows the timing and type of tasks in advance). In the real world, MCS tasks are often uncertain due to their unknown arrival time and incomplete information known to the platform. Additionally, these mechanisms do not protect the privacy of worker bids, which may be exposed to potential inference attacks [25] if published by the platform. Therefore, there is a desperate need for an incentive mechanism that can handle uncertain tasks and protect worker bid privacy in MCS systems, meanwhile satisfying a set of desired properties such as truthfulness, individual rationality, differential privacy, and low social cost.

In an MCS system, tasks such as collecting information on the number of bends, bifurcations, or roadside shops can be completed in advance, which reduces latency and improves efficiency by allowing the platform to respond immediately when these tasks arrive. These tasks are referred to as non-real-time tasks. However, due to the complexity of the real-world environment, it is often difficult for the platform to predict which tasks will arrive in the future and when they will arrive. These unpredictable tasks are referred to as uncertain tasks.

Designing a suitable incentive mechanism for MCS systems, which are susceptible to both uncertain tasks and inference attacks, poses a considerable challenge. To tackle this issue,
we consider a scenario where the arrival of uncertain tasks is governed by a probability distribution and employ the exponential mechanism, a technique from differential privacy, to safeguard the privacy of worker bids. Our proposed solution, HERALD*, which satisfies truthfulness, individual rationality, and differential privacy, while also having low computational complexity and social cost. The main contributions of this paper are:

- **Mechanism**: In contrast to previous approaches, our work introduces a new incentive mechanism named HERALD*, which integrates differential privacy. Specifically, HERALD* is tailored for uncertain tasks that are expected to arrive based on a probability distribution, enabling the platform to gather sensory data in advance. Additionally, we employ the exponential mechanism to protect the privacy of worker bids.

- **Desirable Properties**: HERALD* can effectively encourage worker participation and achieves a set of desirable properties, such as truthfulness, individual rationality, differential privacy, low computational complexity, and low social cost. Unlike other incentive mechanisms such as those proposed in [6]–[10] for traditional MCS, HERALD* is specifically designed for uncertain tasks and is not limited to collecting large amounts of sensory data. Furthermore, we demonstrate that HERALD* has a competitive ratio of $O(\ln{\ln{n}})$ in terms of expected social cost, where $l$ and $n$ represent the number of task subsets and tasks published in advance, respectively. We also prove that HERALD* preserves $(\epsilon, \delta)$-differential privacy for both linear and logarithmic score functions, where $\epsilon > 0$ is a constant.

- **Evaluations**: In addition to HERALD*’s desirable properties, we performed comprehensive simulations to verify its efficacy. Our findings indicate that HERALD* outperforms current approaches by exhibiting a lower anticipated social cost and total payment while simultaneously providing differential privacy.

The subsequent sections of this paper are structured as follows: Section II provides a discussion of the related literature, while Section III presents an introduction to the preliminaries. Section IV outlines the design and theoretical analysis of HERALD*. In Section V, extensive simulations are conducted to validate the properties of the proposed mechanism. Lastly, Section VI concludes the paper.

### II. RELATED WORK

Numerous incentive mechanisms [6]–[24] have been proposed for MCS systems since attracting a considerable number of workers is crucial. In addition to truthfulness and individual rationality, these mechanisms often aim to ensure the benefits of both workers and the platform.

The authors of [6]–[8] proposed mechanisms to minimize the social cost, while [9], [10] aimed to maximize the platform’s profit. In contrast, [11]–[14] focused on minimizing the platform’s payment, and [17], [18] designed mechanisms to maximize social welfare. A novel crowdsourcing assignment strategy proposed in [19] considered fair task allocation for workers. In addition to the above objectives, there has been an effort to achieve other objectives as well. For instance, Bhattacharjee et al. in [20] incentivized workers to act honestly by evaluating the quantity and quality of their data. Gong et al. in [21] introduced an incentive mechanism to encourage workers to submit high-quality data, while Liu et al. in [22] addressed the problem of multi-resource allocation by devising a truthful double auction mechanism.

Moreover, several studies investigated privacy-preserving methods in mobile crowdsourcing. For example, Lin et al. [26] proposed a general privacy-preserving framework for incentivizing crowdsensing using two score functions. Hu et al. [23] developed a privacy-preserving incentive mechanism for dynamic spectrum sharing crowdsensing, while Han et al. [24] focused on privacy-preserving in budget-limited crowdsensing. Yang et al. [27] incorporated both additive secret sharing and local differential privacy technologies. Wei et al. [28] investigated location privacy-preserving in spatial crowdsourcing. Some studies have also used cryptography techniques [29], [30], or blockchain [15], [16], [31] to protect privacy in mobile crowdsourcing, but they do not consider the strategic behavior of the participants.

Most previously mentioned works have focused on static sensing tasks, where the platform has complete knowledge of the task information beforehand. However, in realistic environments with resource constraints, sensing tasks are often uncertain, meaning that the platform has incomplete knowledge of the task information. In contrast to these previous works, this paper addresses the issue of uncertain tasks while also ensuring the bidding privacy of participants. The incentive mechanism proposed in this study is tailored for uncertain tasks, where task information is unknown to the platform and tasks are expected to arrive according to a probability distribution. To safeguard the privacy of participants’ bids, the mechanism uses the exponential mechanism. We believe that this is the first time that an incentive mechanism has been tailored to accommodate uncertain tasks in an MCS system while incorporating differential privacy.

### III. PRELIMINARIES

In this section, we provide an overview of the system and discuss the objectives that guided its design.

#### A. System Overview

In this study, we focus on a mobile crowdsourcing system (MCS) consisting of a cloud-based platform and a group of participating workers (represented as $\mathcal{W} = \{1, 2, \ldots, m\}$). The platform is assumed to have a priori knowledge of a set of $n$ sensing tasks, $\mathcal{T} = \{\tau_1, \ldots, \tau_n\}$, and all future requested tasks will belong to $\mathcal{T}$. The platform divides these $n$ tasks into $l$ task subsets, $\mathcal{Y} = \{\Gamma_1, \Gamma_2, \ldots, \Gamma_l\}$, such that $\bigcup_{j=1}^{l} \Gamma_j = \mathcal{T}$, where $l < 2n$ and the number of tasks in each subset $\Gamma_j$ is random. To ensure task quality and prevent monopolization, we require that each task $\tau_i \in \mathcal{T}$ must be covered by no less than two task subsets. This condition is practical since

\[1\text{The name HERALD* is from incentive mechanism for uncertain tasks under differential privacy in mobile crowdsensing.}\]
the platform typically possesses some prior knowledge about the tasks that need to be performed. For instance, in a traffic monitoring application, the task set may comprise the number of all the intersections on a particular road. This task set remains static over time and has no real-time constraints. A similar application could be the monitoring of road curves. The system framework is depicted in Fig. 2, and its operation remains static over time and has no real-time constraints. A similar application could be the monitoring of road curves. The system framework is depicted in Fig. 2, and its operation remains static over time and has no real-time constraints.

