Quality-Aware Pricing for Mobile Crowdsensing

Kai Han, Associate Member, IEEE, Member, ACM, He Huang, Member, ACM, and Jun Luo, Senior Member, IEEE, Member, ACM

Abstract—Mobile crowdsensing has been considered as a promising approach for large scale urban data collection, but has also posed new challenging problems, such as incentivization and quality control. Among the other incentivization approaches, posted pricing has been widely adopted by commercial systems due to the reason that it naturally achieves truthfulness and fairness and is easy to be implemented. However, the fundamental problem of how to set the “right” posted prices in crowdsensing systems remains largely open. In this paper, we study a quality-aware pricing problem for mobile crowdsensing, and our goal is to choose an appropriate posted price to recruit a group of participants with reasonable sensing qualities for robust crowdsensing, while the total expected payment is minimized. We show that our problem is NP-hard and has close ties with the well-known Poisson binomial distributions (PBDs). To tackle our problem, we first discover some non-trivial submodular properties of PBD, which have not been reported before, and then propose a novel “ironing method” that transforms our problem from a non-submodular optimization problem into a submodular one by leveraging the newly discovered properties of PBD. Finally, with the ironing method, several approximation algorithms with provable performance ratios are provided, and we also conduct extensive numerical experiments to demonstrate the effectiveness of our approach.

Index Terms—Crowdsourcing, approximation algorithms.

I. INTRODUCTION

With the proliferation of the hand-held devices (such as smart phones) and the diverse mobile sensors embedded in them, there has been a growing interest in Mobile Crowdsensing (MC) [1], where collective information about interested events can be acquired through the sensing data contributed by individuals for further investigation or learning. Consequently, we have recently witnessed a flourishing of MC systems in various domains such as healthcare, environmental monitoring, transportation and human activity detection [1]–[4].

While being recognized as a promising approach for sensing data collection, mobile crowdsensing also has posed some unique research challenges. As participants (or users) to MC tasks are self-interested and have to consume substantial efforts and physical resources for sensing activities, it is well acknowledged that suitable monetary compensation is necessary to incentivize mobile crowdsensing while discouraging the strategic behaviors of the participants (i.e., ensuring truthfulness) [5]–[8]. Furthermore, as the MC participants are usually unprofessional in both their sensing equipments and their sensing behaviors (hence resulting in low sensing qualities), the sensing data obtained from a single participant may be noisy or even useless [9]–[11]. This makes it necessary for a judicious MC task owner to select proper participants with reasonable sensing qualities as well as to introduce sensing redundancies (i.e., to form a “crowd”) so as to achieve sensing robustness, which would eventually brings the owner higher revenue gained from the collected data. Indeed, such a requirement on sensing robustness has been raised mandatorily in a lot of crowdsensing applications on data acquisition and distributed computing [12]–[16]. For example, in an urban monitoring application, we may require $k$-coverage of the Point of Interest (PoI), i.e., at least $k$ camera sensors should be activated to get a multi-view coverage of the PoI.

Based on the above observation, a sensing task owner needs to design proper incentive mechanisms for robust crowdsensing, which is a tricky problem. Indeed, even if the sensing quality issue is not taken into account, designing truthful mechanisms for crowdsensing incentivization is highly non-trivial. Most of the previous studies have addressed this problem based on auctions, i.e., the owner solicits sealed-bidding from the users and then decides the payment to each user [5], [15], [17]–[19]. However, as the users may not trust the owner and understand the mechanism to reveal their true costs, it could be complicated to implement this approach [20]. Moreover, sealed-bidding can result in heterogeneous payments to the users (a.k.a. “price discrimination”), which could arouse people’s resentment [21], [22]. Therefore, a more applicable way might be posted pricing, i.e., the sensing owner announces a unique take-it-or-leave-it price to the participants for a crowdsensing campaign, and then the participants choose to join the campaign or not according to their own willingness. It is noted that the posted pricing approach is very easy to be implemented and naturally achieves truthfulness [20]. Consequently, a lot of the current commercial crowdsensing/crowdsourcing systems (e.g., Amazon’s Mechanical Turk [23]) have adopted the posted pricing approach to incentivize participation.

Surprisingly, although posted pricing is a prevailing method for incentivizing participation in commercial systems, the fundamental problem of how to set a “right” price to recruit a
crowd with reasonable quality (and cost) still remains largely open. Intuitively, if the posted price is set too low, then the total (expected) cost for recruiting the participants would also be low; however, there could be too few participants who are willing to join the task, harming the sensing robustness thus the revenue gained from crowdsensing. Conversely, if a high posted price is blindly offered to a large group of users, then the owner may recruit a lot of users to achieve sensing robustness, but it could waste money on the participants with very poor qualities. To the best of our knowledge, such a tradeoff between the recruiting cost and sensing robustness in mobile crowdsensing has never been studied in the literature.

A. Contributions

In this paper, we propose a novel ex-ante posted pricing mechanism to achieve sensing robustness in mobile crowdsourcing. Our aim is to jointly choose an appropriate posted price and a set of candidate participants such that the total expected cost for paying the participants is minimized, while the number of the recruited participants with acceptable sensing qualities are larger than a predetermined lower bound (in a probabilistic sense) to maintain a desired level of sensing robustness. We call this problem as the Posted Pricing for Robust Crowd sensing (PPRC) problem and propose several algorithms to address it with provable performance bounds. More specifically, our major contributions include:

- We formulate the PPRC problem as essentially a non-linear integer programming problem, and we also prove its NP-hardness.
- To solve the PPRC problem efficiently, we perform a study on the well-known Poisson Binomial Distribution (PBD), which is closely related to our problem. We discover non-trivial submodular properties of PBD that have not been reported since Poisson’s seminal work on PBD [24], to the best of our knowledge.
- With the new discoveries on PBD, we further propose a novel “ironing method” that transforms our PPRC problem from a non-submodular optimization problem into a submodular one, and then propose an approximation algorithm, a bi-criteria approximation algorithm, as well as a learning algorithm to address it with provable performance ratios.
- We conduct extensive numerical experiments to demonstrate the effectiveness of our approach.

The rest of this paper is structured as follows. We review the related work in Sec. II. We introduce the models and formulate the PPRC problem in Sec. III. In Sec. IV, we reveal some non-trivial submodular properties of PBD, which are used to design our ironing method introduced by Sec. V. Based on the ironing method, we propose an approximation algorithm in Sec. VI and an bi-criteria approximation algorithm in Sec. VII. We also propose a learning algorithm for our problem in Sec. VIII. The results of extensive experiments are reported in Sec. IX. Finally, we conclude our paper in Sec. X. To maintain fluency, we postpone most of the (sketched) proofs to the Appendix.

II. Related Work

We review two lines of related work in the following.

A. Mobile Crowdsensing/Crowdsourcing

The incentivization problem in Crowdsensing/Crowdsourcing has aroused great interests in the literature [5]–[7], [15], [17]–[19], [32]–[36]. However, most of the work in this line has neglected the quality issue with only a few exceptions such as [9] and [11].

The excellent work in [9] has taken the assumption that the exact value of each user’s quality is known beforehand, and it has proposed some deterministic sealed-bidding auctions to maximize the social welfare of the crowdsourcing system. In contrast, only stochastic knowledge of the users’ qualities are known in our model, so our problem is essentially a stochastic optimization problem. Moreover, our optimization goal is to minimize the expected payment under a probabilistic quality constraint, which is totally different from that in [9].

Peng et al. [11] have put their main efforts on designing an elegant Expectation-Maximization algorithm to estimate the qualities of the users, and their goal is to maximize the social welfare of the system, without considering the sensing robustness constraint as accomplished in our paper. Our problem and methods are essentially different from those in [11], which can be seen from the fact that our problem is NP-hard while the one in [11] can be optimally solved.

Besides the differences on the assumptions, optimization goals and solution methods described above, a key distinction between our work and [9], [11] (as well as other related work) is that our approach works under a posted-pricing model while the others do not. As we have explained in Sec. I, posted pricing is easy to be implemented and naturally achieves truthfulness, so it has been widely adopted in crowdsourcing systems such as Amazon’s Mechanical Turk. However, very few proposals have investigated the crowdsourcing incentivization problem under the posted pricing model, and two notable studies in this line are [20] and [37]. More specifically, both [20] and [37] have studied the problem of deciding the posted prices under a predefined budget, such that the value of the recruited crowdsourcing users can be maximized. Unfortunately, neither [20] nor [37] has considered the quality issue, and their problems and methods are essentially different from ours. To the best of our knowledge, we are the first to propose a quality-aware incentivization mechanism for mobile
crowdsourcing under the posted-pricing model. Moreover, the proposed methods in our paper are also novel, as further discussed below.

B. Studies on PBD and Submodular Optimization

Poisson binomial distribution has been extensively studied (e.g., [25], [26], [28]–[31], [38]) since Poisson’s seminal work in 1837 [24]. Nevertheless, to the best of our knowledge, none of the previous studies has investigated the submodular properties of Poisson Binomial Distribution, as we have done in this work. Due to the wide applications of PBD, we believe that our discovery on PBD’s submodularity could be applied to more applications in the future.

The submodular optimization theory has also been extensively studied in the literature and many approximation algorithms have been proposed (e.g., [39], [40]). Due to the space constraint, the interested readers are kindly referred to a survey in [41]. However, as we will see in the following sections, the target function appeared in our problem is non-submodular, so the existing algorithms on submodular optimization cannot be applied to our problem, if without our proposed method for transforming a non-submodular optimization problem into a submodular one.

III. Problem Setup

In this section, we formally introduce the assumptions and definitions of our problem. For clarity, we list the frequently used notations in Table I. Suppose that we are given a crowdsensing task and a set of $n$ participants who are eligible for doing the task. Each participant $i \in \{1, \ldots, n\}$ has a private type $(q_i, c_i)$ which denotes that participant $i$ would produce output of quality $q_i$ at a cost $c_i$ if she would participate in the crowdsensing task. Following a large body of work on crowdsensing/crowdsourcing such as [5], [7], and [32]–[35], we consider a standard Bayesian setting where the “Bayesian knowledge” on the users’ qualities and costs is available, i.e.: the joint probability distribution of each user’s type $(q_i, c_i)$ is known to the crowdsensing task owner [33], but the exact values of $q_i$ and $c_i$ are unknown. We will also drop this assumption and extend our algorithms in Sec. VIII. To get close to the realistic situations, we assume that the types of different users are independently distributed and may follow different distribution functions due to the users’ heterogeneous behaviors and sensing capabilities.

