Pricing Mechanisms for Crowd-Sensed Spatial-Statistics-Based Radio Mapping

Xuhang Ying, Student Member, IEEE, Sumit Roy, Fellow, IEEE, and Radha Poovendran, Fellow, IEEE

Abstract—Networking on white spaces (i.e., locally unused spectrum) relies on active monitoring of spectrum usage. Spectrum databases based on empirical radio propagation models are widely adopted but shown to be error-prone, since they do not account for built environments like trees and man-made buildings. As an economically viable option, crowd-sensed radio mapping acquires more accurate local spectrum data from mobile users and constructs radio maps using spatial models such as Kriging and Gaussian Process. Success of such crowd-sensing systems presumes some incentive mechanisms to attract user participation. In this work, we consider the scenario where the platform who constructs radio environment maps makes one-time offers to selected users, and collects data from those who accept the offers. We design pricing mechanisms based on expected utility (EU) maximization, where EU captures the tradeoff between radio mapping performance (location and data quality), crowd-sensing cost and uncertainty in offer outcomes (i.e., possible expiration and rejection). Specifically, we consider sequential offering, where one best price offer is sent to the best user in each round, and batched offering, where a batch of multiple offers are made in each round. For the later, we show that EU is submodular in the discrete domain, and propose a mechanism that first fixes the pricing rule and selects users based on Unconstrained Submodular Maximization (USM); it then compares different pricing rules to find the best batch of offers in each round. We show that USM-based user selection has provable performance guarantee. Proposed mechanisms are evaluated and compared against utility-maximization-based baseline mechanisms.

Index Terms—Radio Environment Mapping, Spatial Statistics, Crowd-Sensing, Pricing Mechanism, Expected Utility Maximization, Unconstrained Submodular Maximization.

I. INTRODUCTION

The exponentially increasing mobile data traffic has translated into a greater demand for wireless network capacity and more spectrum for broadband use. A significant portion of spectrum is allocated for various services (e.g., TV broadcasting, radar and satellite services), but often underutilized in practice. To improve spectrum utilization, the FCC allows unlicensed users to opportunistically access locally idle licensed spectrum, aka White Spaces (WS), subject to the no-harmful-interference constraint [2].

A key step of WS networking is to actively monitor spectrum usage and identify WS opportunities. Currently, empirical radio propagation models are widely used and implemented in spectrum databases [3]–[5] for this purpose, but recent studies [7]–[9] have shown that they are often locally inaccurate since they do not account for built environments like trees and man-made buildings. To augment spectrum databases, radio mapping based on spatial statistics (e.g., Kriging [9]–[11] and Gaussian Processes (GP) [12], [13]) has been proposed, which leverages local RSSI data and provides accurate local RSSI estimation at unmeasured locations.

To collect data for spatial statistics, measurement campaigns like drive tests are usually required, but they may be time- and labor-intensive for continuously monitoring a wide area. Deploying sensors is a second option, but large-scale deployment can be costly due to expensive hardware. An economically viable alternative is crowd-sensing, that is, outsourcing sensing tasks to mobile users who have devices that are equipped with spectrum sensors. However, since users consume resources (e.g., battery, CPU and memory) for sensing, they need to be properly compensated or incentivized.

Crowd-sensed radio mapping is different from other crowd-sensing applications [14]–[16] in several ways. First, it aims to sample the (unknown) RSSI field only at reported user locations at a time instant (or within a short duration) and applies spatial statistics to estimate RSSI values at unmeasured locations. For radio mapping, it is not necessary to specify targeted sensing locations, and the relative positioning of selected user locations matters. Second, user devices are heterogeneous and provide data of different quality (i.e., hardware noise). It needs to be taken into account by the spatial model. Third, user devices are not dedicated to sensing and busy with many other tasks simultaneously. Hence, in additional to energy or battery costs, users will incur opportunity costs depending on current device statuses, which is the loss of potential gain when they decide to spend resources on sensing instead of other tasks.

In this work, we consider pricing [17]–[19] in an offline setting for crowd-sensed radio mapping. Given a spatial model (or RSSI data model), the platform (i.e., the party who acquires RSSI data) determines the value of a set of users based on location, data quality and it own preferences, which is the amount of money the platform is willing to pay. Each user has a private sensing cost (i.e., sum of energy and opportunity costs). Selected users receive one-time price offers and have only one chance to make a decision, either accept or reject. Here, we consider rational users who accept offers if its sensing cost is no greater than the offered price. Those who accept offers perform sensing at reported locations, upload data and receive payments. Therefore, from the platform’s perspective, a set of users is associated with a utility or gain.

Received Signal Strength Indicator
which is the difference between value and total payment. Due to possible offer expiration (due to network congestion etc.) and rejection, the platform aims to maximize the expected utility (EU) by determining who to send offers to (i.e., user selection) and how much to offer (i.e., price determination).

Our primary contributions are as follows:

- We design a crowd-sensing system that periodically acquires spectrum data from users for radio mapping.
- We introduce EU and formulate pricing mechanism design as EU maximization. We first propose sequential offering, where the platform sends out the best offer to the best user in each round, and keeps offering until the next one is no longer profitable. Then we generalize it to batched (i.e., single-batch and multi-batch) offering, where a batch of multiple offers are made in each round.
- For batched offering, we show that EU is submodular in the discrete domain. We propose a pricing mechanism that first fixes the pricing rule, and selects users based on Unconstrained Submodular Maximization (USM); it compares different pricing rules to find the best batch of offers in each round. We show that our USM-based user selection that takes the estimated EU (via the Monte-Carlo method), instead of the true EU (which is difficult to evaluate) as input, obtains a solution whose true EU value is no less than one third of the true EU value of the optimal solution minus a term due to estimation errors (which grows linearly in the number of users given the maximum estimation error).
- We conduct simulations to evaluate the proposed mechanisms and compare them against baseline mechanisms based on (best-case) utility maximization. Our results show that with a single batch, the proposed mechanism is better than the baseline mechanism with an improvement ranging from 8.5% to 40.5%. If more batches are permitted (and unlimited), the proposed mechanism has close performance with the baseline mechanism, but requires much fewer batches (2.5 vs. 7.7 batches on average) and thus a much smaller delay. Sequential offering is easy to implement and works better than the single-batch baseline mechanism, but has a very large cumulative delay. Offer expiration adversely affected all mechanisms, but sequential and batched offering are more robust.