**Incentive Mechanism for Uncertain Tasks:** As illustrated in Fig. 2 before the actually requested tasks arrive, the platform first matches workers to task subsets in a task-worker matching phase (step 3). For each task subset \( \Gamma_j \in \mathcal{Y} \), each worker \( i \) submits a bid \( b_i \). We assume that the probability that task subset \( \Gamma_j \) is matched with worker \( i \) is \( Pr(b_i) \). Note that in each round of task-worker matching, each worker can only participate once, and in the \( l \) rounds of matching, each worker \( i \) only has one unique bid \( b_i \). This means that for each task subset \( \Gamma_j \), the probability that worker \( i \) is matched is \( Pr(b_i) \). The platform randomly assigns the task subset to the workers according to this probability distribution. To ensure task quality and prevent monopolization, we require that each task \( \tau_i \in \mathcal{T} \) is matched by at least two different workers. After the task-worker matching phase, the platform receives the matching set \( \mathcal{P} \), which consists of matching pairs \((\Gamma_j, b_i)\) for \( \Gamma_j \in \mathcal{Y} \) and \( i \in \mathcal{W} \), where \( b_i \) is worker \( i \)'s bid for executing task subset \( \Gamma_j \) (step 2). Upon receiving a matching set \( \mathcal{P} \), the platform proceeds to determine the winning set \( \mathcal{S} \) and the corresponding payment \( p_i \) to each winning worker \( i \), as indicated by step 4. Additionally, the platform collects sensory data from the winning workers to enable prompt responses to future requests, as per step 4. It is worth noting that the platform collects sensory data for tasks in \( \mathcal{T} \) in advance, as it lacks any knowledge about future tasks, which are assumed to arrive according to a probability distribution.

In this mechanism, a worker who is not selected to execute any tasks, as a "loser", receives zero payment. We use the notation \( \overrightarrow{p} = (p_1, \ldots, p_m) \) to represent the payment profile of the workers, which is initialized to be zero. If we denote the cost of worker \( i \) as \( c_i \) in HERALD*, we can define the utility of the worker as

\[
\begin{align*}
  u_i &= \begin{cases} 
    p_i - c_i, & \text{if worker } i \text{ wins,} \\
    0, & \text{otherwise.} 
  \end{cases}
\end{align*}
\]

Without loss of generality, we assume that the bid \( b_i \) of each worker \( i \) is constrained within the range of \( [b_{\min}, b_{\max}] \), where \( b_{\min} \) is normalized to 1 and \( b_{\max} \) is a constant. We use the notation \( \overrightarrow{b} = (b_1, \ldots, b_m) \) to represent the bid profile of the workers for convenience. Additionally, it is assumed that for each worker \( i \) who has a specific subset of tasks, denoted as \( \Gamma_i \), there exist other workers \( j \) who have task subsets \( \Gamma_j \) that include the tasks of worker \( i \), i.e., \( \Gamma_i \subseteq \bigcup_j \Gamma_j \).

**B. Design Objectives**

The main objective of this manuscript is to establish that HERALD* exhibits the subsequent advantageous features. Given that workers may exhibit selfish or strategic behavior, it is plausible that any worker \( i \) may present a bid \( b_i \) that differs from their actual cost \( c_i \) of performing all the tasks in \( \Gamma_i \). Consequently, we aim to develop an incentive mechanism that satisfies truthfulness, which is defined as follows.

**Definition 1 (Truthfulness).** An incentive mechanism is truthful if for any worker \( i \in \mathcal{W} \), her utility is maximized when bidding her true cost \( c_i \).

According to Definition 1, our goal is to ensure that workers make truthful bids to the platform. In addition to truthfulness, we also strive to achieve another desirable property known as individual rationality, which is defined below:

**Definition 2 (Individual Rationality).** An incentive mechanism is individually rational if for any worker \( i \in \mathcal{W} \), her utility \( u_i \) satisfies \( u_i \geq 0 \).

Nonetheless, if the platform discloses the bidding results that contain the winning bidders and their rewards directly, it may expose the privacy of the participants to inference attacks, thereby impeding the advantages of the platform. Since any changes in workers’ bids can considerably influence the ultimate bidding results, particularly the payments, an adversarial worker may deduce the bids of other workers based on the distinct payments received in various auctions. To protect against this inference attack and ensure the privacy of the workers’ bids, it is necessary to devise a mechanism that fulfills differential privacy, which is defined as follows.

**Definition 3 (Differential Privacy).** A randomized mechanism \( M \) preserves \((\epsilon, \delta)-\)differential privacy if for any two input sets \( A \) and \( B \) with a single input difference, and for any set of outputs \( \mathcal{O} \in \text{Range}(M) \),

\[
Pr[M(A) \in \mathcal{O}] \leq \exp(\epsilon) \times Pr[M(B) \in \mathcal{O}] + \delta.
\]

If \( \delta = 0 \), we say that \( M \) preserves \( \epsilon \)-differential privacy.

Let \( Pr(b_i) \) represent the probability that worker \( i \) is matched with any task subset \( \Gamma_j \in \mathcal{Y} \) when their bid is \( b_i \). To safeguard the privacy of the workers’ bids, it is necessary to decrease the impact of the distinct bids of workers on the eventual bidding outcome. To do this, we introduce differential privacy, specifically using the exponential mechanism which is defined as follows.

**Definition 4 (The Exponential Mechanism).** The exponential mechanism \( M_E(x, u, R) \) selects and outputs an element \( r \in R \) with probability proportional to \( \exp(u(x,r)) \), i.e., \( Pr[M_E(x, u, R) = r] \propto \exp(u(x,r)) \), where \( x \) is the input set and \( u \) is a utility function that maps input/output pairs to utility.
scores, \( \Delta u = \max_{r \in R} \max_{x,y:|x-y|_{\leq 1}} |u(x,r) - u(y,r)| \) is the sensitivity of the utility function \( u \), and \( \epsilon \) is a small constant.

The following theorem is derived from the exponential mechanism mentioned above.

**Theorem 1 (\cite{23}).** The exponential mechanism \( M_E(x,u,R) \) preserves \((\epsilon,0)\)-differential privacy.

To ensure that changes in workers’ bids do not significantly affect the final bidding results, the exponential mechanism is implemented to achieve differential privacy in HERALD*.

This makes it difficult for malicious workers to infer the bidding details of other workers based on the final bidding results. Similar to previous work \cite{23}, \cite{24}, \cite{26}, \cite{28}, we incorporate randomization into the incentive mechanism’s outcome to achieve differential privacy.

Apart from the above objectives, we also strive for HERALD* to possess a low anticipated social cost, taking into account the probability distribution of the tasks in \( T \) arrive. We assess the competitive ratio of the system’s expected social cost as a measure to accomplish this aim.

**Definition 5 (Competitive Ratio on Expected Social Cost).** Suppose the tasks in the sensing task set \( T \) arrive in a probability distribution, for any set \( A \) of \( k \) tasks that may arrive at the same time from \( T \), let \( S(A,W) \) refer to the winning set chosen by the mechanism such that \( A \subseteq \cup_{(\Gamma_j,c_j) \in S(A,W)} \Gamma_j \) and \( \Gamma_j \cap A \neq \emptyset \) for all \( i \in S(A,W) \), \( C(S(A,W)) = \sum_{(\Gamma_j,c_j) \in S(A,W)} c_i \) be the corresponding social cost, and \( C_{OPT}(A,W) \) be the minimum social cost of a requested task set \( A \), respectively. The competitive ratio on expected social cost is defined as \( \max_k E_{PEF}[E_{A \subseteq T}[C(S(A,W))]/E_{A \subseteq T}[C_{OPT}(A,W)]] \), where \( E_{PEF}[\cdot] \) is the expectation over all matching set results \( F \) and \( E_{A \subseteq T}[\cdot] \) is the expectation over all sets of possibly \( k \) arriving tasks in the future.

It is worth noting that a worker may be included multiple times in the winning set, having the same cost but with a different subset of tasks since each worker can match with several subsets of tasks. Additionally, it is possible for some tasks in the task set \( A \) to be identical. Thus, when referring to \( A \subseteq T \) in the evaluation of the competitive ratio, we imply that every task in \( A \) also belongs to \( T \), as the tasks in \( T \) are distinct. Moreover, the expectation \( E_{A \subseteq T}[\cdot] \) is determined by the variability of the set \( A \) of \( k \) requested tasks. For convenience of notation, the subscript \( A \subseteq T \) is omitted in the rest of this paper, and this expectation is expressed as \( E[\cdot] \).