As it is common for the users to monetarize their sensing costs on participating in crowdsensing, we assume that there is a finite set of candidate prices $W = \{w_1, \ldots, w_\sigma\}$ from which the task owner can choose the price posted to the participants, where $w_1 \leq \cdots \leq w_\sigma$. For example, if the task owner is willing to pay at most 1 dollar to one participant for performing the task, we can set $w_1 = 1, w_\sigma = 100$ and $\sigma = 100$ where $w_1$ denotes 1 cent. As users are rational people, we denote $\Pr(c_i \leq w_1)$ by $u_{i,1}$ for any $(i, l) \in [n] \times [\sigma]$, which is the probability that user $i$ would participate in the task when the posted price is $w_1$. Clearly, the expected payment for user $i$ is $w_{i,1} \cdot u_{i,1}$ when the price $w_{i,1}$ is posted to user $i$.

Note that if the quality of a user is too low, the data output by her could be simply random noise and useless to the task owner (or even harmful, due to the misreadings caused). We call such an user as a “casual user”, otherwise she is called a “normal user”. Based on this observation, the owner would set a lower bound $\gamma$ as the lowest quality demand for any user, and hence each user $i$ has a probability $p_{i,1} = \Pr(q_i \geq \gamma; c_i \leq w_1)$ for each $l \in [\sigma]$ which denotes the probability that she would participate in the crowdsensing task and behave as a normal user when the price $w_1$ is posted to her, and we assume $p_{i,1} \in (0, 1)$ without loss of generality.

To guarantee the overall quality of the sensing data, it is necessary to require that the crowdsensing task must be performed by multiple users such that the measurement error can be reduced by fusing multiple readings. Actually, such a requirement has been raised mandatorily in many proposals on crowdsensing such as [12]–[16]. For example, in the crowdsourced spectrum sensing application studied in [16], each spectrum sensing task needs to recruit at least $d$ users to determine spectrum occupancy, where $d_\text{iv}$ is a predefined integer calculated by using Neyman-Pearson Detectors.

Duan et al. [12] indicate that a task owner would get zero revenue if the number of mobile phones participating in the task is less than a predefined integer threshold (they call this as the “threshold revenue model”). Similarly, the model in [15] assumes that the same task should be performed by multiple users to accomplish a crowdsourcing mission. Therefore, we follow these related work by requiring that at least $k$ normal users must join the crowdsensing task, which is casted as a predefined demand of the task owner for sensing robustness.

Based on the above considerations, a task owner may blindly post a very high price to all users and hence recruit many users, none of them have high enough quality and therefore provide low-quality data.

TABLE I

<table>
<thead>
<tr>
<th>FREQUENTLY USED NOTATIONS</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Notation</strong></td>
</tr>
<tr>
<td>$[n]$</td>
</tr>
<tr>
<td>$(q_i, c_i)$</td>
</tr>
<tr>
<td>$W$</td>
</tr>
<tr>
<td>$u_{i,l}$</td>
</tr>
<tr>
<td>$p_{i,l}$</td>
</tr>
<tr>
<td>$Z_{i,l}$</td>
</tr>
<tr>
<td>$S_i(X)$</td>
</tr>
<tr>
<td>$\Delta_i(X)$</td>
</tr>
<tr>
<td>$\rho_l(X, k)$</td>
</tr>
<tr>
<td>$\tau_l(X, j)$</td>
</tr>
<tr>
<td>$\Gamma_l(X, a, b)$</td>
</tr>
<tr>
<td>$\Phi_l(i, X)$</td>
</tr>
<tr>
<td>$\Psi_l(X, \mu_1, \mu_2)$</td>
</tr>
</tbody>
</table>

---

1As indicated by the existing work [5], [7], [32]–[35], such Bayesian knowledge can be acquired from the historical data about the users; this acquiring process is generally considered to be application-dependent and is out of the scope of our paper.

2We use $[n]$ to denote the set $\{1, 2, \ldots, n\}$.
large number of participants to achieve sensing robustness, but this is probably not economical due to the excessive payments to the users, including those wasted on the casual users. Therefore, a more astute task owner would carefully select a group of target users and offer them a "just enough" posted price, such that at least \( k \) normal users will eventually join the task (in a probabilistic sense), while the total (expected) cost for paying the selected users is minimized. More specifically, let \( Z_{i,l} \) be a Bernoulli random variable with \( \Pr\{Z_{i,l} = 1\} = p_{i,l} \) and \( \Pr\{Z_{i,l} = 0\} = 1 - p_{i,l} \) and let \( S_l(X) = \sum_{i \in X} Z_{i,l} \) for any \( X \subseteq [n] \) and any \( l \in [\sigma] \), then the task owner’s demand on sensing robustness can be expressed as finding certain \((X, l)\) to satisfy \( \Pr\{S_l(X) \geq k\} \geq \theta \) where \( k \) and \( \theta \) are predefined thresholds, and our posted pricing problem can be formally stated by Definition 1:

**Definition 1:** Given a set of users \( \{1, 2, \ldots, n\} \), a set of posted prices \( W = \{w_1, \ldots, w_x\} \), the thresholds \((k, \theta) \in [n] \times (0, 1)\), and the set \( P = \{(u_{i,l}, p_{i,l}) | i \in [n], l \in [\sigma]\} \), let \( \rho_l(X, k) \) denote \( \Pr\{S_l(X) \geq k\} \) for any \( X \subseteq [n] \) and any \( l \in [\sigma] \). The Posted Pricing for Robust Crowdsensing (PPRC) problem is to find an optimal solution to the following mathematical optimization problem:

\[
\text{Minimize} \quad \sum_{i \in X} w_i \cdot u_{i,l} \quad \text{[PPRC]}
\]

\[
\text{s.t.} \quad \rho_l(X, k) \geq \theta \quad X \subseteq [n]; l \in [\sigma]
\]

**Remark:** The formulation of the PPRC problem implies that, we only post a price to a group of target users (i.e., the users in \( X \)), such that the total expected payment can be minimized under the quality constraint.

### A. A Closer Look at the Problem

According to our problem formulation, it can be seen that the random variable \( S_l(X) \) appeared in the PPRC problem follows a Poisson Binomial Distribution. The Poisson binomial distribution is a well-known discrete probability distribution of a sum of independent Bernoulli trials that are not necessarily identically distributed, and there exists an accurate expression of its p.m.f. More specifically, if we use \( F_X \) to denote the set of all subsets of \( i \) integers that can be selected from \( X \subseteq [n] \), we get the following equation [26]:

\[
\Pr\{S_l(X) = i\} = \sum_{D \in F_X} \left( \prod_{j \in D} p_{j,l} \prod_{j \notin X \setminus D} (1 - p_{j,l}) \right)
\]

Note that \( \rho_l(X, k) = \sum_{i=k}^{n} \Pr\{S_l(X) = i\} \), so if we input the above equation into the PPRC problem in Definition 1, we will get a non-linear 0-1 integer programming problem, which is generally very hard to solve [42]. Indeed, we find that the PPRC problem is NP-hard.\(^3\)

**Theorem 1:** The PPRC problem is NP-hard.

Another idea to solve our problem is to see if \( \rho_l(\cdot, k) \) is submodular, so we may apply some submodular optimization techniques. A function \( F : 2^{[n]} \rightarrow \mathbb{R} \) is called a submodular function if it has the “diminishing returns” property, i.e., for any \( X \subseteq Y \) and any \( x \notin Y \), we have \( F(x \mid X) \geq F(x \mid Y) \) where \( F(x \mid X) = F(X \cup \{x\}) - F(X) \), and it is called monotone if \( F(X) \leq F(Y) \). Unfortunately, \( \rho_l(\cdot, k) \) is not submodular. To see this, we study an example where \( n = 4 \), \( X = \{1, 2\} \), \( Y = \{1, 2, 3\} \), \( x = 4 \), \( p_{1,l} = 0.1 \), \( p_{2,l} = 0.2 \), \( p_{3,l} = 0.9 \), \( p_{4,l} = 0.3 \) and \( k = 2 \). It can be verified that \( \rho_l(X \cup \{x\}, k) - \rho_l(X, k) = 0.078 < \rho_l(Y \cup \{x\}, k) - \rho_l(Y, k) = 0.2022 \). So \( \rho_l(\cdot, k) \) is not submodular.

In the following sections, we will show how we can bypass all the difficulties described above to solve our problem.

### IV. On the Submodular Properties of PBD

As our problem is closely related to the Poisson binomial distribution, we start by studying the properties of PBD in this section, which will be used to design our algorithm later.

Recall that the random variable \( S_l(X) \) appeared in our problem obeys PBD. So \( \rho_l(\cdot, \cdot) \) can be regarded as a tail c.d.f. function of PBD. We find that, although \( \rho_l(\cdot, i) \) is not necessarily submodular for a given \( i \), the logarithm of the p.m.f. function of PBD does own some nice properties even stronger than submodularity. Having this in hand, we can prove the submodularity of \( \ln \Pr\{a \leq S_l(\cdot) \leq b\} \) on \( \{X \mid X \subseteq [n] \land |X| \geq a\} \) for any given integers \( 0 \leq a \leq b \leq n \) and any \( l \in [\sigma] \). Note that all the results proposed in this section are non-trivial and hold for general PBDs, which, for the first time, reveal the interesting relationships between PBD and submodularity. In the sequel, we will describe our results in details.

### A. Basic Properties of PBD

We first introduce some notations and basic facts that we will use in the following paragraphs. For any \( X \subseteq [n], l \in [\sigma] \) and \( i \in [n] \), we define \( \tau_l(X, i) = \Pr\{S_l(X) = i\} \). That is, given \( X \), \( \tau_l(X, \cdot) \) can be understood as the p.m.f. of the random variable \( S_l(X) \). Clearly, we have \( \rho_l(X, i) = \sum_{a=i}^{n} \tau_l(X, a) \). For any two integers \( a \leq b \), we define \( \Gamma_l(X, a, b) = \rho_l(X, a) - \rho_l(X, b+1) \), which is the probability that \( S_l(X) \in [a, b] \).