The rest of this paper is organized as follows. We review related work in Section II and provide a two-user tutorial example in Section III. In Section IV, we provide background on submodularity and present our models. Our pricing mechanisms are presented in Section V and evaluated in Section VI. We conclude this study in Section VII.

II. RELATED WORKS

In recent years, spatial-statistics-based radio mapping has been proposed to better capture local radio environments to augment spectrum databases. In [20], Phillips et al. applied a statistical interpolation technique called Ordinary Kriging to map the coverage of WiMax networks. Similar techniques have been applied to estimate the coverage area of single-transmitter [9] and multi-transmitter networks [11] in TV bands. A more detailed discussion is available in [21].

Radio mapping requires a large amount of sensing data, and incentivized crowd-sensing is considered as an economically viable option. A number of various incentive mechanisms have been proposed based on different models, such as Stackelberg game [14], reverse auction [14], [22], [23], all-pay auction [24], Bayesian models [15], Tullock contests [25] and posted pricing [17], [19], [26]. Some are proposed in an online setting with constraints like budget limits [26]–[28], where users arrive in a random order and the goal is to maximize a certain objective (e.g., revenue).

Incentive mechanisms are typically designed to account for requirements of the crowd-sensing application being considered, such as user locations, data quality and user availability. For example, in [22], each task has a specific location tag and each user can only compete for tasks within its service region. In [16], Peng et al. extended the well-known Expectation Maximization algorithm to estimate the quality of sensing data and incorporated it in determining rewards. In [19], Han et al. studied a quality-aware Bayesian pricing problem where both users’ sensing costs and qualities are random variables, drawn from known distributions. The goal is to choose an appropriate posted price to recruit a group of users with reasonable sensing quality, and minimize the total expected payment. If users need to move to designated sensing locations or are available at different time periods, then incentive mechanism design is closely coupled with task allocation [18] or scheduling [29].

In this study, we consider incentive mechanism design in an offline setting specifically for crowd-sensed radio mapping. In our system, the platform acquires data periodically from users who are readily available for sensing, and measurements are taken at their current locations. Besides, we consider data quality in terms of hardware noise, and incorporate it into the spatial model (i.e., GP). Different from the auction-based incentive mechanism for crowd-sensed radio mapping [23], we are interested in pricing mechanisms, where the platform makes one-time price offers to a set of selected users, and collects data from those who accept offers. To select users and determine corresponding price offers, we define utility for the platform to trade the value (which reflects radio mapping performance) against the total price (crowd-sensing cost), and further extend it to EU to account for possible offer expiration and rejection. We formulate the pricing mechanism design as EU maximization, and propose mechanisms based on USM.

III. A TWO-USER TUTORIAL EXAMPLE

Before presenting our general model, we first present our system architecture and provide a two-user example to illustrate the basic idea of pricing for crowd-sensed radio mapping.

A. System Architecture

As shown in Fig. 1, the platform acquires data periodically from users. At the beginning of each period, the platform broadcasts a sensing task to all users in the area of interest (AoI) with specific sensing parameters (e.g., center frequency, sampling rate and FFT bin size) to ensure a consistent sensing procedure across different hardware. Note that the task does not specify sensing locations with twofold reasons. First,
there is no need to do so, since the platform will take a posteriori sampling approach to select users. It is the relative positioning of measurements instead of exact sensing locations that matters for radio mapping. Second, it requires extra time and costs for users to move to target sensing locations, which means extra incentivization costs for the platform and added complexity for mechanism design.

To avoid excessive delay due to communication delay or failure, network congestion etc., each offer has a deadline, by which a decision has to be received by the platform (along with the data if accepted); otherwise, the offer will be expired.

In this study, we assume no entry or other overhead costs, that is, a user does not incur a fee to communicate with the platform. We consider users of low mobility (e.g., pedestrians), who are honest in providing their information and following the protocol. We assume small displacements between reported locations and eventual sensing locations. We will leave the high-mobility case and security considerations as future work.

B. A Two-User Scenario

In this example, the platform wants to estimate the RSSI $Z(x_0)$ at location $x_0$. There are two users $S = \{1, 2\}$ at $x_1$ and $x_2$ in the AoI. In each period, each user will incur a sensing cost $c_i > 0$ and receive an offer $p_i > 0$, when selected by the platform. We assume that users are rational and will accept the offer if $c_i \leq p_i$, and reject it otherwise. In the following discussion, we assume no expired offers and will consider them later in Section IV-A.

The platform has a valuation function $v : 2^S \rightarrow \mathbb{R}_+$ and a pricing function $p : 2^S \rightarrow \mathbb{R}_+$. For each set of users $A$, there is an associated value $v(A)$ and a total price of all offers $p(A) = \sum_{i \in A} p_i$, (i.e., crowd-sensing cost for the platform), assuming that each offer is unexpired and accepted. It makes sense for the platform, as a rational decision maker and company, to maximize its utility (or profit), i.e.,

$$\max_{A \subseteq S} u(A) = \max_{A \subseteq S} (v(A) - p(A)), \quad (1)$$

where $\mu$ is the path-loss-impaired RSSI (unknown but constant), $\delta(x_i) \sim N(0, \sigma_1^2)$ is shadowing and $\epsilon_i \sim N(0, \sigma_2^2)$ is hardware noise. Since shadowing is a spatial phenomenon, $\{\delta(x_i)\}$ are correlated with each other and RSSIs at two closer locations are more correlated (i.e., a larger covariance), while $\{\epsilon_i\}$ are independent of each other and of shadowing.