Ultimately, we strive for HERALD* to be computationally efficient, and we define this objective as follows.

**Definition 6.** An incentive mechanism is computationally efficient if it can be executed within polynomial time.

In essence, our aims are to guarantee that the proposed mechanism is both truthful and individually rational, as well as being differentially private, while also having a small social cost and computational complexity.

**IV. Incentive Mechanism for Uncertain Tasks**

In this section, we introduce an incentive mechanism designed for uncertain tasks, and we demonstrate that our mechanism satisfies the properties of truthfulness, individual rationality, and differential privacy. Additionally, we examine the competitive ratios on the expected social cost of the mechanism, as stated in Theorem \( \ref{thm:competitive_ratio} \). Furthermore, we evaluate the computational complexity of the mechanism, as presented in Proposition \( \ref{prop:computational_complexity} \) in addition to the aforementioned properties.

**A. Design Rationale**

The design of an incentive mechanism must take into account the possibility of tasks arriving simultaneously, as this can affect the mechanism’s construction. In the offline scenario, all tasks arrive simultaneously, and the platform has prior knowledge of all tasks. However, in the case of uncertain tasks, the number of tasks arriving simultaneously is also uncertain, and the probability distribution of the number of arriving tasks is different. To clarify this concept, consider the following straightforward example.

**Example 1.** In this example, the platform possesses a sensing task set \( T = \{\tau_1, \tau_2, \tau_3\} \) containing three tasks, each of which has a probability of \( \frac{1}{3} \) to arrive in the future. As a result, the task arrivals follow a uniform distribution. If a single task is set to arrive in the future, it could be \( \tau_1, \tau_2 \) or \( \tau_3 \) with the same probability \( \frac{1}{3} \). While, if two tasks arrive simultaneously in the future, they may be \( \{\tau_1, \tau_2\} \) and \( \{\tau_3, \tau_3\} \) with the same probability \( \frac{1}{9} \), and may be \( \{\tau_1, \tau_2\}, \{\tau_1, \tau_3\} \) and \( \{\tau_2, \tau_3\} \) with the same probability \( \frac{2}{9} \). Furthermore, if three tasks simultaneously arrive in the future, they may be \( \{\tau_1, \tau_2, \tau_3\} \) with probability \( \frac{1}{9} \); \( \{\tau_1, \tau_1, \tau_1\} \); \( \{\tau_2, \tau_2, \tau_2\} \); \( \{\tau_3, \tau_3, \tau_3\} \) with the same probability \( \frac{2}{9} \), and may be \( \{\tau_1, \tau_2, \tau_3\}, \{\tau_1, \tau_3, \tau_3\}, \{\tau_1, \tau_1, \tau_2\}, \{\tau_1, \tau_1, \tau_3\}, \{\tau_2, \tau_2, \tau_2\}, \{\tau_2, \tau_3, \tau_3\}, \{\tau_3, \tau_3, \tau_3\} \) with the same probability \( \frac{1}{9} \).

As demonstrated in the example, our approach differs from existing works. Rather than assuming a fixed number of tasks arriving simultaneously, we propose HERALD*, an adaptive incentive mechanism that adjusts to varying numbers of tasks as they arrive. The first phase of HERALD* involves matching workers to all possible task subsets, before the actual task requests are received. Subsequently, an estimated number of tasks arriving simultaneously is inputted into HERALD*. Based on the different input numbers, HERALD* will produce varying outcomes for both winning selection and payment determination.

**B. Design Details**

Within this section, we shall furnish an in-depth account of the operational mechanics of HERALD*. This mechanism consists of three phases: the task-worker matching phase (Alg. \( \ref{alg:task-worker_matching} \)), the winning selection phase (Alg. \( \ref{alg:winning_selection} \)), and the payment determination phase (Alg. \( \ref{alg:payment_determination} \)).

**Task-Worker Matching Phase:** Before the real requested tasks arrive, the platform first matches workers for all task subsets. We refer to this as the task-worker matching phase.
Algorithm 1 HERALD*: Task-Worker Matching

Input: The task set $\mathcal{T}$, the set of task subsets $\mathcal{Y} = \{\Gamma_1, \Gamma_2, \ldots, \Gamma_l\}$ such that $\bigcup_{j=1}^{l} \Gamma_j = \mathcal{T}$, worker set $\mathcal{W}$, each worker’s bid, a score function $u$ and its sensitivity $\Delta u$, a small constant $\epsilon$.

Output: Matching set $\mathcal{P}$.

1: $\mathcal{P} \leftarrow \emptyset$
2: for each worker $i \in \mathcal{W}$ do
3:   Calculate the probability $Pr(b_i)$ that worker $i$ with her unique bid $b_i$ is matched with each task subset $\Gamma_j \in \mathcal{Y}$: $Pr(b_i) \propto \exp\left(\frac{u(b_i)}{\Delta u}\right)$
4: end for
5: for each task subset $\Gamma_j \in \mathcal{Y}$ do
6:   Select the worker $i$ randomly according to the computed probability distribution to match the task subset $\Gamma_j$
7:   $\mathcal{P} \cup \{(\Gamma_j, b_i)\}$
8: end for
9: return The matching set $\mathcal{P}$ of the task subset $\Gamma_j$ and its matched worker $i$’s bid.

As can be seen in Alg. 1 for each task subset $\Gamma_j \in \mathcal{Y}$, the probability that worker $i$ with the unique bid $b_i$ is matched is $Pr(b_i)$. In particular, as we introduced in definition 2, we employ the exponential mechanism to achieve the differential privacy of the bidding results. Thus we set $Pr(b_i) \propto \exp\left(\frac{u(b_i)}{\Delta u}\right)$, where $r \in R$ is the output and $u$ is a score function that maps input/output pairs to utility scores, $\Delta u = \max_{r \in R} \max_{x \neq y} |u(x, r) - u(y, r)|$ is the sensitivity of the utility function $u$, and $\epsilon$ is a small constant. For each round of matching, the platform eventually assigns the task subset to the worker randomly according to the computed probability distribution. To ensure task quality and prevent monopolization, we require that each task $\tau_i \in \mathcal{T}$ is matched by at least two different workers.

Upon completion of the task-worker matching phase, the platform proceeds to identify the winning set $\mathcal{S}$ and the corresponding payment $p_i$ for each winning worker $i$ based on the submitted matching set $\mathcal{P}$. We refer to these as the winning selection phase and payment determination phases, which is illustrated in Alg. 2 and Alg. 3 respectively.

Winning Selection Phase: In order to obtain the sensory data of the uncertain tasks, we define a selection threshold (ST) $T \geq 0$ that remains unchanged throughout the implementation process of HERALD*.

- **Type I Selection**: If there exist any task subsets with cost-effectiveness (CF) values equal to or lower than $\frac{T}{|\Gamma|}$, the platform will choose the worker who matches the task subset with the lowest CF as the winner.
- **Type II Selection**: If the CFs of all task subsets exceed $\frac{T}{|\Gamma|}$, the platform will choose a winning worker from among those matched to all task subsets. This worker must possess the lowest bid and have a task subset that includes at least one unassigned task in $\mathcal{T}$.

Subsequently, the matching pair that was selected as per the aforementioned criteria are incorporated into the winning set $\mathcal{S}$. The platform then proceeds to calculate the payment for each winner based on the members of the winning set.