Note that \( \rho_l(\cdot, i) \) is monotone, i.e., for any \( X \subseteq Y \subseteq [n] \), we have \( \rho_l(X, i) \leq \rho_l(Y, i) \). However, \( \tau_l(\cdot, i) \) can be non-monotone. Moreover, given any \( x \in [n] \) \( X \), we have:

\[
\rho_l(X \cup \{x\}, i) = \left(1 - p_{x,l}\right)\rho_l(X, i) + p_{x,l}\rho_l(X, i - 1), \quad (2)
\]

which is a simple reduction about PBD [26]. Note that equation (2) also holds for \( i = 0 \) or \( X = \emptyset \).

For any \( i \in [n] \) and \( l \in [\sigma] \), we define \( \beta_{i,l} = \frac{p_{i,l}}{1 - p_{i,l}} \). For any \( \emptyset \neq X \subseteq [n] \), we define \( \Delta_l(X) = \prod_{i \in X} \left(1 - p_{i,l}\right) \). So we have the following equation:

\[
\tau_l(X, i) = \begin{cases} 
\Delta_l(X) \cdot \sum_{D \in F_X} \left( \prod_{j \in D} \beta_{j,l} \right) ; & \text{if } 1 \leq i \leq |X| \\
\Delta_l(X) ; & \text{if } i = 0
\end{cases}
\]

Note that \( \tau_l(X, i) = 0 \) for any \( i \neq \{0, \ldots, |X|\} \). To get a simplified expression of \( \tau_l(X, i) \), we use \( \Phi_l(i, X) \) to denote
∑_{D ∈ F(X)} \left( \prod_{j ∈ D} \beta_j, l \right) and define Φ_l(0, X) = 1. Hence the following equation holds for any \(0 ≤ i ≤ |X|\) and any \(l ∈ [σ]\):
\[τ_l(X, i) = Δ_l(X) · Φ_l(i, X)\]

**B. Revealing the Submodularity of PBD**

In this section, we will show that both \(τ_l(·, i)\) and \(Γ_l(·, a, b)\) have nice submodular properties. All the theorems and lemmas presented in this section hold for any \(l ∈ [σ]\).

We first introduce a new “set blending” operation \(∪\) which is similar to the set union operation but without removing duplicate elements, as defined in Definition 2:

**Definition 2 (“Set Blending” operation):** For any two sets \(A\) and \(B\), \(A ∪ B\) is an unordered collection of elements such that each element in \((A ∪ B) \setminus (A ∩ B)\) appears once in \(A ∪ B\), whereas each element in \(A ∩ B\) appears twice in \(A ∪ B\). If \(C = A ∪ B\), then we use \(g(C)\) to denote the set \(A ∩ B\), i.e., the set of elements that appear two times in \(C\).

For example, we have \(\{1, 8\} ∪ \emptyset = \{1, 8\}, \{1, 2, 3\} ∪ \{2, 4\} = \{1, 3, 2, 4\}, \prod_{i ∈ \{2, 3, 1\}} i = 12\) and \(g(\{5, 5, 6, 7, 1, 1\}) = \{1, 5\} \). With Definition 2, we now introduce Lemma 1-3, which reveal some non-trivial algebraic properties of PBD:

**Lemma 1:** Given a set \(X\) and two integers \(a, b\) which satisfy \(\emptyset ≠ X ⊆ [n]\), \(0 ≤ a, b ≤ |X|\) and \(ab ≠ 0\), let \(W(X, a, b) = \{c|∃C_1 ∈ \mathbb{F}^X ∩ C_2 ∈ \mathbb{F}^X ∩ (C = C_1 ∪ C_2)\}\). Then we have
\[Φ_l(a, X) · Φ_l(b, X) = \sum_{c ∈ W(X, a, b)} \left( a + b - 2g(c) \right) \prod_{i ∈ c} β_{k,i} \]

**Lemma 2:** Given a set \(X\) and three integers \(a, b, c\) satisfying \(\emptyset ≠ X ⊆ [n]\) and \(0 ≤ c ≤ a ≤ b ≤ |X|\), we have
\[Φ_l(b - c, X) · Φ_l(a, X) ≥ Φ_l(b, X) · Φ_l(a - c, X)\]

**Lemma 3:** Given two sets \(X, Y\) and an integer \(a\) which satisfy \(\emptyset ≠ X ⊆ Y ⊆ [n]\) and \(0 ≤ a ≤ |X|\), we have
\[Φ_l(a, Y) = \sum_{i = 0}^{\min\{a, |Y| - |X|\}} \Phi_l(a - i, X)Φ_l(i, Y \setminus X)\]

Using Lemma 1-3, we can get the following theorem:

**Theorem 2:** Given two sets \(X, Y\) and two integers \(a, b\) which satisfy \(X ⊆ Y ⊆ [n]\) and \(0 ≤ a ≤ b ≤ |X|\), we have:
\[τ_l(X, a)τ_l(Y, b) ≥ τ_l(X, b)τ_l(Y, a)\]

Note that Theorem 2 implies that the function \(ln τ_l(·, b)\) obeys submodularity on \(\{X| X ⊆ [n] \land |X| ≥ b\}\). Actually, Theorem 2 is even stronger, because we only need \(τ_l(X, b - 1)τ_l(Y, b) ≥ τ_l(X, b)τ_l(Y, b - 1)\) to prove the submodularity of \(ln τ_l(·, b)\) by using (2). In the following theorem, we generalize Theorem 2 to show that \(ln Γ_l(·, a, b)\) also obeys submodularity on \(\{X| X ⊆ [n] \land |X| ≥ a\}\):

**Theorem 3:** Given two sets \(X, Y\) and two integers \(a, b\) which satisfy \(X ⊆ Y ⊆ [n]\) and \(0 ≤ a ≤ b ≤ n\) and \(a ≤ |X|\), we have
\[Γ_l(X ∪ \{x\}, a, b)Γ_l(Y, a, b) ≥ Γ_l(X, a, b)Γ_l(Y ∪ \{x\}, a, b)\]

for any \(x ∈ [n] \setminus Y\).

V. THE IRONING METHOD

With Theorem 3, we are closer to solving our problem now, as we have shown that \(ln ρ_l(x, k) = ln Γ_l(x, k, n)\) is submodular on the domain \(\{X| X ⊆ [n] \land |X| ≥ k\}\). However, we still cannot use existing submodular optimization algorithms for our problem because: (1) \(ln ρ_l(X, k)\) is negative; (2) \(ln ρ_l(x, k)\) has no definition when \(|X| < k\) as \(ρ_l(x, k) = 0\). Therefore, we further design two surrogate functions \(Ψ_l\) and \(φ_l\) for our problem, which are defined on \(2^n\) and align with the function \(ρ_l\) for any \(l ∈ [σ]\).

Intuitively, these surrogate functions take the effect that “iron” the non-submodular function \(ρ_l\) to a submodular one without compromising the optimality of the original solution to the PPRC problem, and hence enable us to transform the PPRC problem into a submodular optimization problem. We call this approach as the “ironing method” and will describe it in details in the following.

A. A General Surrogate Function for PBD

We first propose a submodular function \(Ψ_l\) (shown in Theorem 4) which serves as a general surrogate function for arbitrary PBDs, and hence it could also be applied in other related optimization problems. The function \(Ψ_l\) is not only non-negative and submodular on \(2^n\), but also has the nice property that \(Ψ_l\) reaches its smallest values on \(\{X| |X| < μ_1\}\), so \(Ψ_l\) is aligned with \(Γ_l\) as \(Γ_l(X, μ_1, μ_2) = 0\) for any \(X\) with \(|X| < μ_1\).

**Theorem 4:** Given any \(l ∈ [σ]\) and any two integers \(μ_1, μ_2\) satisfying \(0 ≤ μ_1 ≤ μ_2 ≤ n\), let \(δ_l = \min\{Γ_l(X, μ_1, μ_2)| X ⊆ [n] \land |X| ≥ μ_1\}\). Define
\[Ψ_l(X, μ_1, μ_2) = \begin{cases} α_l |X|; & \text{if } |X| < μ_1 \\ μ_1 α_l + ln Γ_l(X, μ_1, μ_2); & \text{if } |X| ≥ μ_1 \end{cases}\]

If \(α_l ≥ -2ln δ_l\), then we have
1) \(Ψ_l(·, μ_1, μ_2)\) is a non-negative and submodular function defined on \(2^n\).
2) For any \(X_1, X_2 ⊆ [n]\) such that \(|X_1| < μ_1\) and \(|X_2| ≥ μ_1\), we have \(Ψ_l(X_1, μ_1, μ_2) ≤ Ψ_l(X_2, μ_1, μ_2)\).

Let \((s_1, s_2, ..., s_n)\) be a permutation of \([n]\) satisfying \(p_{s_1, l} ≤ p_{s_2, l} ≤ ⋯ ≤ p_{s_n, l}\). When \(μ_1 = k\) and \(μ_1 = n\), it can easily seen that \(δ_l = \prod_{i = 1}^{k} s_i, l\), as \(ρ_l(X, l)\) is monotone with respect to the probabilities in \(\{p_i, l| i ∈ X\}\). Therefore, we can fix \(α_l = -2 \sum_{i ∈ X} ln p_{s_i, l}\) according to Theorem 4 and get the following [PPRICO] problem:

\[
\text{Minimize} \ \sum_{X,l} w_{i,l} \cdot u_{i,l} \quad \text{[PPRICO]} \\
\text{s.t.} \ Ψ_l(X, k, n) ≥ kα_l + ln θ; \\
X ⊆ [n]; l ∈ [σ] \quad (4)
\]

As Theorem 4 has proved that \(Ψ_l\) is submodular on \(2^n\), it can be seen that the [PPRICO] problem is essentially a “submodular cover” problem [40] for any given \(l\).