Suppose that Ordinary Kriging \cite{2} is used for RSSI estimation. With either user, we have $Z(x_0) = \hat{Z}(x_i)$, where $\hat{Z}(x_0)$ is the estimated value of $Z(x_0)$. If both are selected, we have $\hat{Z}(x_0) = \omega_1 \hat{Z}(x_1) + \omega_2 \hat{Z}(x_2)$, where $\{w_i\}$ are weights\cite{2} and $\omega_1 + \omega_2 = 1$. The popular metric called (minimized) mean squared error (MSE) $\mathbb{E}[(\hat{Z}(x_0) - Z(x_0))^2]$ is used to quantify the performance, and a smaller MSE implies a better estimator.

To translate MSE to value that is comparable to payments, the platform adopts the following valuation function,

$$v(A) = \alpha_1 \cdot \log(1 + \alpha_2 - \text{MSE}(A)), \quad (3)$$

where $\alpha_1, \alpha_2$ are constants and $\alpha_2 - \text{MSE}(A) \geq 0$ for any $A \subseteq S$. Since MSE is undefined for $A = \emptyset$, we set $v(\emptyset) = 0$. Intuitively, $v(\cdot)$ captures the diminishing returns behaviors: adding a new element increases $v$ more, if there are fewer elements so far, and less, if there are more elements. This is further formalized by the notion of submodularity, which will be elaborated in Section IV-A.

We consider two cases under this model:

\cite{18} for more discussions on allocation of tasks with specific locations.
evaluating the task and current device status\(^4\). We assume that the platform has only \textit{a priori} probabilistic knowledge of \(C_i\). Let \(f_{C_i}(c_i)\) and \(\bar{F}_{C_i}(\bar{c}_i)\) be the probability density function (PDF) and corresponding cumulative density function (CDF), respectively. The PDF and \(c_i, \bar{c}_i\) could be learned by the platform from the empirical distribution out of prior cost declarations by users of the same device type, or from its long-term interaction with users (e.g., whether or not accept an offer with a known price). If such prior information is absent, \(C_i\) may be assumed to be uniformly distributed over \([\tilde{c}_i, \bar{c}_i]\). Hence, it is reasonable to assume that the platform can infer \(f_{C_i}(c_i)\) or \(\bar{F}_{C_i}(\bar{c}_i)\) based on the reported device type.

Since the platform does \textit{not} know \(\{c_i\}\), it needs to consider possible offer rejections. For \(A = \{1\}\) or \(\{2\}\), the uncertainty in user decisions implies the following utility,

\[
u_{i4} = \begin{cases} v_{i4}^* - p_{i4}, & \text{if user } i \text{ accepts the offer} \\ 0, & \text{otherwise} \end{cases}
\]

which is a Bernoulli random variable and the acceptance probability is \(\Pr(c_i \leq p_{i4}) = \int_{0}^{p_{i4}} f_{C_i}(c_i)\, dc_i = F_{C_i}(p_{i4})\). In this case, it makes more sense to consider the EU,

\[
EU_{i4} = \mathbb{E}[\nu_{i4}] = (v_{i4} - p_{i4})F_{C_i}(p_{i4}),
\]

and the platform wants to find \(p_{i4}^*\) that maximizes \(EU_{i4}\), i.e., \(EU\) maximization. For \(A = \{1, 2\}\), the EU is given by

\[
EU_{i12} = (v_{i12} - p_{i1} - p_{i2})F_{C_i}(p_{i1})F_{C_j}(p_{i2}) + (v_{i1} - p_{i1})F_{C_i}(p_{i1})(1 - F_{C_j}(p_{i2})) + (v_{i2} - p_{i2})(1 - F_{C_i}(p_{i1}))F_{C_j}(p_{i2}).
\]

and the goal is to find \(p^* = [p_{i1}^*, p_{i2}^*]\) that maximizes \(EU_{i12}\). Note that the platform does not have to send out offers all at once and stop (i.e., single-batch offering); it can send more batches of backup offers (i.e., multi-batch offering) based on the knowledge of outcomes of previous offers. For illustration, we set \(C_1 \sim U[4, 7]\) and \(C_2 \sim U[5, 6]\), where \(U[\cdot, \cdot]\) represents the uniform distribution.

For pricing, the platform’s first thought could be setting \(p_{i4}^* = \bar{c}_i\), and the rest is the same with the deterministic-cost case. A natural generalization is to choose a desired probability of acceptance\(^5\) \(\gamma \in [0, 1]\) and set \(p_{i4}^* = F_{C_i}^{-1}(\gamma)\) for each user \(i\), where \(F_{C_i}^{-1}(\cdot)\) is the inverse CDF. Given \(\gamma\), prices are fixed and the platform wants to maximize the EU. Taking Case 2 of Model II and \(\gamma = 0.95\) as an example, we have \(p_{i1} = 6.85, p_{i2} = 5.95, p_{i12} = 12.8\), and \(A^* = \{2\}\) is the best with \(EU_{i2} = 3.78\) by Eq. (5) and (6). Note that the platform may further consider user \(i\) if user 2 rejects the offer. Then the overall EU with multi-batch offering would be \(EU_{i12} + (1 - \gamma)EU_{i1} = 3.91 > EU_{i2}\). \(^4\)

\(^5\) In practice, we would expect a crowd-sensing application to be installed and running on users’ mobile devices, which has some function that estimates the perceived sensing cost in each period based on the needed resources for the sensing task and the current device status. Designing such a function for sensing cost estimation will be of practical importance, not only to this work, but also to many other crowd-sensing applications (e.g., [15]). But this topic is out of the scope of this paper, and will be left as future work.