Payment Determination Phase: For each winning pair $(\Gamma_j, b_i) \in S$, the platform defines a copy set $T_j = \Gamma_j$ and builds a covering set $\mathcal{W}_i = \{\ell | \ell \in \mathcal{W}_{\{i\}}, \Gamma_i \cap T_j \neq \emptyset\}$. It then derives a replaced set denoted as $\mathcal{R}_i$ consisting of workers

Algorithm 2 HERALD*: Winning Selection

Input: The task set $\mathcal{T}$, matching set $\mathcal{P}$, each worker’s bid, the number $k$ of tasks arriving simultaneously.

Output: The winning set $\mathcal{S}$.

1: $\mathcal{S} \leftarrow \emptyset$
2: Calculate the selection threshold $T = 64\mathbb{E}[C_{OPT}(A, W)]$, where $A$ is the set of $k$ possibly simultaneously arriving tasks from the sensing task set $\mathcal{T}$
3: while $T \neq \emptyset$ do
4:   for each matching pair $(\Gamma_j, b_i) \in \mathcal{P}$ do
5:      Calculate the cost effectiveness (CF) $\frac{b_i}{|\Gamma_i \cap \Gamma_j|}$, where $b_i$ is the bid of worker $i$ for executing the task subset $\Gamma_j$
6:   end for
7:   Type I Selection:
8:      if $\exists (\Gamma_j, b_i) \in \mathcal{P}$, s.t $\frac{b_i}{|\Gamma_i \cap \Gamma_j|} \leq \frac{T}{|\mathcal{T}|}$ then
9:         Among the task subsets whose CFs are less than $\frac{T}{|\mathcal{T}|}$, the worker $i \in \mathcal{W}$ that matches the task subset $\Gamma_j$ with the minimum CF value is selected as the winner
10:      else
11:         Among the workers matched by all task subsets, the worker $i \in \mathcal{W}$ whose bid is the lowest and whose task subset $\Gamma_j$ she matched contains at least one undiscovered task is selected as the winner
12: end if
13: $\mathcal{S} \leftarrow \mathcal{S} \cup \{(\Gamma_j, b_i)\}$
14: $\mathcal{P} \leftarrow \mathcal{P}\{\{(\Gamma_j, b_i)\}\}$
15: $\mathcal{T} \leftarrow \mathcal{T}\{\Gamma_j\}$
16: end while
17: return $\mathcal{S}$.

Algorithm 3 HERALD*: Payment Determination

Input: The worker set $\mathcal{W}$, winning set $\mathcal{S}$, each worker’s bid.

Output: The payment $\mathcal{\overline{p}}$.

1: for each winning pair $(\Gamma_j, b_i) \in \mathcal{S}$ do
2:   Define a copy set $T_j \leftarrow \Gamma_j$
3:   Build a covering set $\mathcal{W}_i = \{\ell | \ell \in \mathcal{W}_{\{i\}}, \Gamma_i \cap T_j \neq \emptyset\}$
4:   Define a replaced set $\mathcal{R}_i \leftarrow \emptyset$
5: while $T_j \neq \emptyset$ do
6:   Choose a worker $\ell \in \mathcal{W}_i$ whose matched task subset has the minimum CF
7:   $\mathcal{R}_i \leftarrow \mathcal{R}_i \{\ell\}$
8:   $T_j \leftarrow T_j\{\ell\}$
9: end while
10: $p_i = p_i + \max\{b_i, p_{\mathcal{R}_i}\}$ for $p_{\mathcal{R}_i} = \sum_{\ell \in \mathcal{R}_i} b_{\ell}$
11: end for
12: return $\mathcal{\overline{p}}$. 
in $\mathcal{W}$, with the least CFs in each iteration such that $\Gamma_j \subseteq \cup_{t \in R}, \Gamma_t$. The payment to winner $i$ is $p_i = p_1 + \max\{b_i, p_R\}$, where $p_R = \sum_{t \in R} b_t$.

**Example 2.** In this example, the platform has a task set $\mathcal{T} = \{\tau_1, \tau_2, \tau_3, \tau_4, \tau_5\}$ with five tasks and divides them into seven task subsets $\Gamma_1 = \{\tau_1, \tau_2\}$, $\Gamma_2 = \{\tau_3, \tau_5\}$, $\Gamma_3 = \{\tau_1, \tau_3, \tau_4\}$, $\Gamma_4 = \{\tau_4, \tau_5\}$, $\Gamma_5 = \{\tau_1\}$, $\Gamma_6 = \{\tau_2, \tau_5\}$ and $\Gamma_7 = \{\tau_2, \tau_4, \tau_5\}$. There are seven workers whose costs are $c_1 = 1.4, c_2 = 1.8, c_3 = 2.8, c_4 = 2.6, c_5 = 3.1, c_6 = 3.3$ and $c_7 = 3.6$. As HERALD* is truthful, which will be demonstrated later, it follows that $b_i = c_i$. Before the arrival of the actually requested tasks, the platform first matches workers for all possible task subsets. We refer to this as the task-worker matching phase which is shown in Alg. 2. For each task subset $\Gamma_j \in \mathcal{Y}$, the probability that worker $i$ with the unique bid $b_i$ is matched is $P_r(b_i)$. For each round of matching, the platform eventually assigns the task subset to the worker randomly according to the computed probability distribution. To prevent any monopolistic behavior and ensure the quality of the sensing task, we require that each task $\tau_j \in \mathcal{T}$ be matched by at least two different people. We assume that the matching set $P = \{(\Gamma_1, b_1), (\Gamma_2, b_2), (\Gamma_3, b_3), (\Gamma_4, b_4), (\Gamma_5, b_5), (\Gamma_6, b_6), (\Gamma_7, b_7)\}$.

We make the assumption that the task arrival follows a Poisson distribution with arrival rate $\lambda$, and the tasks arrive sequentially in a sequential manner in order to derive the set of winning entries and their respective payment amounts.

**Remark 1.** It is apparent that when the input parameter is set to the total number of tasks $n$ in the platform’s task set, HERALD* exhibits a probability of $\frac{A_n}{n^2}$, which is reduced to an offline incentive mechanism. Several prior studies have explored offline scenarios where the platform has complete knowledge of the task information. Hence, this implies that HERALD* can be employed in a wider range of scenarios than the conventional offline incentive mechanisms.

C. Design of Score Functions

To apply the exponential mechanism to achieve the differential privacy of bidding results, it is necessary to devise score functions. Two score functions, namely a linear score function, and a logarithmic score function are created for this purpose. We will show that they have theoretical bounds on differential privacy and produce different impacts in simulations.

**Linear score function:** $f_{lin}(x) = -x$. For any worker $i \in \mathcal{W}$, the probability that worker $i$ with bid $b_i$ is matched with any task subset $\Gamma_j \in \mathcal{Y}$ is

$$P_r(b_i) \propto \begin{cases} \exp\left(-\frac{c_i}{b_{\max} - b_{\min}}\right), & \text{if } i \in \mathcal{W}, \\ \exp\left(-\frac{c_i}{b_{\max} - b_{\min}}\right), & \text{otherwise.} \end{cases}$$

(3)

Since $u = f_{lin}(x) = -x$, we have

$$\Delta u = \max_{x \in R_{max} \times y, ||x - y||_1 \leq 1} |u(x, x) - u(y, y)| = \frac{b_{\max} - b_{\min}}{b_{\max}}.$$ 

(4)

In order to guarantee that the score function’s value is non-negative, we apply the following normalization process.

$$P_r(b_i) \propto \begin{cases} \frac{\exp\left(-\frac{c_i}{b_{\max} - b_{\min}}\right)}{\sum_{j \in \mathcal{W}} \exp\left(-\frac{c_j}{b_{\max} - b_{\min}}\right)}, & \text{if } i \in \mathcal{W}, \\ 0, & \text{otherwise.} \end{cases}$$

(5)