---

4Setting \(α_l\) to be some value larger than \(-2ln δ_l\) would not affect the correctness of our results. The detailed analysis on this fact is omitted due to the space constraint.
B. Further Transformation for the PPRC Problem

Although we have shown that the original [PPRC] problem can be transformed into a submodular cover problem [PPRCC], we still cannot readily solve it with provable bounds, as current techniques for solving the submodular cover problem usually require additional conditions or relaxations to get some nice performance ratios [39], [40]. To the best of our knowledge, the best performance bound for solving the non-integer submodular cover problem (as in our case) is proposed by Wan et al. [40], but only for some special submodular functions. We quote the results of [40] below:

Theorem 5 [40]: Let $F$ be a non-negative, monotone submodular function defined on $2^G$ where $G$ is the ground set. If $F(\emptyset) = 0$, then such an $F$ is called a polymatroid function. Suppose that $Y_{opt}$ is an optimal solution to a submodular cover problem and $\text{cst}(Y_{opt})$ is the cost of $Y_{opt}$. If $F(Y_{opt}) = F(G) \geq \text{cst}(Y_{opt})$ and a greedy algorithm always selects an element $x \in G \setminus X$ such that $F(x|X) \geq \text{cst}(x)$ when $F(X) < F(G)$, then the greedy algorithm is a $1 + \bar{g} \ln \text{cst}(Y_{opt})$ approximation to the submodular cover problem, where $\bar{g} = 1$ if the cost function $\text{cst}$ is linear (i.e., modular).

Note that we have $G = [n]$ and the cost function is modular in [PPRCC]. However, Theorem 5 requires $F(Y_{opt}) = F(G) \geq \text{opt}$ and $F(x|X) \geq \text{cst}(x)$ to get a logarithmic ratio. To apply Theorem 5, we design a new surrogate function $\phi_l$ based on $\Psi_l$, for any $l \in [\sigma_l]$, and prove that the function $\phi_l$ satisfies all the conditions required in Theorem 5. More specifically, suppose that $X_{opt,l}$ is an optimal solution to [PPRCC] for any given $l$. Then we have:

Theorem 6: Given $l \in [\sigma_l]$, let $\alpha_l = -2 \sum_{i=1}^{k} \ln p_{s_i}$, $\phi_l = \sum_{i \in X_{opt,l}} w_i \cdot p_{i,l}$, $d_l = \max\{w_i \cdot u_{i,l} | i \in [n]\}$, $\lambda_l = \ln p_l([n]) - \ln p_l([s_1])$, $\lambda_l = \ln \theta - \max\{\ln p_l(X_i), X_i \subseteq [n] \land \rho_l(X_i, k) < \theta\}$ and $\lambda_l = \min\{\lambda_l, \lambda\}$. For any $X \subseteq [n]$, define $\phi_l(X) = \frac{\phi_l + d_l}{\lambda_l} \min\{\Psi_l(X, k, n), \alpha_l k + \ln \theta\}$

Then $\phi_l$ is a polymatroid function. Moreover, we have:

1. $\phi_l(X_{opt,l}) = \phi_l([n]) \geq \text{opt}.$

2. For any $X \subseteq [n] : \phi_l(X) < \phi_l([n])$ and any $x \in [n] \setminus X$, we have $\phi_l(x|X) \geq d_l$.

Using the surrogate function $\phi_l$, we can further transform the [PPRCC] problem into the following [PPRC1] problem:

Minimize $\sum_{i \in X} w_i \cdot u_{i,l}$ [PPRC1]

s.t. $\phi_l(X) = \frac{\phi_l + d_l}{\lambda_l} (\alpha_l k + \ln \theta)$

$X \subseteq [n]; l \in [\sigma_l]$

As Theorem 6 proves that the function $\phi_l$ satisfies the conditions required in Theorem 5, we can design an algorithm for [PPRC1] based on Theorem 5 to achieve a provable approximation ratio. However, [PPRC1] requires to know $\text{opt}$ (and hence the optimal solution to [PPRCC]), which is impractical. In the next section, we will first propose an algorithm under the assumption that $\text{opt}$ is known, and then provide an equivalent algorithm without this assumption.

VI. APPROXIMATION ALGORITHM

With the ironing method proposed in Section V, our idea for solving the [PPRC] problem is to solve the [PPRC1] problem instead, hence to get a provable performance bound based on Theorem 5. However, to apply this idea, we need to check if [PPRC1] is equivalent to [PPRC]. Fortunately, we find the following theorem:

Theorem 7: Given any $l \in [\sigma_l]$, if [PPRC] and [PPRC1] are not equivalent, then we can find an optimal solution to [PPRC] in polynomial time.

Proof: It can be easily seen that [PPRC1] and [PPRC] are equivalent, as eqn. (4) and eqn. (5) are equivalent according to the definition of $\phi_l$. Note that each feasible solution to [PPRC] is also a feasible solution to [PPRC1]. Therefore, if [PPRC] and [PPRC1] are not equivalent, then there must exist a feasible solution $(X', l)$ to [PPRC1] that is not a feasible solution to [PPRC]. In this case, we must have $|X'| < k$, because otherwise we have

$\Psi_l(X', k, n) = k \alpha_l + \ln \Gamma_l(X', k, n) \geq k \alpha_l + \ln \theta$

and hence $\rho_l(X', k) = \Gamma_l(X', k, n) \geq \theta$, which contradicts that $(X', l)$ is not a feasible solution to [PPRC].

Let $X^*_l$ be a set of $k$ users in $[n]$ with the minimum total expected payment under the posted price $w_l$ (ties broken arbitrarily). As $|X'| < k$, we can use 2) of Theorem 4 to get $k \alpha_l + \ln \rho_l(X^*_l, k) = \Psi_l(X^*_l, k, n) \geq \Psi_l(X', k, n) \geq k \alpha_l + \ln \theta$ and hence $\rho_l(X^*_l, k) \geq \theta$, which implies that $(X^*_l, l)$ is an optimal solution to [PPRC]. Note that $X^*_l$ can be computed in polynomial time: we can first sort $[n]$ into $(v_1, \ldots, v_n)$ such that $u_{s_1,l} \leq u_{v_1,l} \leq \cdots \leq u_{v_n,l}$, and then set $X^*_l = \{v_1, \ldots, v_k\}$. Hence the lemma follows.

With Theorem 5-7, a straightforward approach is directly using $\phi_l$ in a greedy algorithm to solve [PPRC1], which is shown in Algorithm 1. For each $l \in [\sigma_l]$, Algorithm 1 first tries to find an optimal solution to [PPRC1] according to the

Algorithm 1 Approximation Algorithm Using $\phi$

Input: $n, P, \theta, k, W$

Output: A selected group $Y$ and a posted price $c$

1. $t \leftarrow +\infty$; $Y \leftarrow \emptyset$; $e \leftarrow 0$;

2. for $l = 1$ to $\sigma_l$

3. Sort $[n]$ into $(s_1, \ldots, s_n)$ and $(v_1, \ldots, v_n)$ such that $p_{s_1,l} \leq \cdots \leq p_{s_n,l}$ and $u_{v_1,l} \leq \cdots \leq u_{v_n,l}$;

4. if $\rho_l(v_1, \ldots, v_k, k) \geq \theta$ then

5. $X \leftarrow \{v_1, \ldots, v_k\}$

6. else

7. $\alpha_l \leftarrow -2 \sum_{i=1}^{k} \ln p_{s_i}$;

8. $\xi \leftarrow \frac{\phi_l + d_l}{\lambda_l} (\alpha_l k + \ln \theta)$;

9. $\phi_l(X) < \xi \land |X| < n$ do

10. $x \leftarrow \arg \max_{0 \neq x \in [n]} \phi_l(x|X) - \phi_l(x)$;

11. $X \leftarrow X \cup \{x\}$;

12. if $\rho_l(X, k) \geq \theta$ then

13. $t \leftarrow \sum_{i \in X} w_i u_{i,l} < t$ then

14. return $Y, e$
proof of Theorem 7 (lines 3-5), then it uses a greedy strategy to find an approximate solution (lines 7-11) if the optimal solution is not found. The final solution output by Algorithm 1 is the one with the minimum cost for all \( l \in [\sigma] \) (lines 12-13). Unfortunately, as \( \text{opt} \) is unknown, Algorithm 1 is not implementable due to line 8. However, by a careful study on the definitions of \( \phi_l \) and \( \Psi_l \), it can be seen that, when line 10 is executed, we always have:

\[
\phi_l(x|X) = \begin{cases} 
\ln \min \{ \rho_l(X \cup \{x\}, k, \theta) - \ln \rho_l(X, k); \alpha_l \} & \text{if} \\
\ln \min \{ \rho_l(X \cup \{x\}, k, \theta) + \alpha_l \} & \text{otherwise}
\end{cases}
\]

where \( \alpha_l = \frac{\text{opt} + d_l}{\lambda} \) and i)-iii) correspond to the cases of \(|X| \geq k, |X| < k-1 \) and \(|X| = k-1 \), respectively. Hence we can abandon \( \phi_l \) and get an algorithm equivalent to Algorithm 1, shown by Algorithm 2.

**Algorithm 2 Revised Approximation Algorithm Without Using \( \phi \)**

**Input:** \( n, P, \theta, k, W \)

**Output:** A selected group \( Y \) and a posted price \( e \)

1. \( t \leftarrow +\infty; \ Y \leftarrow \emptyset; \ e \leftarrow 0; \)

2. for \( l = 1 \) to \( \sigma \) do

3. Sort \([n]\) into \((s_1, \ldots, s_n)\) and \((v_1, \ldots, v_n)\) such that \( p_{s_1, l} \leq \cdots \leq p_{s_n, l} \) and \( u_{s_1, l} \leq \cdots \leq u_{s_n, l} \);

4. if \( \rho_l(v_1, \ldots, v_k, k) \geq \theta \) then

5. \( X \leftarrow \{v_1, v_2, \ldots, v_k-1\} \);

6. else

7. \( X \leftarrow \{v_1, v_2, \ldots, v_{k-1}\} \);

8. \( \alpha_l \leftarrow -2 \sum_{i=1}^{k} \ln p_{s_i, l} \);

9. \( x \leftarrow \arg \max_{x \in [n \setminus X]} \left[ \ln \min \{ \rho_l(X \cup \{x\}, k, \theta) + \alpha_l \}ight] \);

10. \( X \leftarrow X \cup \{x\} \);

11. while \( \rho_l(X, k) < \theta \land |X| < n \) do

12. \( x \leftarrow \arg \max_{x \in [n \setminus X]} \left[ \ln \min \{ \rho_l(X \cup \{x\}, k, \theta) - \ln \rho_l(X, k) \}ight] \);

13. \( X \leftarrow X \cup \{x\} \);

14. if \( \rho_l(X, k) \geq \theta \land \sum_{i \in X} w_{i,l} \leq t \) then

15. \( t \leftarrow \sum_{i \in X} w_{i,l}; \) \( \langle X, e \rangle \leftarrow \langle X, w_l \rangle \);

16. return \( \langle Y, e \rangle \)

VII. A BI-CRITERIA APPROXIMATION ALGORITHM

In this section, we consider the tradeoff between the quality constraint (i.e., equation (1)) and the total expected payment in PPRC, and propose a bi-criteria approximation algorithm to solve our problem. We clarify the concept of “bi-criteria approximation” in Definition 3:

**Definition 3:** Let \( (X^*, t^*) \) be an optimal solution to the PPRC problem and \( \varepsilon \) be any number in \((0, \theta)\). If \( (X, t) \in 2^{|n|} \times [\sigma] \) satisfies: (1) \( \rho_l(X, k) \geq \theta - \varepsilon; \) (2) \( \sum_{i \in X} w_{i,l} \leq \zeta \cdot \sum_{i \in X} w_{i,l}, t^*, \) then \( (X, t) \) is called a \( (\varepsilon, \zeta) \) bi-criteria approximation solution to the PPRC problem.