\(^6\) Compared to choosing the same desired price for all users, it makes more sense to choose the same desired probability for all users, since users have different cost distributions in general. If all user devices are of the same type, then these two approaches are the same.

\begin{table}[h]
\centering
\begin{tabular}{|c|c|c|c|c|}
\hline
 & \(v(0)\) & \(v(1)\) & \(v(2)\) & \(v(\{1, 2\})\) \\
\hline
Model I & 0 & 10 & 10 & 10 \\
Model II - Case 1 & 0 & 10.90 & 9.56 & 11.49 \\
Model II - Case 2 & 0 & 9.56 & 9.56 & 11.49 \\
\hline
\end{tabular}
\caption{Valuation of users in the two-user example.}
\end{table}
Given $A$, $\gamma$ can also be optimized in each batch. Taking Case 2 of Model II as an example, when $A = \{1\}$, $EU(1) = (9.56 - (4 + 3\gamma))\gamma$ and $\gamma^* = 0.93$, $EU^*(1) = 2.58$. Similarly, for $A = \{2\}$, $EU(2) = (9.93 - (5 + \gamma))\gamma$ and $\gamma^* = 1$, $EU^*(2) = 3.93$. When $A = \{1, 2\}$, we have $EU_{\{1, 2\}} = -8.09\gamma^2 + 10.49\gamma$ and $\gamma^* = 0.65$, $EU^*_{\{1, 2\}} = 3.40$. In this case, instead of using the same $\gamma$, the platform can also choose $\{\gamma_i\}$ for each user separately, and maximizing Eq. (6) leads to $\gamma^*_1 = 0, \gamma^*_2 = 1$.

As we can see, the notation of utility accounts for locations, data quality and sensing costs, and the notation of EU further considers possible offer rejections. We can also see that user selection and price determination are closely coupled in a pricing mechanism. More will be discussed in Section IV.

### IV. Preliminaries and Our Model

In this section, we first provide brief background on submodularity. Then we present our RSSI data model and define the metric for measuring radio mapping performance. Finally, we present our valuation model and explore its properties.

**A. Preliminaries**

The submodularity property is formally defined as follows.

**Definition 1 (Submodularity).** Let $\Omega$ be a finite set. A function $f : 2^\Omega \mapsto \mathbb{R}$ is submodular if for any $A, B \subseteq \Omega$,

$$ f(A) + f(B) \geq f(A \cup B) + f(A \cap B). $$

Equivalently [32], a function $f$ is submodular if, for any $A \subseteq B \subseteq \Omega$ and any $i \in \Omega \setminus B$,

$$ f(A \cup \{i\}) - f(A) \geq f(B \cup \{i\}) - f(B). $$

The notion of submodularity captures diminishing returns behaviors: adding a new element increases $f$ more, if there are fewer elements so far, and less, if there are more elements.

**Definition 2 ((Approximately) monotonic function).** Let $\Omega$ be a finite set. A function $f : 2^\Omega \mapsto \mathbb{R}$ is said to be monotone (or monotonic), if $f(A \cup \{i\}) - f(A) \geq 0$ for any $A \subseteq \Omega$ and any $i \in \Omega \setminus A$; $f$ is said to be $\alpha$-approximately monotonic, if $f(A \cup \{i\}) - f(A) \geq -\alpha$ for some small $\alpha > 0$, and for any $A \subseteq \Omega$ and any $i \in \Omega \setminus A$.

One of the most basic submodular maximization problems is USM, which is formally defined as follows.

**Definition 3 (USM).** Given a nonnegative submodular function $f : 2^S \mapsto \mathbb{R}^+$, $\max_{A \subseteq S} f(A)$ is called Unconstrained Submodular Maximization.

It is well known that USM is NP-hard [32], [33] and thus heuristic-based algorithms are often used to find approximate solutions. One state-of-art linear-time deterministic algorithm is proposed in [34] and provided in Algorithm 1 for reference in the rest of this work. It is essentially a greedy algorithm, and achieves a $1/\beta$-approximation, i.e., the algorithm obtains a solution $A$ with the guarantee that $f(A) \geq \frac{1}{\beta}f(OPT)$, where $OPT$ is the optimal solution.

#### Algorithm 1: USM

**input :** $S$ – ground set, $f$ – nonnegative submodular function

**output :** $A_n$ (or $B_n$) – selected subset

1. $A_0 \leftarrow \emptyset$, $B_0 \leftarrow S$;
2. for each $i = 1 \text{ to } n$ do
   3. $a_i \leftarrow f(A_{i-1} \cup \{u_i\}) - f(A_{i-1})$;
   4. $b_i \leftarrow f(B_{i-1} \setminus \{u_i\}) - f(B_{i-1})$;
   5. if $a_i \geq b_i$ then $A_i \leftarrow A_{i-1} \cup \{u_i\}$, $B_i \leftarrow B_{i-1}$;
   6. else $A_i \leftarrow A_{i-1}$, $B_i \leftarrow B_{i-1} \setminus \{u_i\}$;
3. return $A_n$ (or equivalently $B_n$);

**B. RSSI Data Model – Gaussian Process (GP)**

In this study, we employ GP [12], [13] (a generalization of Kriging) for radio mapping. Let the set of $n$ interested users be $S$, and the finely discretized AoI be $U$, where $|U| \gg |S| = n$, where $|\cdot|$ is the cardinality operator. Define $V = S \cup U$ and each index $i \in V$ corresponds to a location $x_i$. Since the platform obtains noisy RSSI measurements from users $S$ and wants to estimate noiseless front-end RSSIs at unmeasured locations $U$, the RSSI $Z(x_i)$ or $Z_i$ is modeled as a Gaussian random variable in GP, which is given by

$$ Z(x_i) = \begin{cases} \mu(x_i) + \delta(x_i), & \text{for } i \in U; \\ \mu(x_i) + \delta(x_i) + \epsilon_i, & \text{for } i \in S, \end{cases} \quad (dBm) \quad (9) $$

where $\mu(x_i)$ is path-loss-impaired RSSI, $\delta(x_i) \sim N(0, \sigma^2_\delta)$ is spatially correlated shadowing and $\epsilon_i \sim N(0, \sigma^2_\epsilon)$ is hardware noise of user $i$’s device.