**Logarithmic score function:** $f_{ln}(x) = -\ln(x)$. For any worker $i \in \mathcal{W}$, the probability that worker $i$ with bid $b_i$ is matched with any task subset $\Gamma_j \in \mathcal{Y}$ is

$$P_r(b_i) \propto \begin{cases} \exp\left(-\frac{c_i}{b_{\max}}\right), & \text{if } i \in \mathcal{W}, \\ 0, & \text{otherwise.} \end{cases}$$

(6)

Since $u = f_{ln}(x) = -\ln(x)$, we have

$$\Delta u = \max_{x \in R_{max} \times y, ||x - y||_1 \leq 1} |u(x, x) - u(y, y)| = \ln\frac{b_{\max} - b_{\min}}{b_{\max}} = -\ln \frac{b_{\min}}{b_{\max}},$$

(7)

where $b_{\min}$ is normalized to 1 and $b_{\max}$ is a constant. We also need to normalize the score function to ensure that its value is non-negative.

$$P_r(b_i) \propto \begin{cases} \frac{\exp\left(-\frac{c_i}{b_{\max}}\right)}{\sum_{j \in \mathcal{W}} \exp\left(-\frac{c_j}{b_{\max}}\right)}, & \text{if } i \in \mathcal{W}, \\ 0, & \text{otherwise.} \end{cases}$$

(8)

D. Analysis

This subsection will provide evidence that HERALD* conforms to the characteristics outlined in Section III-B.

**Theorem 2 ([33]).** A mechanism satisfies truthfulness only if the following conditions are met:

1. The selection rule is monotonic: If a matching pair wins by offering a bid of $b_i$, it will also win if it bids $b'_j \leq b_i$;
2) Each winning pair is compensated with the critical value: A matching pair will not win if it bids higher than this value.

**Theorem 3.** HERALD* is truthful.

**Proof.** To prove the truthfulness of HERALD*, we will demonstrate its adherence to the criteria outlined in Theorem 2. 

**Monotonicity:** Given a matching pair \((\Gamma_j, b_i)\), we will prove that if it wins with a bid of \(b_i\), it will also win with a bid of \(b_i' \leq b_i\). We will demonstrate this in the following two scenarios.

**Case 1:** During a winning selection phase iteration, if the CF of the winning pair \((\Gamma_j, b_i)\) satisfies \(\frac{b_i}{\Gamma_j} \geq \frac{\tilde{T}}{T_j}\), then it implies that it possesses the smallest CF among all the matching pairs. Consequently, it will also win with a bid of \(b_i' \leq b_i\).

**Case 2:** During an iteration, if the CF of the winning pair \((\Gamma_j, b_i)\) satisfies \(\frac{b_i}{\Gamma_j} > \frac{T_j}{T_j}\), then it indicates that it possesses the least cost among the matching pairs and that there exists no matching pair \((\Gamma_j, b_j)\) such that \(\frac{b_j}{\Gamma_j} \leq \frac{T_j}{T_j}\). We must subsequently examine two sub-cases.

- **Subcase 2.1:** If the bid \(b_i' \leq b_i\) satisfies \(\frac{b_i'}{\Gamma_j} > \frac{T_j}{T_j}\), then it will also win with a bid of \(b_i'\), as \(b_i'\) is the lowest and there is no matching pair \((\Gamma_j, b_j)\) such that \(\frac{b_j}{\Gamma_j} \leq \frac{T_j}{T_j}\).

- **Subcase 2.2:** If the bid \(b_i' \geq b_i\) satisfies \(\frac{b_i'}{\Gamma_j} \leq \frac{T_j}{T_j}\), then it will also win by bidding \(b_i'\), as it is the sole matching pair with a CF that is less than or equal to \(\frac{T_j}{T_j}\).

**Critical Value:** If a matching pair \((\Gamma_j, b_i)\) wins, it follows that worker \(i\)'s payment is \(p_i = p_i + \max\{b_i, p_{\Gamma_j}\}\), where \(p_{\Gamma_j} = \sum_{\ell \in \Gamma_j} b_{\ell}\). If worker \(i\) increases her bid to \(b_i'\) such that \(b_i' \geq p_{\Gamma_j}\), her payments remain constant. However, if \(b_i' > p_{\Gamma_j}\), we must examine the following two cases during each iteration of the winning selection phase.

- **Case 1:** When CF of matching pair \((\Gamma_j, b_i)\) satisfies \(\frac{b_i}{\Gamma_j} \leq \frac{T_j}{T_j}\), we will prove that there is a matching pair \((\Gamma_{\ell}, b_{\ell})\), where \(\ell \in \Gamma_i\), such that \(b_{\ell} \geq p_{\Gamma_{\ell}}\), and \(b_i \geq b_{\ell}\), where \(p_{\Gamma_{\ell}} = \sum_{\ell \in \Gamma_{\ell}} b_{\ell}\).

- **Case 2:** When CF of matching pair \((\Gamma_j, b_i)\) satisfies \(\frac{b_i}{\Gamma_j} \geq \frac{T_j}{T_j}\), we need to consider two subcases.

  - **Subcase 2.1:** Once there exist some matching pairs \((\Gamma_q, b_{\ell})\) for \(\Gamma_q \in Y\) such that \(\frac{b_{\ell}}{\Gamma_q} \leq \frac{T_j}{T_j}\), the platform will select a matching pair \((\Gamma_q, b_{\ell})\) instead of \((\Gamma_j, b_i)\).

  - **Subcase 2.2:** Once the CFs of all matching pairs \((\Gamma_q, b_{\ell})\) for \(\Gamma_q \in Y\) and \(\ell \in \Gamma_i\) satisfies \(\frac{b_{\ell}}{\Gamma_q} > \frac{T_j}{T_j}\), the platform will always find a matching pair \((\Gamma_q, b_{\ell})\) with the minimum bid \(b_{\ell}\) such that \(b_{\ell} \leq p_{\Gamma_{\ell}} \leq \tilde{b}_{\ell}\), which means that the platform will not select the matching pair \((\Gamma_j, b_i)\).

Therefore, the conclusion holds.

**Lemma 1.** HERALD* is individually rational.

**Proof.** Theorem 3 demonstrates that each worker bids their actual cost \(c_i\). The individual rationality of HERALD* is ensured by the fact that the payment made to each winner \(i\) is equal to \(p_i = p_i + \max\{b_i, p_{\Gamma_j}\} \geq b_i = c_i\).

Besides proving truthfulness and individual rationality, we also demonstrate that HERALD* preserves the intended differential privacy of the bidding outcomes. Initially, we examine the impact of the linear score function on the differential privacy outcomes.

**Theorem 4.** For any constant \(\epsilon > 0\), HERALD* with the linear score function preserves \(\frac{\epsilon}{2}\)-differential privacy, where \(\epsilon > 0\) is a constant and \(l\) is the number of task subsets.

**Proof.** Consider two input bid profiles \(b\) and \(b'\) that differ in only one bid. Let \(M(b)\) and \(M(b')\) denote the task-worker matching results by HERALD* with inputs \(b\) and \(b'\), respectively. We aim to prove that HERALD* achieves differential privacy for an arbitrary sequence of task-worker matching results \(Y = \{\Gamma_1, b_1, \Gamma_2, b_2, \ldots, \Gamma_l, b_l\}\) of length \(l\) for \(m\) workers, where each worker can match \(k\) task subsets with \(0 \leq k \leq l\). To analyze the relative probability of HERALD* for the given bid inputs \(b\) and \(b'\), we consider:

\[
P_r[M(b) = \Gamma_j] = \sum_{j=1}^{l} \exp(-\frac{e}{2(b_{\max} - b_{\min})}) \times \prod_{j=1}^{l} \exp(-\frac{e}{2(b_{\max} - b_{\min})})
\]

where equation (a) is derived by formula (5) and \(b_j, b_j'\) denote the bids of workers matching the task subsets \(\Gamma_j, \Gamma_j'\), respectively. Then, we prove this theorem in two cases. When \(b_j > b_j'\), the value of the first product is at most 1, we have

\[
P_r[M(b) = \Gamma_j] \leq \sum_{j=1}^{l} \exp(-\frac{e(b_j - b_j')}{2(b_{\max} - b_{\min})})
\]

where equation (b) is derived by formula (5).
where the inequality (a) holds because for all \( x \leq 1, e^x \leq 1 + (e - 1)x \). The inequality (b) holds because for all \( x \in R, 1 + x \leq e^x \). In Section III-A, it was mentioned that the bid \( b_i \) of every worker \( i \) is confined within the interval \([b_{\text{min}}, b_{\text{max}}]\), where \( b_{\text{min}} \) is normalized to 1 and \( b_{\text{max}} \) is a fixed constant such that inequality (c) is satisfied.