Based on Definition 3, we propose a bi-criteria approximation algorithm for PPRC, as shown by Algorithm 3. In Algorithm 3, we use \( \Psi_l \) as the surrogate function for \( \rho_l \) \((\forall l \in [\sigma])\), and employ a greedy searching process similar to that in Algorithm 1. However, we change the stopping rule of the greedy searching process from the condition of \( \phi_l(X) < \xi \) (shown in line 9 of Algorithm 1) to a new stopping rule

**Algorithm 3 A Bi-Criteria Approximation Algorithm**

**Input:** \( n, P, \theta, k, W, \varepsilon \)

**Output:** A selected group \( Y' \) and a posted price \( e' \)

1. \( t \leftarrow +\infty; \ Y' \leftarrow \emptyset; \ e' \leftarrow 0; \)

2. for \( l = 1 \) to \( \sigma \) do

3. Sort \([n]\) into \((s_1, \ldots, s_n)\) and \((v_1, \ldots, v_n)\) such that \( p_{s_1, l} \leq \cdots \leq p_{s_n, l} \) and \( u_{s_1, l} \leq \cdots \leq u_{s_n, l} \);

4. if \( \rho_l(v_1, \ldots, v_k, k) \geq \theta - \varepsilon \) then

5. \( X \leftarrow \{v_1, v_2, \ldots, v_k\} \);

6. else

7. \( \alpha_l \leftarrow -2 \sum_{i=1}^{k} \ln p_{s_i, l} \);

8. \( \eta \leftarrow \alpha_l k + \ln \theta; \ X \leftarrow \emptyset; \ v \leftarrow -\ln \theta - \ln(\theta - \varepsilon) \);

9. while \( \Psi_l(X, k, n) \leq \eta \) do

10. \( x \leftarrow \arg \max_{x \in [n \setminus X]} \{ \Psi_l(X \cup \{x\}, k, n) - \Psi_l(X, k, n) \}; \)

11. \( X \leftarrow X \cup \{x\} \);

12. if \( \rho_l(X, k) \geq \theta - \varepsilon \land \sum_{i \in X} w_{i,l} \leq t \) then

13. \( t \leftarrow \sum_{i \in X} w_{i,l}; \) \( \langle Y', e' \rangle \leftarrow \langle X, w_l \rangle \);

14. return \( \langle Y', e' \rangle \)

It can be seen that Algorithm 2 differs from Algorithm 1 in lines 7-13, where it first adds \( k - 1 \) users with the smallest expected costs into the result set (line 7). After that, the algorithm adds the users one by one, each time selects a user that maximizes the ratio of its “incremental contribution” to its cost (lines 9 and 12). Interestingly, when selecting the \( k \)th user, the algorithm uses \( \ln \min \{ \rho_l(X \cup \{x\}, k, \theta) + \alpha_l \} \) to calculate the incremental contribution (line 9), where \( \alpha_l \) can be understood as the estimated contribution of \( k - 1 \) users, although we have \( \rho_l(X, k) = 0 \) for \(|X| < k \).

Let \( \text{opt} \) be the cost of an optimal solution to [PPRC] and let \( \lambda = \min \{\lambda_1, \ldots, \lambda_{\sigma} \} \). Using Theorem 5-7, we can readily get:

**Theorem 8:** The approximation ratio of Algorithm 2 for the PPRC problem is at most \( 1 + \ln \left( \frac{\sum_{i=1}^{k-1} \ln p_{s_i, l}}{\lambda} \right) + \ln \left( 1 + \frac{w_{\text{opt}}}{\lambda} \right) \).
of $Ψ_t(X,k,n) < \eta - \varepsilon_1$ (shown in line 9 of Algorithm 3). Such a revision is due to the relaxed quality requirement in a bi-criteria approximation solution to PPRC, and the parameters $\eta$ and $\varepsilon_1$ are set according to Definition 3 and the definition of $Ψ_t$ in Theorem 4 (see line 8 of Algorithm 3). By the reasoning similar to that for Algorithm 2, it can be seen that the time complexity of Algorithm 3 is also $O(n^2\sigma)$.

Inspired by the method proposed by [43], we can prove the approximation ratio of Algorithm 3, which is shown by Theorem 9.

**Theorem 9:** For any $\varepsilon \in (0, \theta)$, Algorithm 3 outputs a $(\varepsilon, 1 + \ln \frac{\ln(\ln(N\varepsilon^{-1}))}{\varepsilon^2})$ bi-criteria approximation solution to the PPRC problem.

Note that Theorem 9 has revealed a tradeoff between the approximation ratio and the quality constraint. Moreover, when $\varepsilon$ is relatively large, the approximation ratio in Theorem 9 can be smaller than that got by directly calling Algorithm 2 with $\theta - \varepsilon$. This is due to the reason that, by relaxing the constraint of [PPRC], Theorem 9 has used different reasoning to prove the approximation ratio.

**VIII. HANDLING UNKNOWN DISTRIBUTIONS AND MULTIPLE TASKS**

In this section, we generalize our approach to handle the case of unknown distributions of the users and multiple homogeneous crowdsensing tasks. More specifically, we assume that there are $T$ homogeneous crowdsensing tasks which arrive in an online manner, and the sensing qualities and costs of any user $i \in [n]$ for different tasks are drawn i.i.d. from an unknown underlying distribution specific to that user. In such a case, our problem turns into a “Learning to Dynamically Pricing (LDP)" problem, where the task owner has to find an approximate solution $(X_i, l_i)$ to the PPRC problem for any task $t \in [T]$, by observing and learning from the users' behaviors on the tasks processed in the past (i.e., the tasks $1, \ldots, t - 1$).

When the users’ distributions are unknown, the tuple $(X_i, l_i)$ selected by the task owner for any task $t$ may violate the quality constraint in the PPRC problem (i.e., equation (1)). Therefore, we introduce a penalty function $Q(\cdot)$ to model the loss of the task owner due to the violation of equation (1), where $Q(\cdot)$ can be any non-decreasing function satisfying $\lim_{x \to 0^+} Q(x) = Q(0) = 0$. Based on this penalty function, we further define the total loss of any solution $(X, l)$ to the PPRC problem as follows:

$$L(X, l) = C(X, l) + Q(\max\{\theta - P_t(X, k), 0\}) ,$$  \hspace{1cm} (6)

where $C(X, l) = \sum_{i \in X} w_i \cdot u_{i, t}$ is the total expected payment of choosing $(X, l)$. Intuitively, if equation (1) is satisfied by $(X_i, l_i)$ for any task $t$, then $L(X_i, l_i)$ equals to $C(X_i, l_i)$; otherwise, a positive additive factor in proportion to the quality degradation (i.e., $Q(\max\{\theta - P_t(X, k), 0\})$) is considered in the total loss of $(X_i, l_i)$.

Inspired by the framework of Multi-Armed Bandit algorithms [44], we formally formulate the concept of “$(\zeta_1, \zeta_2)$-regret" as the performance measure of our algorithm for the LDP problem, shown in Definition 4:

**Definition 4:** Let $(X^*, l^*)$ be an optimal solution to the PPRC problem under the Bayesian setting. A pricing policy $X = \{(X_1, l_1), \ldots, (X_T, l_T)\}$ for $T$ tasks has a $(\zeta_1, \zeta_2)$ regret if and only if

$$\mathbb{E}\left\{\sum_{t=1}^{T} L(X_t, l_t) - \zeta_1 \cdot T \cdot L(X^*, l^*)\right\} = \zeta_2$$

Intuitively, the parameter $\zeta_1$ appeared in Definition 4 can be understood as the Approximation Ratio of $X$ under the case of Unknown Distributions (ARUD), and $\zeta_2$ is the Accumulated Expected Loss (AEL) of the pricing policy on $T$ tasks. Based on Definition 4, we propose Algorithm 4 to solve the LDP problem, which satisfies: (1) the ARUD of Algorithm 4 asymptotically approaches the approximation ratio of Algorithm 2. (2) The AEL of Algorithm 4 is upper bounded by $O(\sqrt{T})$, so the expected average loss (per task) of the pricing policy output by Algorithm 4 converges to zero when $T$ increases. In other words, Algorithm 4 is a “Hannan-consistent" (or "zero regret") learning algorithm [44].