Define a kernel (or covariance) function $K(\cdot, \cdot)$ such that $K(i, j)$ is the covariance between $\delta(x_i)$ and $\delta(x_j)$. In GP, the RSSIs at locations $V$ constitute a Gaussian random vector $Z_V = [Z(x_i)]_{i \in V}$ with a joint distribution of

$$ f_{Z_V}(z_V) = \frac{1}{(2\pi)^{n/2} |\Sigma_{VV}|} e^{-\frac{1}{2} (z_V - \mu_V)^T \Sigma_V^{-1} (z_V - \mu_V)}, \quad (10) $$

where $z_V = [z(x_i)]_{i \in V}$ is a realization of $Z_V$, $\mu_V = [\mu(x_i)]_{i \in V}$ is the mean vector and $\Sigma_V$ is the covariance matrix. For any pair of indices $i, j \in V$, their covariance $\sigma_{ij}$ is the $(i, j)$-th entry of $\Sigma_{VV}$, which is given by

$$ \sigma_{ij} = \begin{cases} K(i, j), & \text{if } i \neq j \\ K(i, j) or \sigma^2_\delta, & \text{if } i = j \in U \\ K(i, j) + \sigma^2_\delta or \sigma^2_\delta + \sigma^2_\epsilon, & \text{if } i = j \in S \end{cases} \quad (11) $$

Given a set of measurements $Z_A$ where $A \subseteq S$, $Z(x_i)$ is a conditional Gaussian random variable with a mean $\mu_{Z(x_i)|Z_A}$ (or simply $\mu_{i|A}$) and a variance of $\sigma^2_{Z(x_i)|Z_A}$ (or simply $\sigma^2_{i|A}$),

$$ \mu_{i|A} = \mu(x_i) + \Sigma_{Ai} A_i^{-1} (Z_A - \mu_A), \quad (12) $$

$$ \sigma^2_{i|A} = \sigma^2_i - \Sigma_{Ai} A_i^{-1} \Sigma_A. \quad (13) $$

Note that the posterior variance in Eq. (13) only depends on $\Sigma_V$, not the actual measured values $Z_A$.

Estimating $K(\cdot, \cdot)$ can be difficult in practice, and it is often assumed that $K(\cdot, \cdot)$ is stationary (i.e., a function of location displacement) and isotropic (i.e., a function of distance). In
other words, \( K(i, j) = K_{\theta}(|x_i - x_j|) \), where \( \theta \) is a set of parameters. That being said, our following discussions do not assume stationarity or isotropy, and thus can be applied to general kernel functions. But we do assume both mean and kernel functions have been estimated from previous measurements and available in the current period.

C. Mutual Information (MI) for Uncertainty Reduction

To measure radio mapping performance, we adopt the MI metric \(^{[13]}\), which is defined as follows,
\[
MI(A) = I(Z_A; Z_{V\backslash A}) = H(Z_{V\backslash A}) - H(Z_{V\backslash A} | Z_A),
\]
(14)
which is the amount of uncertainty reduction about RSSIs at unmeasured locations given measurements at \( A \).

Note that the platform is interested in \( Z_{V\backslash A} \), which includes RSSIs at \( S \backslash A \) (i.e., locations with confirmed user presence) and \( U \) (i.e., locations with possible user presence). As implicitly assumed in \(^{[13]}\), \( Z(x_i) \) includes noise for \( i \in S \backslash A \) in the
\[
MI(A) \text{ is not a big issue, since noise is relatively assumed in } \[13\].
\]

MI tends not to select users along the boundaries and avoids the “waste” of information.

Denote by \( MI(i|A) \) the marginal MI of an additional user \( i \in S \backslash A \) given \( A \). It is given by
\[
MI(i|A) = MI(i \cup A) - MI(A),
\]
(15)
where \( H(Z_i | Z_{A \cup i}) = \frac{1}{2} \log(2\pi e \sigma^2 Z_i) \) is the conditional entropy, and it can be easily computed from Eq. \(^{[13]}\).

It has been shown in \(^{[13]}\) that \( MI(A) \) is both submodular and \( \alpha \)-approximately monotone\(^7\) for any \( \alpha > 0 \), a discretization level exists so that \( MI(A) \) is approximately monotone.

D. Valuation Function

We consider the following valuation function \( v : 2^{S} \rightarrow \mathbb{R}_+ \) for the platform,
\[
v(A) = \kappa \cdot \log(1 + MI'(A)),
\]
(17)
where \( \kappa > 0 \) is a constant and \( MI'(A) = MI(A) + \alpha |A| \). Intuitively, \( \kappa \) is the currency that reflects the platform’s preference over per unit MI (in log scale). Commonly used in economics, \( \log(\cdot) \) further emphasizes the diminishing returns behavior. We introduce \( \alpha |A| \) to ignore the extreme case where some users are arbitrarily close to each other, which rarely occurs and/or can be avoided in practice (see Footnote \(^7\)).

Lemma 1. The valuation function \( v(\cdot) \) in Eq. \(^{[17]}\) is monotone submodular.

\[ \text{Proof.} \text{ See Appendix } \text{A-A} \text{ for proof.} \]

V. PRICING MECHANISM

In this section, we formulate pricing mechanism design as expected utility (EU) maximization and propose two schemes: (1) sequential offering and (2) batched offering.