When \( b_k \leq b_k' \), the value of the second product is at most 1, we have

\[
\frac{\Pr[M(b) = I]}{\Pr[M(b') = I]} \leq \prod_{j=1}^{l} \exp(-\frac{\epsilon (b_j' - b_j)}{2(b_{\text{max}} - b_{\text{min}})})
\]

\[
= \prod_{j=1}^{l} \exp\left(\frac{\epsilon}{2(b_{\text{max}} - b_{\text{min}})} \sum_{j=1}^{l} (b_j' - b_j)\right).
\]

(a) \leq \exp\left(\frac{\epsilon}{2(b_{\text{max}} - b_{\text{min}})} \times l(b_{\text{max}} - b_{\text{min}})\right).

in which the inequality (a) holds because of the same reason of inequality (c) in formula (10). Combining the formulas (10) and (11), the proof is completed. 

Next, we analyze the effect of the logarithmic score function on the differential privacy results.

**Theorem 5.** For any constant \( \epsilon > 0 \), HERALD* with the logarithmic score function preserves \( \frac{1}{l} \)-differential privacy, where \( \epsilon > 0 \) is a constant and \( l \) is the number of task subsets.

**Proof.** The proof is analogous to that of the linear score function, and thus we will not repeat the default setting.

Instead, we examine the HERALD* relative probability for a particular input bid profile \( b \) and its perturbed version \( b' \):

\[
\frac{\Pr[M(b') = I]}{\Pr[M(b) = I]} \leq \prod_{j=1}^{l} \exp\left(\frac{-\epsilon \ln \frac{b_j'}{b_{\text{max}}}}{2 \ln b_{\text{max}}}\right)
\]

\[
= \prod_{j=1}^{l} \exp\left(\frac{-\epsilon \ln \frac{b_j'}{b_{\text{max}}}}{2 \ln b_{\text{max}}}\right) \times \prod_{j=1}^{l} \exp\left(\frac{-\epsilon \ln \frac{b_j}{b_{\text{max}}}}{2 \ln b_{\text{max}}}\right),
\]

where equation (a) is derived by formula (8). Then, we prove this theorem in two cases. When \( b_k > b_k' \), the value of the first product is at most 1, we have

\[
\frac{\Pr[M(b') = I]}{\Pr[M(b) = I]} \leq \prod_{j=1}^{l} \exp\left(\frac{-\epsilon \ln \frac{b_j'}{b_{\text{max}}}}{2 \ln b_{\text{max}}}\right)
\]

\[
= \prod_{j=1}^{l} \exp\left(\frac{-\epsilon \ln \frac{b_j}{b_{\text{max}}}}{2 \ln b_{\text{max}}}\right)
\]

\[\leq \exp\left(\frac{\epsilon}{2 \ln b_{\text{max}}} \times \sum_{j=1}^{l} \ln \frac{b_j}{b_{\text{max}}}\right)
\]

\[= \exp\left(\frac{\epsilon}{2 \ln b_{\text{max}}} \times \frac{l}{b_{\text{max}}}\right), \quad \text{(13)}
\]

and

\[
\frac{\Pr[M(b') = I]}{\Pr[M(b) = I]} \leq \prod_{j=1}^{l} \exp\left(\frac{-\epsilon \ln \frac{b_j'}{b_{\text{max}}}}{2 \ln b_{\text{max}}}\right)
\]

\[
= \prod_{j=1}^{l} \exp\left(\frac{-\epsilon \ln \frac{b_j}{b_{\text{max}}}}{2 \ln b_{\text{max}}}\right)
\]

\[\leq \exp\left(\frac{\epsilon}{2 \ln b_{\text{max}}} \times \sum_{j=1}^{l} \ln \frac{b_j}{b_{\text{max}}}\right)
\]

\[= \exp\left(\frac{\epsilon}{2 \ln b_{\text{max}}} \times \frac{l}{b_{\text{max}}}\right), \quad \text{(14)}
\]

in which the inequality (a) holds because of the same reason of inequality (c) in formula (10). Combining the formulas (13) and (14), the proof is completed. 

In addition to its truthfulness, individual rationality, and differential privacy, HERALD* exhibits a low level of computational complexity, as can be observed.

**Proposition 1.** The computational complexity of the HERALD* is \( O(lm + ln) \).
Proof. In order to determine the computational complexity of HERALD*, we must analyze the task-worker matching phase (as outlined in Alg. 1), the winning selection phase (in Alg. 2), and the payment determination phase (as described in Alg. 3), separately.

1) task-worker Matching Phase: In Alg. 1 the first loop (Lines 2–3) calculates the probability distribution in $m$ iterations, and the second loop (Lines 4–6) completes the task-worker matching in $l$ iterations. Therefore, the computational complexity of the task-worker matching phase is $O(l)$.

2) Winning Selection Phase: Likewise, in Alg. 2 the primary loop (Lines 3–12) of the winning selection phase concludes in the most unfavorable scenario after $n$ rounds. During each round, type I selection (Lines 6–7) is executed $l$ times to identify the worker with the least bidding price effectiveness, or type II selection (Lines 8–9) is conducted to determine the worker with the minimum bidding price. Consequently, the computational complexity of the winning selection phase is $O(ln)$.

3) Payment Determination Phase: Similar to the previous two, in Alg. 3 the primary loop (Lines 1–10) of the payment determination phase concludes at worst after $l$ iterations. Each iteration involves constructing a covering set in $m$ iterations (Line 3) and constructing a replaced set in $n$ iterations (Lines 5–9). Consequently, the computational complexity of the payment determination phase is $O(ln + n)$.

By merging the task-worker matching phase, winning selection phase, and payment determination phase, HERALD* has a computational complexity of $O(ln + n)$. We have specified in Section II–A that $l < mn$.

In the following parts, we will demonstrate the competitive ratio of expected social cost attained by HERALD* when the tasks in $T$ arrive according to a uniform distribution. Specifically, we let $T = 64\mathbb{E}[C_{OPT}(\mathcal{A}, W)]$, where $\mathcal{A} \subseteq T$ is a subset of $k$ tasks that could potentially arrive concurrently from the task set $T$. In order to calculate the competitive ratio of the expected social cost of HERALD*, we analyze the costs of type I selection and type II selection independently.

**Lemma 2.** HERALD* achieves a competitive ratio of $O(ln n)$ on the expected social cost through type I selection when the tasks in the task set $T$ are distributed uniformly.