**Algorithm 4 The LDP Algorithm**

**Input:** $n, \theta, k, W, T$

**Output:** A pricing policy $X$

1. $z \leftarrow \sqrt{T}$
2. for $l = 1$ to $\sigma$
3. for $j = 1$ to $z$
4. $t \leftarrow (l - 1) \cdot z + j$
5. Post price $w_j$ to all the users and observe $U_{i, t, l}$ and $P_{i, t, l}$ ($\forall i$) for task $t = (l - 1) \cdot z + j$
6. $(X_t, l_t) \leftarrow ([n], l)$
7. for $i = 1$ to $n$
8. for $l = 1$ to $\sigma$
9. $t_1 \leftarrow (l - 1) \cdot z + 1$; $t_2 \leftarrow l - 1$
10. $\hat{u}_{i, t} \leftarrow \sum_{t_1 \leq t \leq t_2} U_{i, t, l}$; $\hat{p}_{i, t} \leftarrow \sum_{t_1 \leq t \leq t_2} P_{i, t, l}$
11. $\theta' \leftarrow \theta - 1/T$
12. Using $P' = \{(\hat{u}_{i, t}, \hat{p}_{i, l}) | i \in [n], l \in [\sigma]\}$ and $\theta'$ to call Algorithm 2 and get an approximate solution $(\hat{X}, \hat{l})$ to the PPRC problem
13. for $t = \sigma + 1$ to $T$
14. $(X_t, l_t) \leftarrow (\hat{X}, \hat{l})$
15. return $X = \{(X_1, l_1), \ldots, (X_T, l_T)\}$

Algorithm 4 employs an “explore-then-exploit" strategy to find a pricing policy to the PPRC problem. In the "exploration phase" (lines 1-12), it posts every price to each user for $O(\sqrt{T})$ tasks, and then observes the responses of any user $i$ for any task $t$ and any price $l$, which is denoted by $(U_{i, t, l}, P_{i, t, l}) \in \{0, 1\} \times \{0, 1\}$, More specifically, $U_{i, t, l}$ indicates whether the user $i$ accepts the price $w_l$ for task $t$, while $P_{i, t, l}$ indicates whether the user $i$ accepts the price $w_l$ for task $t$ and behaves as a normal user. Based on these observations, Algorithm 4 derives estimations on the probability values in $\{\hat{u}_{i, t, p, i}, \hat{p}_{i, l} | i \in [n], l \in [\sigma]\}$ (line 10), and then uses these estimations and a revised parameter on quality demand (i.e., $\theta'$) to find an approximation solution $(\hat{X}, \hat{l})$ to the PPRC problem (line 12).
Fig. 1. Comparing different algorithms for the PPRC problem when the threshold scales. (a) $n = 500, k = 100$. (b) $n = 1000, k = 200$. (c) $n = 2000, k = 500$.

Fig. 2. Comparing different algorithms for the PPRC problem when the crowd size scales. (a) $k = 100, \theta = 0.2$. (b) $k = 200, \theta = 0.5$. (c) $k = 500, \theta = 0.8$.

Fig. 3. Comparing different algorithms for the PPRC problem when the penalty factor scales. (a) $n = 500, k = 100, \theta = 0.2$. (b) $n = 1000, k = 200, \theta = 0.5$. (c) $n = 2000, k = 500, \theta = 0.8$.

Afterwards, $(\bar{X}, \bar{l})$ is used in the “exploitation phase” to handle the leftover tasks (lines 13-14). Note that Algorithm 4 spends $O(\sqrt{Tn\sigma})$ time in the exploration phase, and then spends $O(n^2\sigma)$ time for calling Algorithm 2 to get $(\bar{X}, \bar{l})$. After that, it spends $O(T)$ time in Lines 13-14. Therefore, the total time complexity of Algorithm 4 for handling $T$ tasks is $O(\sqrt{Tn\sigma} + T + n^2\sigma)$.

Using Definition 4 and Hoeffding’s inequality (Lemma 4), we can prove the regret bound of Algorithm 4 as follows:

**Theorem 10:** Let $\kappa$ be the approximation ratio of Algorithm 2. Given any $\varepsilon > 0$, Algorithm 4 asymptotically achieves a $((1 + \varepsilon)\kappa, O(\sqrt{T}))$ regret bound.

**IX. PERFORMANCE EVALUATION**

We conduct extensive numerical experiments to evaluate the performance of our algorithms. The objective of our experiments is to corroborate the effectiveness of our algorithms under different parameter settings (such as $\theta, n$ and $k$). To simulate the users’ types in mobile crowdsensing (see Sec. III), we generate various bivariate distributions including the bivariate normal, exponential and uniform distributions as the prior knowledge on the qualities and costs of the users, such that any
two different users’ types follow different distributions. All the generated distributions are truncated to have the support \([0, 1] \times [0, 1]\), and the correlation coefficient of each generated bivariate distribution is randomly sampled from \((0,1]\) to model the positive correlations between the participants’ qualities and costs in reality. Without loss of generality, we set \(\gamma = 0.1\) as the lowest quality demand for any user, and set \(W = \{0.1, 0.2, \ldots, 0.9\}\) as the set of candidate prices.

In our experiments, we implement Algorithm 2, Algorithm 3 and Algorithm 4 for comparisons, and they are denoted by “PPRC”, “BCAA” and “LDP” in our figures, respectively. The input parameter \(\varepsilon\) in BCAA is set as \(\varepsilon = \theta/3\). We also implement the EXACT algorithm as a benchmark, which finds an optimal solution to the PPRC problem by enumerating all subsets of \(\{n\}\). As both BCAA and LDP may output solutions that only approximately satisfy equation (1), we use a linear penalty function \(Q(x) = \varepsilon x\) to measure the loss of the task owner when equation (1) in the PPRC problem is violated. Unless otherwise stated, we set the penalty factor \(\varepsilon\) to 50 in the experiments. With this penalty function, the total loss \(L(X, l)\) defined in equation (6) is used as the metric in our comparisons.

As EXACT has exponential time complexity, we first compare PPRC, BCAA and LDP under the case of large crowd sizes in Figs. 1-3, where the reported data are the Average Total Losses (ATLs) of the tested algorithms over 1000 data sets (for 1000 tasks).

In Fig. 1, we study how the performance of the implemented algorithms is affected by the thresholds used for controlling the quality of the selected users. For this, we set \((n = 500, k = 100)\), \((n = 1000, k = 200)\) and \((n = 2000, k = 500)\) in Fig. 1(a), Fig. 1(b) and Fig. 1(c), respectively, and scale the value of \(\theta\) from 0.1 to 0.9 with an increment of 0.1 in each sub-figure. It can be seen that PPRC outperforms the other algorithms by outputting smaller ATLs, as PPRC incurs zero value on the penalty function \(Q(\cdot)\). Meanwhile, it is shown that the ATLs output by all algorithms increase when \(\theta\) increases, which can be understood by the fact that more users need to be selected to satisfy equation (1) for a larger \(\theta\).

In Fig. 2, we scale the size of the crowd from 1500 to 3500 with an increment of 200 in each sub-figure, and the values of \(k\) and \(\theta\) also vary, as shown in Fig. 2(a)-Fig. 2(c). Again, it can be seen that PPRC outperforms the other algorithms due to its zero penalty. Besides, the ATLs output by PPRC and BCAA both decrease when \(n\) increases, which can be explained by the reason that the searching space for selecting users enlarges when the crowd size increases, so groups with smaller costs can be found to meet the quality demand. Finally, it can be seen that LDP’s ATL increases with \(n\), as the total losses incurred in the “exploration phase” of LDP increase with the crowd size.

In Fig. 3, we study the performance of the implemented algorithms by scaling the penalty factor \(\varepsilon\) from 10 to 100 under different settings of \(n, k\) and \(\theta\). It can be seen that the relative performance of the implemented algorithms is similar to that in Fig. 1, irrespective of the variation of \(\varepsilon\). Besides, Fig. 3 shows that, the ATL of PPRC is not affected by \(\varepsilon\), while the ATLs of both LDP and BCAA increase with \(\varepsilon\). This is due to the reason that, both LDP and BCAA can output solutions violating constraint (1), while PPRC does not.

In Fig. 4, we scale the number of tasks from 100 to 1000 and test the ATLs of PPRC and LDP, and the values of \(n, k, \theta\) are set as 500, 100 and 0.2, respectively. It can be seen that the performance of LDP gradually approaches PPRC when \(T\) increases. This corroborates the asymptotic regret bound shown in Theorem 10.

Note that the total loss of EXACT or PPRC equals to the value of the optimization function (i.e., the total expected cost) of the PPRC problem, because the output of these two algorithms always satisfy equation (1). Therefore, we are interested in comparing EXACT and PPRC to see the optimality of the solution output by PPRC. As EXACT has exponential time complexity, we have to compare PPRC and EXACT under the cases with small crowd sizes, and the results are shown in Fig. 5. In Fig. 5, we scale the value of \(n\) under different settings of \(k\) and \(\theta\), and each data point is the average of simulation results over 50 data sets. It can be seen from Fig. 5 that the total costs of both EXACT and PPRC decrease when \(n\) increases. This can be explained by the reason similar to that for Fig. 2, i.e., the user groups with smaller costs can be found in a larger searching space with more users. More importantly, Fig. 5 reveals that the performance of PPRC is very close to EXACT, irrespective of the variations of \(n, k\) and \(\theta\). This corroborates the effectiveness of our algorithm.

**X. Conclusion and Discussion**

We have studied a novel pricing problem for economically recruiting participants with reasonable sensing qualities to
achieve sensing robustness in crowdsensing. To address the problem, we have discovered some non-trivial submodular properties of Poisson binomial distributions, and we have proposed an ironing method to transform our problem from a non-submodular optimization problem into a submodular one. Based on these results, we have proposed several approximation algorithms to solve our problem with provable performance bounds, and the experimental results have corroborated the effectiveness of our approach.

As we have adopted a general quality-aware pricing model in this work, our methods could be applied to both crowdsensing systems and crowdsourcing systems. In the future work, we will make further studies on the performance of our algorithms by running our algorithms on a real mobile crowdsensing system, as well as study our problem under an ex-post model. We also plan to investigate the methods for acquiring Bayesian knowledge on the users from the historical data, based on a real mobile crowdsourcing system.

APPENDIX

In this section, we first quote two lemmas that are useful to us, then present our proofs in details.

Lemma 4 (Hoeffding’s Inequality [45]): Let \( Z_1, Z_2, \ldots, Z_n \) be a sequence of random variables with common support \( [0,1] \). If \( \mathbb{E}[Z_i|Z_1, Z_2, \ldots, Z_{i-1}] \leq \psi \) for any \( i \leq n \), then we have \( \Pr(\sum_{i=1}^{n} Z_i \geq n\psi + \ell) \leq e^{-2\ell^2/n} \) for any \( \ell > 0 \).

If \( \mathbb{E}[Z_i|Z_1, Z_2, \ldots, Z_{i-1}] \geq \psi \) for any \( i \leq n \), then we have \( \Pr(\sum_{i=1}^{n} Z_i \leq n\psi - \ell) \leq e^{-2\ell^2/n} \) for any \( \ell > 0 \).