A. EU Maximization

Given \( S, v(\cdot) \) and \( \{F_{C_i}(c_i)\} \), the platform wants to determine a set of offers \( (A, p) \), where \( A \subseteq S \) are selected users and \( p = [p_i]_{i \in A} \) is the corresponding prices. Let the decision of the \( i \)-th selected user be \( X_i \), which is given by
\[
X_i = \begin{cases} 
1, & \text{if } c_i \leq p_i \text{ (i.e., offer is accepted)} \\
0, & \text{else (i.e., offer is rejected)}
\end{cases}
\]
(18)
It is a Bernoulli random variable (from the platform’s perspective), and \( Pr(X_i = 1) = \int_{0}^{p_i} f_{C_i}(c_i) dc_i = F_{C_i}(p_i) \).

As mentioned in Section \(^{[III-A]}\), an offer may be expired, and this event can be captured by a random variable \( X'_i \), i.e.,
\[
X'_i = \begin{cases} 
1, & \text{if offer is unexpired} \\
0, & \text{if offer is expired}
\end{cases}
\]
(19)
where \( p_i = Pr(X'_i = 1) \) is the probability of an unexpired offer. We assume that the platform can estimate \( p_i \) and that \( X'_i \) is independent of \( X_i \).

Let \( Y_i \) be a random variable that represents whether a user is successfully recruited (i.e., offer is unexpired and accepted),
\[
Y_i = \begin{cases} 
1, & \text{if offer is unexpired AND accepted} \\
0, & \text{if offer is expired OR rejected}
\end{cases}
\]
(20)
where
\[
\gamma_i = Pr(Y_i = 1) = Pr(X'_i = 1, X_i = 1) = Pr(X'_i = 1) \cdot Pr(X_i = 1 | X'_i = 1) = \rho_i \cdot F_{C_i}(p_i) \in [0, 1],
\]
(21)
is the probability that the \( i \)-th selected user is recruited.

Define \( Y = \{Y_i\}_{i \in A} \) and let \( y \) be the realization of \( Y \). Then \( A_Y \subseteq A \) is the set of recruited users whose offers are unexpired and accepted. Then the EU is given by
\[
EU(A, p) = \mathbb{E}_Y[u(A_Y, p)] = \sum_{y} Pr(A_Y, p) u(A_Y, p),
\]
(22)
where
\[
Pr(A_Y, p) = \prod_{i \in A_Y} \gamma_i \cdot \prod_{i \in A_Y} (1 - \gamma_i),
\]
(23)
\[
u(A_Y, p) = v(A_Y) - \sum_{i \in A_Y} p_i,
\]
(24)
are the probability and utility of $A_y$ given $p$, respectively.

The goal of the platform is to design a pricing mechanism based on EU maximization, that is,

$$\max_{A \in S, p} EU(A, p). \quad (25)$$

In this sense, a pricing mechanism consists of a selection rule and a pricing rule, which is joint optimization in the discrete domain of $A$ and the continuous domain of $p$.

**B. Sequential Offering**

We first consider a special case of EU maximization, where $|A| = 1$. That is, the platform only selects one best user with its best offer in each round, and its decision before making the next offer. We call it sequential offering. Formally, the task in each round is

$$\max_{i \in S \setminus A, p, y} \text{EU}(A \cup \{i\}, [p, p_i] | Y = y), \quad (26)$$

where $y$ represents the outcomes of offers that have been sent so far and is known to the platform.

**Algorithm 2: Sequential Offering**

```
input: S - set of users, v(·) - valuation function, 
{FC_i(·)} - cost distributions, {ρ_i} - probabilities of unexpired offers, τ - threshold
output: A - selected users, p - prices, y - outcomes
1 A ← Ø, p ← NULL, y ← NULL;
2 while A ≠ S do
3     foreach each user i in S \ A do
4         p_i^* ← arg max_{p_i \in [c_i, \bar{c}_i]} [v(i|Ay) - ρ_i \cdot FC_i(p_i)];
5         EU_i ← [v(i|Ay) - p_i^* \cdot ρ_i \cdot FC_i(p_i^*)];
6         i^* ← arg max_{i \in S \setminus A} EU_i;
7     while EU_i > τ do
8         Send the offer (i^*, p_i^*) and observe y_i^*;
9         A ← A ∪ {i^*}, p ← [p^{i^*}, p_i^*], y ← [y, y_i^*];
10        if y_i = 1 then break;
11    else i^* ← arg max_{i \in S \setminus A} EU_i;
12 return A, p, y;
```

The algorithm for sequential offering is described in Algorithm 2. The idea is to determine the best price for each remaining user that maximizes the EU, and then selects the best user with the maximum EU obtained at its best price.

1) **Price Determination (Lines 3-5):** Depending on whether user $i \in S \setminus A$ is successfully recruited, the utility is

$$u(A_y \cup \{i\}, [p, p_i]) = u(A_y, p) + \begin{cases} v(i|A_y) - p_i, & \text{if } Y_i = 1 \\ 0, & \text{otherwise}, \end{cases} \quad (27)$$

where $v(i|A_y) = v(\{i\} \cup A_y) - v(A_y)$ is the marginal value of $i$ given $A_y$. The task is to find

$$p_i^* = \arg\max_{p_i \in [c_i, \bar{c}_i]} E_Y[v(A_y \cup \{i\}, [p, p_i])]
= \arg\max_{p_i \in [c_i, \bar{c}_i]} [v(i|A_y) - p_i] \cdot Pr(Y_i = 1)
= \arg\max_{p_i \in [c_i, \bar{c}_i]} [v(i|A_y) - p_i] \cdot FC_i(p_i). \quad (28)$$

Note that $ρ_1$ in Pr($Y_1 = 1$) in Eq. (21) is omitted since it is a constant and does not affect $p_i^*$. If $f_{C_i}(c_i) = F'_{C_i}(c_i)$ is differentiable and non-increasing, the objective function in Eq. (25) will be concave in $p_i$, and $p_i^*$ can be obtained with efficient algorithms (e.g., gradient descent). If $F_{C_i}(c_i)$ is twice continuously differentiable, techniques such as interval analysis may be used to find $p^*$ [35]. More sophisticated algorithms for non-convex (non-concave) optimization are available in [36].