Proof. Assume that HERALD* selects workers for the winning set $S$ using type I selection in the following order: $S_I = \{1, \ldots, h\}$. Let $\tilde{T}_i$ denote the set of tasks whose sensory data has not been collected just before worker $i$ is selected. Since HERALD* carries out type I selection, $c_i \leq \frac{1}{|T_i|} \mathbb{E}[C_{OPT}(\mathcal{A}, W)]$, where $\mathcal{A}$ is a subset of $k$ tasks possibly arriving simultaneously from $T_i$. Hence, the social cost of workers in $S_I$ can be bounded by

$$\sum_{i \in S_I} c_i \leq \sum_{i \in S_I} \frac{64\mathbb{E}[C_{OPT}(\mathcal{A}, W)]}{|\tilde{T}_i|} \leq 64\mathbb{E}[C_{OPT}(\mathcal{A}, W)] \sum_{i=1}^h \frac{1}{|\tilde{T}_i|},$$

which is at most $64\mathbb{E}[C_{OPT}(\mathcal{A}, W)] \ln n$, in which $\Gamma_i$ denotes the worker $i$’s matching task subset. Therefore, the conclusion holds due to the property of expectation.

We still need to establish a bound on the expected social cost of workers in the winning set $S$ chosen by HERALD* via type II selection. To accomplish this, we must introduce the following notations.

We define $S_{II} = \{1, \ldots, \ell\}$ as the group of workers in the winning set $S$ selected by HERALD* via type II selection in the specified sequence. Also, we define $\tilde{T}_i$ as the set of tasks whose sensory data is not gathered right before worker $i$ is selected. Additionally, $n_i = |\tilde{T}_i|$ represents the number of tasks in $\tilde{T}_i$, and $k_i = n_i \frac{k}{n}$ is the expected count of requested tasks arriving from $\tilde{T}_i$. We use $\mathcal{A}_i$ to denote the subset of $\mathcal{A}$ that contains requested tasks belonging only to $\tilde{T}_i$. Furthermore, we define $S^*(\mathcal{A}, W)$ as the winning set with the least social cost for any $\mathcal{A}$ set. Then, $\mathcal{S}^*(\mathcal{A}, W)$ be the subset of $S^*(\mathcal{A}, W)$ such that for each task $\tau_j \in \mathcal{A}_i$, the worker in $S^*(\mathcal{A}, W)$ has the corresponding task subset containing task $\tau_j$ and has the least cost among workers in $S^*(\mathcal{A}, W)$.

**Lemma 3.** When the arrivals of tasks in the task set $T$ follow a uniform distribution, HERALD* achieves a competitive ratio of $O(ln n)$ on the expected social cost through type II selection.

Proof. Let us recall that the set of workers with the bids in the winning set $S$ selected by HERALD* via type II selection is denoted by $S_{II} = \{1, \ldots, \ell\}$. Set $k_{\ell+1} = 0$ and $c_0 = 0$ for notational convenience. Let $j$ be $k_j \geq 8\ln 2n$ but $k_j < 8\ln 2n$. Then, we observe that there are at most $8\ln 2n$ tasks from $\tilde{T}_{j+1}$ in expectation. As each of these tasks is executed by a worker whose cost is not greater than the one performing it in $S^*(\mathcal{A}, W)$, the cost incurred by workers $j + 1, \ldots, \ell$ is bounded by $8\ln 2n \mathbb{E}[C_{OPT}(\mathcal{A}, W)]$. Therefore, the expected cost incurred by using the remaining workers $1, \ldots, j$, satisfies

$$\sum_{i=1}^j c_i \mathbb{P}[A \cap (\Gamma_i \cap \tilde{T}_i) \neq \emptyset] \leq \sum_{i=1}^j c_i \mathbb{E}[|A \cap (\Gamma_i \cap \tilde{T}_i)|]$$

$$\sum_{i=1}^j c_i \mathbb{E}[|A \cap (\Gamma_i \cap \tilde{T}_i)|] \leq \sum_{i=1}^j c_i \mathbb{E}[|A \cap (\tilde{T}_i \setminus \tilde{T}_{i+1})|]$$

$$\sum_{i=1}^j c_i (k_i - k_{i+1}) \leq \sum_{i=1}^j \sum_{i=1}^n (c_i - c_{i-1})$$

$$\leq \sum_{i=1}^j 16\mathbb{E}[|S^*(\mathcal{A}_i, W)|] \ln (c_i - c_{i-1})$$

$$= 16 \ln l \left(c_j \mathbb{E}[S^*(\mathcal{A}_j, W)]\right)$$

$$+ \sum_{i=1}^j \left(c_i \left(|S^*(\mathcal{A}_i, W)| - |S^*(\mathcal{A}_{i+1}, W)|\right)\right)$$

$$\leq 16 \ln l \left(|\mathbb{E}[C(S^*(\mathcal{A}_j, W))]|\right)$$

$$+ \sum_{i=1}^j \left(|\mathbb{E}[C(S^*(\mathcal{A}_i, W))]| - |\mathbb{E}[C(S^*(\mathcal{A}_{i+1}, W))]|\right)$$

$$\leq 16 \ln l |\mathbb{E}[C_{OPT}(\mathcal{A}, W)]|,$$
Therefore, the expected cost incurred by workers 1, . . . , ℓ can be bounded as follows:

\[
\sum_{i=1}^{\ell} c_i \Pr[A \cap (\Gamma_i \cap \tilde{T}_i) \neq \emptyset] \leq 8 \ln 2n + 16 \ln \ell \cdot \text{E}[C_{OPT}(A, W)]. \quad (17)
\]

Then we have

\[
\frac{\sum_{i=1}^{\ell} c_i \Pr[A \cap (\Gamma_i \cap \tilde{T}_i) \neq \emptyset]}{\text{E}[C_{OPT}(A, W)]} \leq O(\ln \ln n). \quad (18)
\]

The above is based on one matching set \( P \) among all the matching results \( F \), we need to take the expectation over all situations, i.e.,

\[
\max_{k \in \{1, \ldots, n\}} \frac{\text{E}_{A \subseteq T} \left[ C(S(A, W)) \right]}{\text{E}_{A \subseteq T} \left[ C_{OPT}(A, W) \right]} = \sum_{j=1}^{m_j} P_j \times \frac{\sum_{i=1}^{\ell} c_i \Pr[A \cap (\Gamma_i \cap \tilde{T}_i) \neq \emptyset]}{\text{E}[C_{OPT}(A, W)]} \leq \sum_{j=1}^{m_j} P_j \times O(\ln \ln n) = O(\ln \ln n),
\]

where \( P_j \) is the probability of each task-worker matching result. This proof is completed. □

By combining Lemma 2 and Lemma 3, the following theorem is established.

**Theorem 6.** HERALD* achieves a competitive ratio of \( O(\ln \ln n) \) on expected social cost when the arrivals of tasks in the task set \( T \) follow a uniform distribution.

Based on the results from Theorem 6 we can deduce that HERALD* yields a low expected social cost. Therefore, HERALD* can be applied to many scenarios with uncertain sensing tasks.

### V. Performance Evaluation

The subsequent section will showcase the benchmark methods utilized for evaluating the performance of HERALD* and expound on the simulation settings employed in the experiment. Furthermore, the outcomes of the simulation will be presented.

#### A. Baseline Methods

In this simulation, we compare the following two incentive mechanisms with HERALD*.

**Cost-effectiveNess greedy auction (CONE):** When it comes to uncertain tasks, the platform is aware that the tasks belonging to set \( T \) are expected to arrive in the future with a probability distribution. To obtain sensory data for these tasks, the platform computes the CF for each matching pair and picks the winning pair, denoted by \((\Gamma_j, b_i)\), which has the lowest CF \( b_i \) value among those of matching pairs in each iteration. The platform then proceeds to acquire the sensory data from worker \( i \).

**Cost greedy auction (COSY):** When it comes to uncertain tasks, the platform acquires sensory data by examining the bids of matching pairs and determining the winning pair, denoted by \((\Gamma_j, b_i)\). This pair has the lowest bid value \( b_i \) among those of winning pairs and corresponds to a task subset \( \Gamma_j \) that includes at least one task that has not yet been covered in the current iteration. After selecting the winning pair, the platform proceeds to collect the sensory data from worker \( i \).