Lemma 5 [43]: Let \( F(\cdot) \) be a non-negative, monotone submodular function defined on \( 2^G \) where \( G \) is the ground set. Given a threshold \( 0 < \Lambda < F(G) \), let \( R^* \) be a subset of \( G \) with the minimum cost that satisfies \( F(R^*) \geq \Lambda \). Let \( R \) be a subset of \( G \) which is computed by a standard greedy algorithm to satisfy \( F(R) \geq \Lambda - \nu(\nu > 0) \). Then the cost of \( R \) is no more than the product of \( R^* \)’s cost and \( 1 + \ln \frac{1}{\nu} \).

Proof of Theorem 1: We prove the NP-hardness of the PPRC problem by a reduction from the NP-complete Partition problem. Given a set of positive integers \( S = \{a_1, \ldots, a_n\} \), the Partition problem asks if \( S \) can be partitioned into two subsets \( S_1 \) and \( S_2 \) such that the sum of numbers in \( S_1 \) equals the sum of numbers in \( S_2 \). Let \( o \) be an arbitrary positive number that is smaller than the minimum number in \( S \). Let \( M = \sum_{i \in [n]} a_i \) and \( z = 1 - \frac{z}{M} \). Consider an instance of the decision version of the PPRC problem where \( \sigma = k = w_1 = 1; u_{i,1} = \frac{\sigma}{a_i}; p_{i,1} = 1 - z^{\frac{1}{2}}; \theta = 1 - z^{\frac{1}{2}} \), and we need to answer the question whether there exist \( X \subseteq [n] \) such that \( \rho_1(X,1) \geq \theta \) and \( \sum_{i \in X} w_1 u_{i,1} \leq 1/2 \). Note that \( 0 < \theta < 1 \) and

\[
0 < \rho_1(X,1) \geq \theta \iff \left( 1 - \frac{M}{M} \right) \frac{1}{a_i} < 1 - \left( 1 - \frac{a_i}{M} \right) \frac{1}{2} <= 0 \frac{a_i}{M} = u_{i,1} < 1,
\]

so the construction of the PPRC problem instance is valid. Moreover, as \( \sum_{i \in X} w_1 u_{i,1} \leq \frac{1}{2} \), \( \sum_{i \in X} u_{i,1} \leq \frac{1}{2} \) and \( \sum_{i \in [n]} a_i \),

\[
\rho_1(X,1) \geq \theta \iff 1 - \prod_{i \in X} (1 - p_{i,1}) \geq 1 - z^{\frac{1}{2}} \Rightarrow \prod_{i \in X} \frac{a_i}{2} \Rightarrow \sum_{i \in X} a_i \geq \frac{1}{2} \sum_{i \in [n]} a_i,
\]

it can be seen that solving this PPRC problem instance is equivalent to solving the Partition problem. Therefore, the PPRC problem is NP-hard.

Proof of Lemma 2: Clearly, the lemma holds for \( c = 0 \) or \( a = b = 0 \). So we only consider the case that \( 0 < c \leq a < b \leq |X| \).

For any \( C \in \mathcal{W}(X,b,a-c) \), there must exist \( C_1 \in \mathcal{F}_X^c \) and \( C_2 \in \mathcal{F}_X^{n-c} \) such that \( C = C_1 \cup C_2 \). Suppose that \( |C_1 \cap C_2| = h \). As \( |C_1 \setminus C_2| = b - h \geq a - h \), we can find a subset \( D \subseteq C_1 \setminus C_2 \) such that \( |D| = a - h \). Let \( C' = (C_1 \cup C_2) \setminus D \) and \( C'' = D \cup (C_1 \cap C_2) \), so we know

\[
|C'| = |C_1| + |C_2| - |C_1 \cap C_2| - |D| = b + (a-c) - h - (a-h) = b - c
\]

and \( |C''| = |D| + |C_1 \cap C_2| = a - h + h = a \). Meanwhile, we have \( C_1 \cup C_2 = C_1 \cup C_2 \). This means that \( C' \subseteq \mathcal{F}_X^{n-c} \) and \( C'' \subseteq \mathcal{F}_X^c \), and hence \( C \subseteq \mathcal{W}(X,b,a-c) \). According to the above reasoning, we get

\[
\forall C \in \mathcal{W}(X,b,a-c), \quad |g(C)| \leq a - c
\]

One the other hand, for any \( C \in \mathcal{W}(X,b,a-c) \), we have

\[
g(C) = \frac{(a+b-c-2z)!}{(b-z)!(a-c-z)!} \cdot \frac{(a-z)!}{(b-c-z)!} = \prod_{i=1}^{b-a} \frac{a-c-z+i}{b-z} \geq \frac{a+b-c-2z}{b-z},
\]}

where \( z = |g(C)| \). Recall that \( 0 < c \leq a \leq b \leq |X| \). So we have \( b - c > 0, a > b > 0 \) and \( a - c \geq 0 \). Combining equation (7), (8) and Lemma 1 gives us:

\[
\Phi_1(b-c, X) \cdot \Phi_1(a, X) = \prod_{i \in \mathcal{C}} \left( \frac{a+b-c-2|g(C)|}{b-c-|g(C)|} \right) \prod_{i \in \mathcal{C}} \beta_{i,t}
\]

\[
\geq \prod_{i \in \mathcal{C}} \left( \frac{a+b-c-2|g(C)|}{b-c-|g(C)|} \right) \prod_{i \in \mathcal{C}} \beta_{i,t}
\]

\[
\geq \sum_{i \in \mathcal{C}} \left( \frac{a+b-c-2|g(C)|}{b-|g(C)|} \right) \prod_{i \in \mathcal{C}} \beta_{i,t}
\]

hence the lemma follows.

Proof of Lemma 3: When \( a = 0 \), the lemma is easily proved as \( \Phi_1(0,X) = \Phi_1(0,Y \setminus X) = 1 \). Next, we consider
the case that $a > 0$. Let $h = \min\{a, |Y| - |X|\}$. Let $K_i = \{C \cup D | C \in \mathcal{F}^{-1}_X \land D \in \mathcal{F}^{-1}_Y \} \land K_i = \{(C, D) | C \in \mathcal{F}^{-1}_X \land D \in \mathcal{F}^{-1}_Y \}$ for any $i \in \{0, 1, \ldots, h\}$. So there exists a bijective function $f : K_i \to K_i$ defined as $f(C, D) = C \cup D$. Meanwhile, it can be seen that $K_i \cap K_j = \emptyset$ for any $i \neq j$ and $\bigcup_{i=0}^h K_i = \mathcal{F}_Y$. Therefore, we have

$$
\Phi_l(a - i, X)\Phi_l(i, Y \setminus X) = \sum_{C \in \mathcal{F}^{-1}_X} \left( \prod_{j \in C} \beta_{j,l} \right) \sum_{D \in \mathcal{F}^{-1}_Y \cup X} \left( \prod_{j \in D} \beta_{j,l} \right)
= \sum_{(C, D) \in K_i} \left( \prod_{j \in C \cup D} \beta_{j,l} \right) = \sum_{A \subseteq K_i} \left( \prod_{j \in A} \beta_{j,l} \right) \quad (9)
$$

and

$$
\Phi_l(a, Y) = \sum_{A \subseteq K_i} \left( \prod_{j \in A} \beta_{j,l} \right) = \sum_{i=0}^h \sum_{A \subseteq K_i} \left( \prod_{j \in A} \beta_{j,l} \right) \quad (10)
$$

The lemma then follows by combining (9) and (10). □

**Proof of Theorem 4:** For simplicity, we abuse the notations a little in this proof by setting $\Psi(X) = \Psi_l(X, \mu_1, \mu_2)$ and $\Gamma(X) = \Gamma_l(X, \mu_1, \mu_2)$. We first prove that $\Psi(\cdot)$ is submodular. Given any $X \subseteq Y \subseteq \{n\}$ and any $x \in \{n\} \setminus Y$, we discuss the following cases according to the definition of $\Psi$:

1) When $|X| < \mu_1 - 1$ and $|Y| < \mu_1 - 1$.
   In this case, we have $\Psi(x|X) = \Psi(x|Y) = \alpha_l$.

2) When $|X| < \mu_1 - 1$ and $|Y| = \mu_1 - 1$.
   In this case, we have
   $$
   \Psi(x|X) = \alpha_l \geq \alpha_l + \ln \Gamma(Y \cup \{x\}) = \alpha_l + \ln \Gamma(Y) + \ln \Gamma(Y \cup \{x\}) = \Psi(x|Y).
   $$

3) When $|X| < \mu_1 - 1$ and $|Y| \geq \mu_1$.
   In this case, we have
   $$
   \Psi(x|X) = \alpha_l \geq 2\ln \delta_l
   \geq -\ln \delta_l \geq -\ln \Gamma(Y)
   \geq \ln \Gamma(Y \cup \{x\}) - \ln \Gamma(Y) = \Psi(x|Y).
   $$

4) When $|X| = \mu_1 - 1$ and $|Y| \geq \mu_1$.
   In this case, we have
   $$
   \Psi(x|X) = \mu_1 \alpha_l + \ln \Gamma(X \cup \{x\}) = \alpha_l(\mu_1 - 1)
   \geq -2\ln \delta_l + \ln \Gamma(X \cup \{x\}) = \ln \Gamma(X \cup \{x\}) - \ln \Gamma(Y)
   \geq -\ln \delta_l \geq -\ln \Gamma(Y)
   \geq \ln \Gamma(Y \cup \{x\}) - \ln \Gamma(Y) = \Psi(x|Y).
   $$

5) When $|X| \geq \mu_1$.
   In this case we can directly use Theorem 3 to get $\Psi(x|X) \geq \Psi(x|Y)$.

Therefore, we have proved that the function $\Psi$ is submodular.

Clearly, $\Psi(X)$ is non-negative when $|X| < \mu_1$. When $|X| \geq \mu_1$, we also have

$$
\Psi(X) \geq \alpha_l + \ln \Gamma(X) \geq 2\ln \delta_l + \ln \Gamma(X)
\geq \ln \Gamma(X) - \ln \delta_l \geq 0.
$$

Hence $\Psi$ is always non-negative.

Finally, for any $|X_1| < \mu_1$ and $|X_2| \geq \mu_1$, we have

$$
\Psi(X_1) = \alpha_l |X_1| \leq \alpha_l(\mu_1 - 1)
\leq \alpha_l(\mu_1 - 1) + \ln \Gamma(X_2) - \ln \delta_l
\leq \alpha_l(\mu_1 - 1) + \ln \Gamma(X_2) - 2\ln \delta_l
\leq \alpha_l(\mu_1 - 1) + \ln \Gamma(X_2) + \alpha_l = \Psi(X_2)
$$

So the theorem follows.