2) **User Selection (Line 6):** The best user $i^*$ that maximizes the EU is found,

$$i^* = \arg\max_{i \in S \setminus A} [v(i|A_y) - p_i^*] \cdot ρ_i \cdot FC_i(p_i^*). \quad (29)$$

Note that the above selection also takes $ρ_1$ into account. If the user is recruited (Lines 10), the algorithm will go to Line 3 to recompute best prices for remaining users; otherwise, it sends out the next best offer immediately until one is accepted (Lines 7-11). To enable fast convergence, the platform can set a minimum threshold $τ > 0$ (e.g., 0.01) for the marginal EU (Line 7). The platform stops making offers when there are (1) no remaining users or (2) none of the remaining users leads to a non-trivial marginal EU.

3) **Complexity Analysis:** If we assume $O(1)$ for computing the best price for a single user, the overall computational complexity of Algorithm 2 is $O(n^2)$, since it takes $O(n)$ to compute best prices for all remaining users and may select up to $n$ users in the worst case. The inner while-loop does not require re-computation of best prices and is dominated by the for-loop. Note that $O(n^2)$ is very conservative, since the algorithm may stop much earlier based on the configuration.

**C. Batched Offering**

As we can see, sequential offering is intuitive and straightforward, but its main drawback is the (possibly) large delay accumulated over multiple rounds of offering. Hence, a natural generalization is to make multiple offers (i.e., a batch) in each round and continue offering for multiple rounds. We call it batched (i.e., single-batch or multi-batch) offering.

In batched offering, the platform is faced with the general case of EU maximization in Eq. (25) in each round. Unfortunately, joint optimization can be difficult in practice, mainly because $EU(A, p)$ is a multi-variate function in the continuous domain of $p$ given $A$, and there may not exist structural properties like concavity in general to enable efficient computation of the global optimum. Exhaustive search is prohibitive as the space of $p$ is huge. Fortunately, $EU(A, p)$ has a useful structural property (i.e., submodularity) in the discrete domain of $A$ as a set function, which inspires our following pricing mechanism design.

**Lemma 2.** For a given $p$, $EU(A, p)$ is submodular in $A$.

**Proof.** See Appendix A-B for proof.

The basic idea of our pricing mechanism is to first fix the pricing rule and then focus on user selection by exploiting the submodularity property. As mentioned in Section IV-A if a set function $f$ is nonnegative submodular and the problem is $\max_{A \subseteq S} f(A)$, there exist heuristic-based algorithms (e.g.,
Algorithm 1 that provide solutions with performance guarantee at low complexity. Next, we will present our pricing mechanisms for single-batch and multi-batch offering.

1) Single-Batch Offering: As mentioned in Section III we consider the following pricing rule in this work: the platform chooses a desired probability of recruitment $\gamma \in (0, 1]$ such that $\gamma_i = \min(\gamma, \rho_i)$ for any $i \in S$ and determines corresponding prices, i.e.,

$$p_\gamma(A) = \sum_{i \in A} p_\gamma(\{i\}) \leq \sum_{i \in A} E_{C_i}^{-1}(\min(\gamma/\rho_i, 1)) .$$

(30)

Given $p_\gamma(\cdot)$, user selection then becomes

$$\max_{A \subseteq S} EU_\gamma(A) = \max_{A \subseteq S} \sum_{y} Pr_\gamma(A_y)u_\gamma(A_y),$$

(31)

where $Pr_\gamma(A_y) = \prod_{i \in \gamma} (1 - \gamma_i)$ and $u_\gamma(A_y) = v(A_y) - p_\gamma(A_y)$. By Lemma 2, $EU_\gamma(A)$ is submodular.

However, the USM formulation also requires the objective function to be nonnegative, but $EU_\gamma(A)$ can be negative. To bypass this issue, one straightforward way is to define

$$EU'_\gamma(A) = EU_\gamma(A) + p_0$$

(32)

where

$$p_0 = \sum_{i \in S} \gamma_i p_\gamma(\{i\})$$

(33)

is a constant that represents the maximum expected price, and adding a constant preserves submodularity. It is easy to see that $EU'_\gamma(A)$ is both submodular and nonnegative, and $\max_{A \subseteq S} EU'_\gamma(A)$ is equivalent to $\max_{A \subseteq S} EU_\gamma(A)$.

Another issue is that it may be difficult to analytically evaluate $EU_\gamma(A)$ (or $EU'_\gamma(A)$), since it involves an exponentially growing number of terms due to the summation in Eq. (31). In practice, the Monte-Carlo (MC) method [37] can be used to obtain estimates of $EU_\gamma(A)$ as well as $EU'_\gamma(A)$ (by adding the constant $p_0$ to $EU_\gamma(A)$). We show that USM$(S, EU'_\gamma)$ has the following performance.

Theorem 1. If $|EU'_\gamma(A) - EU'_\gamma(A)| \leq \epsilon$ for any $A \subseteq S$, USM$(S, EU'_\gamma)$ (or equivalently USM$(S, EU_\gamma)$) returns a solution $A$ with the following performance,

$$EU'_\gamma(A) \geq \frac{1}{3} EU'_\gamma(OPT) - \frac{1}{3}(2n + 2)\epsilon,$$

$$EU_\gamma(A) \geq \frac{1}{3} EU_\gamma(OPT) - \frac{1}{3}(2n + 2)\epsilon,$$

(34)  

(35)

where $OPT$ is the optimal solution for $\max_{A \subseteq S} EU'_\gamma(A)$ as well as $\max_{A \subseteq S} EU_\gamma(A)$, $p_0 = \sum_{i \in S} \gamma_i p_\gamma(\{i\})$ and $n = |S|$.

Proof. See Appendix A-C for proof.