The payment determination process in both CONE and COSY is identical to that of HERALD*. It is evident that both CONE and COSY are truthful and individually rational.

#### B. Simulation Settings

In Table I, we present the assessment metrics for various scenarios, which include the cost \( c_i \) that worker \( i \) performs the assigned subset \( \Gamma_j \) of matching tasks, along with the number of tasks in that subset denoted as \([\Gamma_j]\). Moreover, the table also includes the number of workers in worker set \( W \) denoted by \( m \) and the number of sensing tasks in the task set \( T \) represented by \( n \).

We present an assessment of HERALD*, which highlights how the expected social cost and expected total payment are affected by the number of workers and sensing tasks. Specifically, we conduct two evaluations to analyze the impact of the worker set and task set on the system. In the first evaluation, labeled as setting I, we fix the number \( n \) of sensing tasks at 120 and vary the number \( m \) of workers from 60 to 150 in steps of 5 to evaluate the impact of \( m \). In the subsequent evaluation, referred to as setting II, we maintain the number \( m \) of workers at 80 while incrementally varying the number \( n \) of sensing tasks from 80 to 160 in steps of 5 to assess the effect of \( n \). In both scenarios, we randomly and independently sample the cost \( c_i \) of worker \( i \) and the number of tasks in task subset \( \Gamma_j \) from uniform distributions within the intervals \([1, 5]\) and \([15, 20]\), respectively.

We will now examine how worker costs affect the expected social cost and expected total payment yielded by HERALD*. To investigate the impact of worker costs \( c_i \), we consider three different intervals: \([1, 5]\), \([5, 10]\) and \([10, 15]\), in setting III. In this setting, we also randomly sample the number \( |\Gamma_j| \) of tasks in the subset from the interval \([15, 20]\). Additionally, we set the number \( n \) of sensing tasks to 120, while varying the number \( m \) of workers from 60 to 150.

In conclusion, we analyze the impact of the number of tasks in task subsets on the expected social cost and expected total payment derived by HERALD*. In particular, we consider three different intervals for the number of tasks \(|\Gamma_j|\) in each matching task subset, which is \([10, 15]\), \([15, 20]\) and \([20, 25]\) in setting IV. Meanwhile, we sample the cost \( c_i \) of each worker from the interval \([1, 5]\). Additionally, in setting IV, we fix the number \( n \) of sensing tasks to 120, while varying the number \( m \) of workers from 60 to 150.

| Settings | individual cost \( c_i \) | number \( |\Gamma_j| \) of each task subset | number \( m \) of workers | number \( n \) of sensing tasks |
|----------|-----------------|---------------------------------|----------------|---------------------|
| I        | \([1, 5]\)       | \([15, 20]\)                     | \([60, 150]\)  | 120                 |
| II       | \([1, 5]\)       | \([15, 20]\)                     | 80             | \([80, 160]\)      |
| III      | \([1, 5],[5, 10]\), \([10, 15]\)     | \([15, 20]\)                     | \([60, 150]\)  | 120                 |
| IV       | \([1, 5]\)       | \([10, 15],[15, 20],[20, 25]\)   | \([60, 150]\)  | 120                 |
C. Simulation Results

In Fig. 3 we analyze the effect of the number of workers. Specifically, Fig. 3(a) and Fig. 3(b) illustrate the impact on the expected social cost and expected total payment obtained by HERALD*. It is observed that HERALD* performs better than CONE and COSY. Interestingly, the expected social cost and expected total payment obtained by HERALD* using the logarithmic score function is lower than those obtained using the linear score function. This is due to the logarithmic score function giving higher chances of selection to users with low bids, resulting in a preference for such users.

Fig. 4 examines the impact of the number of tasks for uncertain tasks. (b). Expected total payment versus different numbers of sensing tasks for uncertain tasks.

Fig. 5. (a). The impact of worker’s cost on the expected social cost obtained by HERALD* for uncertain tasks with the liner score function. (b). The impact of worker’s cost on the expected total payment obtained by HERALD* for uncertain tasks with the liner score function.

Fig. 6. (a). The impact of worker’s cost on the expected social cost obtained by HERALD* for uncertain tasks with the logarithmic score function. (b). The impact of worker’s cost on the expected total payment obtained by HERALD* for uncertain tasks with the logarithmic score function.
expected total payment of HERALD* also increase due to the need for more workers to collect sensory data. Furthermore, consistent with the earlier results, the expected social cost and expected total payment of HERALD* with the logarithmic score function are lower than those of HERALD* with the linear score function for the same reasons.

Fig. 5 depicts the effect of workers’ cost on the performance of HERALD* under the linear score function. Specifically, Fig. 5(a) and Fig. 5(b) illustrate the impact of workers’ cost on the expected social cost and expected total payment generated by HERALD*. The results indicate that as the workers’ cost increases, both the expected social cost and expected total payment of HERALD* also increase. This is because a higher workers’ cost implies that more social cost is required for the same tasks, and the platform has to pay more to the workers compared to the scenario with a lower workers’ cost. Similar findings are observed for HERALD* under the logarithmic score function, as shown in Fig. 6.

Fig. 7 illustrates how the number of worker’s matching tasks affects the expected social cost and expected total payment generated by HERALD* under the linear score function. Specifically, Fig. 7(a) and 7(b) demonstrate the impact on the expected social cost and expected total payment obtained by HERALD*. It can be observed that increasing the number of matching tasks per worker leads to a decrease in the expected social cost and expected total payment in HERALD*. This is because, with more matching tasks assigned to a worker, the platform requires fewer workers to execute the requested tasks compared to the scenario with fewer matching tasks per worker. Consequently, this results in a lower expected social cost and expected total payment in HERALD*. Similar results are obtained for HERALD* under the logarithmic score function, as shown in Fig. 8.

VI. CONCLUSION

In this manuscript, we introduce HERALD*, a novel incentive mechanism for a task allocation system in which tasks arrive randomly according to a probability distribution. Our investigation indicates that HERALD* meets several desirable properties, such as truthfulness, individual rationality, differential privacy, low computational complexity, and low social cost. More specifically, we have shown that HERALD* guarantees \(\epsilon_1\) -differential privacy for both linear and logarithmic score functions, and it achieves a competitive ratio of \(\ln ln n\) on expected social cost. We have also validated the effectiveness of HERALD* through both theoretical analysis and extensive simulations.

REFERENCES


Yuan Luo (Member, IEEE) received the B.S. degree in applied mathematics, and the M.S. and Ph.D. degrees in probability statistics from Nankai University, Tianjin, China, in 1993, 1996, and 1999, respectively. From July 1999 to April 2001, he was a Post-Doctoral Researcher with the Institute of Systems Science, Chinese Academy of Sciences, Beijing, China. From May 2001 to April 2003, he was a Post-Doctoral Researcher with the Institute for Experimental Mathematics, University of Duisburg–Essen, Essen, Germany. Since June 2003, he has been with the Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, China. Since 2006, he has been a Full Professor and the Vice Director of the Department, from 2016 to 2018 and since 2021. His current research interests include coding theory, information theory, and big data analysis.

Boris Dödder is an associate professor at the department of computer science (DIKU) at the University of Copenhagen (UCPH), Denmark. He is head of the research group Software Engineering & Formal Methods at DIKU. His primary research interests are formal methods and programming languages in software engineering of trustworthy distributed systems, where he is studying automated program generation for adaptive systems with high-reliability guarantees. He is working on the computational foundations of reliable and secure Big Data ecosystems. His research is bridging the formal foundations of computer science and complex industrial applications.