**Proof of Theorem 6:** In the proof, we assume that $n > k$ and $p_i(\lceil n \rceil, k) \geq \theta$ and $\delta_l \leq \theta$ where $\delta_l = \min\{\Gamma_l(X, k, n) | X \subseteq \{n\} \land |X| \geq k\} = \prod_{1 \leq i \leq k} p_{s_i, i}$, because otherwise the PPRC problem can be trivially solved for the given $l$

Clearly, $\phi_l(0) = 0$. Note that $\Gamma_l(X, k, n) = \rho_l(X, k, n)$, which is non-decreasing. Hence, we can easily know that $\frac{\rho_l + \delta_l}{\lambda_l} \Psi_l(\cdot, k, n)$ is a polymatroid function according to Theorem 4. As $\phi_l(X)$ is the minimum of $\frac{\rho_l + \delta_l}{\lambda_l} \Psi_l(X, k, n)$ and the constant $\frac{\rho_l + \delta_l}{\lambda_l} (\alpha_l + \ln \theta)$, it can be verified that $\phi_l(\cdot)$ is also a polymatroid function.

As $p_l(\lceil n \rceil, k) \geq \theta$, we have $\phi_l(X_{\text{opt, } l}) = \phi(\lceil n \rceil)$. Meanwhile, as $n > k$, we have $\delta_k \leq \rho_l(\lceil n \rceil \setminus \{s_1\}, k)$ and hence

\begin{align*}
\alpha_l \geq & -2\ln \delta_k + \ln \delta_l \geq \ln \rho_l(\lceil n \rceil \setminus \{s_1\}) + \ln \delta_l \\
\geq & \ln \rho_l(\lceil n \rceil, k) - \ln \rho_l(\lceil n \rceil \setminus \{s_1\}, k) \geq \lambda_l
\end{align*}

So we get:

$$
\phi_l(\lceil n \rceil) \geq \frac{\rho_l + \delta_l}{\lambda_l} \alpha_l(\mu_1 + \ln \theta) \geq \frac{\rho_l + \delta_l}{\lambda_l} (\alpha_l + \ln \theta)
\geq \frac{\rho_l + \delta_l}{\lambda_l} (\ln \delta_k - \ln \theta + \ln \theta) \geq \rho_l + \delta_l
$$

where (11) is due to $\ln \delta_k < \ln \theta$. This ends the proof for 1).

To prove 2), we discuss the following cases according to the definition of $\phi_l$:

1) When $|X| < k - 1$.
   In this case, as $\alpha_l + \ln \theta \geq -\ln \delta_l \geq 0$, we have
   $$
   \Psi_l(X, k, n) \leq \Psi_l(X \cup \{x\}, k, n) \leq \alpha_l(k - 1)
   \leq \alpha_l(k - 1) + \alpha_l + \ln \theta
   = \alpha_l k + \ln \theta
   $$

   Therefore
   $$
   \phi_l(x|X) = \frac{\rho_l + \delta_l}{\lambda_l} \alpha_l \geq \rho_l + \delta_l \geq \rho_l + \delta_l
   $$

2) When $|X| = k - 1$.
   In this case, we have either
   $$
   \alpha_l + \ln \rho_l(\lceil X \cup \{x\}, k \setminus k\rceil, k)
   \geq \alpha_l k + \ln \rho_l(\lceil X \cup \{x\}, k \setminus k\rceil, k) - \ln \delta_l - \ln \delta_l
   \geq -\ln \delta_l \geq \lambda_l
   $$
or
\[
\alpha_l + \min\{\ln \rho_l(X \cup \{x\}, k), \ln \theta\} = \alpha_l + \ln \theta \geq \ln \theta - 2\ln \delta_l \\
\geq \ln \theta - \ln \delta_l - \ln \theta \\
\geq -\ln \delta_l \geq \lambda_l
\]

So we get
\[
\phi_l(x|X) \geq \frac{\text{opt}_l + d_l}{\lambda_l} + \min\{\ln \rho_l(X \cup \{x\}, k), \ln \theta\} \geq \frac{\text{opt}_l + d_l}{\lambda_l} \geq d_l
\]

3) When $|X| \geq k$:

In this case, if $\rho_l(X \cup \{x\}, k) < \theta$, we have:
\[
\phi_l(x|X) = \frac{\text{opt}_l + d_l}{\lambda_l} \ln \left(\frac{\rho_l(X \cup \{x\}, k)}{\rho_l(X, k)}\right)
\]

As $|X| < n$ and $x \notin X$, there must exist $Y = [n]\{x\}$ such that $X \subseteq Y$. According to Theorem 3, we know that
\[
\ln \rho_l(X \cup \{x\}, k) - \ln \rho_l(X, k) \geq \ln \rho_l([n]\{s_1\}, k) - \ln \rho_l(Y, k)
\]

Recall that $p_i \leq p_{i+1} \leq \ldots \leq p_{n-1}$, so for any $Q \subseteq [n]$ with cardinality $n-1$, we must have
\[
\rho_l(Q, k) \leq \rho_l([n]\{s_1\}, k).
\]

So we get
\[
\ln \rho_l([n]\{s_1\}, k) - \ln \rho_l(Y, k) \geq \ln \rho_l([n]\{s_1\}, k) - \ln \rho_l([n]\{s_1\}, k) \geq \lambda_l
\]

Combining equations (12)-(14), we get
\[
\phi_l(x|X) \geq \text{opt}_l + d_l \geq d_l
\]

On the other hand, when $\rho_l(X \cup \{x\}, k) \geq \theta$, we also have
\[
\phi_l(x|X) = \frac{\text{opt}_l + d_l}{\lambda_l} \ln \theta - \ln \rho_l(X, k) \\
\geq \frac{\text{opt}_l + d_l}{\lambda_l} \lambda_l = \text{opt}_l + d_l \geq d_l
\]

Until now, we have proved 2) under all the different cases. Hence the theorem follows.

Proof of Theorem 9: If line 5 is executed in Algorithm 3, then the theorem trivially holds. Otherwise, we can directly use Lemma 5 to prove the theorem (by setting the variables $\Lambda$ and $\nu$ in Lemma 5 to $\eta$ and $\varepsilon_1$ appeared in line 8 of Algorithm 3, respectively), as we can see from Theorem 4 that $\Psi_l(k, n)$ is a monotone and submodular function defined on $[2n]\{s\}$.

Proof of Theorem 10: Let $g = \sigma \sqrt{\ln \frac{2}{\delta}}$. Given any $(i, l) \in [n] \times [\sigma]$, using Lemma 4 we can get
\[
\Pr\{\bar{u}_{i,l} - u_i, l \geq G\} = \Pr\{\bar{u}_{i,l} - u_i, l \geq \sqrt{\ln g / 2g}\} \\
\leq 2/g = O(T^{-\frac{\delta}{2}})
\]

Similarly, we have
\[
\Pr\{\bar{p}_{i,l} - p_i, l \geq G\} = O(T^{-\frac{\delta}{2}}).
\]

Let $\bar{i}_l = \max\{|\bar{p}_{i,l} - p_i, l|, |\bar{u}_{i,l} - u_i, l|\}$. Using the union bound, we can get
\[
\Pr\{\exists (i, l) \in [n] \times [\sigma] : \bar{i}_l \geq G\} \leq O(T^{-\frac{\delta}{2}})
\]

Note that $\lim_{T \to \infty} G = 0$ and $\Psi_l(X)$ (or $\rho_l(X, k)$) is continuous with respect to any variable in $\{u_{i,l}, p_{i,l}|i \in [n], l \in [\sigma]\}$ for any given $(X, l)$. According to line 11-12 of Algorithm 4 and equation (16), when $T$ is sufficiently large, we must have
\[
\Pr\{C(\bar{X}, l) \leq C(Y, e)\} \\
\geq \Pr\{\forall (i, l) \in [n] \times [\sigma] : \bar{i}_l \leq G\} \geq 1 - O(T^{-\frac{\delta}{2}}),
\]

and
\[
\rho_l(\bar{X}, k) > \theta - Q^{-1}(\varepsilon C(Y, e)),
\]

where $(Y, e)$ is the output of Algorithm 2 and $\varepsilon$ is the given parameter in the statement of Theorem 10. This implies $Q(\max\{\max(\theta - \rho_l(\bar{X}, k), 0)\}) \leq \varepsilon C(Y, e)$. Therefore, the regret of Algorithm 4 in the exploitation phase can be upper-bounded by
\[
E\{\sum_{t=1}^{T=\sigma+1} (\bar{L}(X_t, l_t) - (1 + \varepsilon)\kappa \cdot \bar{L}(X^*, l^*))\} \\
= E\{\bar{L}(X, l) - (1 + \varepsilon)\kappa \cdot \bar{L}(X^*, l^*)\} \cdot (T - O(\sqrt{T})) \\
\leq E\{C(\bar{X}, l) + Q(\max\{\max(\theta - \rho_l(\bar{X}, k), 0)\}) \cdot (T - O(\sqrt{T})) \\
= (1 + \varepsilon)C(Y, e) \cdot (T - O(\sqrt{T})) \\
\leq \Pr\{C(\bar{X}, l) > C(Y, e)\} \cdot (T - O(\sqrt{T})) \\
\leq (\sum_{i \in [n]} w_i \cdot u_{i, \sigma}) \cdot O(T^{-\frac{\delta}{2}}) \cdot (T - O(\sqrt{T})) \\
= O(\sqrt{T})
\]

Meanwhile, the regret of Algorithm 4 in the exploitation phase can be upper-bounded by
\[
E\{\sum_{t=1}^{T=\sigma} (\bar{L}(X_t, l_t) - (1 + \varepsilon)\kappa \cdot \bar{L}(X^*, l^*))\} \\
\leq E\{\sum_{t=1}^{T=\sigma} \bar{L}(X_t, l_t)\} \\
\leq (\sum_{i \in [n]} w_i \cdot u_{i, \sigma}) + Q(\theta) \cdot O(\sqrt{T}) \\
= O(\sqrt{T})
\]

The theorem then follows by combining (17) and (18). □

Proofs of Lemma 1, Theorem 2 and Theorem 3: Due to the space constraint, these proofs are omitted and can be found in [46]. □

References