Algorithm 3 describes the algorithm for single-batch offering. Since each $\gamma$ leads to a different solution $A_\gamma$, it is better for the platform to search through a list of $l$ candidate $\gamma$ values (e.g. \{0.1, 0.2, ..., 1.0\}) to find the best one that maximizes $EU_\gamma(A_\gamma)$. As USM takes $O(l)$ time, the overall complexity of Algorithm 3 is $O(ln)$.

Algorithm 3: Single_Batch_Offering

input : $S$ – set of users, $f - EU'_\gamma(\cdot)$ (or equivalently $EU_\gamma(\cdot)$), $\Gamma = [\gamma_1, ..., \gamma_l]$ – candidate $\gamma$ values where $\gamma_1 < \gamma_2 < \cdots < \gamma_l$, $p_\gamma(\cdot)$ – pricing rule

output: $A$ – selected users, $p$ – prices, $\gamma^*$ – best probability of recruitment

1. $A \leftarrow \emptyset$, $p \leftarrow NULL$, $\gamma^* \leftarrow 0$;
2. foreach $\gamma$ in $[\gamma_1, ..., \gamma_l]$ do
3. $A_\gamma \leftarrow$ USM$(S, f)$;
4. if $A_\gamma = \emptyset$ then break;
5. if $EU_\gamma(A_\gamma) > EU_\gamma(A)$ then
6. $A \leftarrow A_\gamma$, $\gamma^* \leftarrow \gamma$;
7. foreach each user $i$ in $A$ do
8. $p \leftarrow [p, p_\gamma(\{i\})]$;
9. return $A$, $p$, $\gamma^*$;

2) Multi-Batch Offering: In the case of expired or rejected offers in the previous batch, the platform may send out more batches until the next batch is no longer profitable.

Denote by $EU_\gamma(B|A_y)$ the marginal EU of additional offers $B$ conditioned on the set of recruited users $A_y$,

$$EU_\gamma(B|A_y) = E_{\gamma'}[u_{\gamma'}(B_{\gamma'}|A_y)],$$

(36)

where $u_{\gamma'}(B_{\gamma'}|A_y) = u_{\gamma'}(B_{\gamma'} \cup A_y) - u_{\gamma'}(A_y) = v(B_{\gamma'}|A_y) - p_{\gamma'}(B_{\gamma'})$ is the marginal utility of $B_{\gamma'}$ given $A_y$. We can see from Lemma 2 that $EU_\gamma(B|A_y)$ is again submodular in $B$.

Algorithm 4: Multi_Batch_Offering

input : $S$ – set of users, $f - EU'_\gamma(\cdot)$ (or equivalently $EU_\gamma(\cdot)$), $\Gamma = [\gamma_1, ..., \gamma_l]$ – candidate $\gamma$ values where $\gamma_1 < \gamma_2 < \cdots < \gamma_l$, $p_\gamma(\cdot)$ – pricing rule, $\tau$ – threshold

output: $A$ – selected users, $p$ – prices, $\gamma -$ outcomes

1. $A \leftarrow \emptyset$, $p \leftarrow NULL$, $\gamma \leftarrow NULL$;
2. while $A \neq S$ do
3. $(B, p_B, \gamma^*) \leftarrow$ Single_Batch_Offering($S \setminus A$, $f(\cdot|A_y), \Gamma, p_\gamma(\cdot)$);
4. if $EU_\gamma(B|A_y) > \tau$ then
5. send out offers $(B, p_B, \gamma)$ and observe $y_B$;
6. $A \leftarrow A \cup B$, $p \leftarrow [p, p_B]$, $\gamma \leftarrow [\gamma, y_B]$;
7. else break;
8. return $A$, $p$, $\gamma$;

The algorithm for multi-batch offering is provided in Algorithm 4. Starting from the second batch, the marginal EU function is passed as an input to Single_Batch_Offering (Line 3). If the (estimated) marginal EU of $B$ is higher than a preset threshold $\tau > 0$ (e.g. 0.01), then it is profitable on average to send out the next batch of offers (Lines 5). The platform will then wait for the results and update $A$, $p$, $\gamma$ accordingly (Line 6). The offering process stops when (1) there are no more users to consider (Line 2), or (2) the next batch is no longer profitable on average (Line 4).
VI. Evaluation

In this section, we conduct simulations to evaluate proposed pricing mechanisms based on EU maximization, and compare them against baseline mechanisms based on (best-case) utility maximization. We also study the impact of offer expiration on our pricing mechanisms.

A. Simulation Setup

Fig. 2a is a sample topology of 60 users, whose locations are randomly generated from the spatial Poisson process. The AoI is a 6km-by-6km region, discretized into a total of 169 points with a resolution of 450 meters. We assume an exponential kernel function $K(d) = 15.5 \cdot \exp(-\frac{d}{0.7})$, as shown in Fig. 2b, which is adapted from the semivariogram fitted from real measurements [9]. Given the above settings, negative marginal MI values are not observed and $\alpha$ is set to 0.

The domain of $C_i$ is $[c_i, c_i + \Delta c]$, where $\Delta c > 0$ and $c_i$ is randomly generated from $U[0,1,0,2]$, where $U[\cdot,\cdot]$ denotes the uniform distribution. We consider two types of distributions: uniform (UN) and truncated normal (TN) distributions. TN is the normal distribution $N(c_i, (\Delta c/3)^2)$ truncated to $[c_i, c_i + \Delta c]$. Compared to UN, TN represents the situation where the majority of users have sensing costs closer to the energy costs $c_i$, despite the presence of opportunity costs. Besides, the same set of noise variances independently drawn from $U[0.5,1]$ is used throughout our simulation.

![Sample topology](image1)

![Kernel function](image2)

Fig. 2: (a) Sample topology of 60 users (in red) in a 6km-by-6km area that is discretized into a mesh grid of 169 points (in blue). (b) The kernel function $K(d) = 15.5 \cdot \exp(-\frac{d}{0.7})$.

1) Baseline Mechanisms: As mentioned in Section III, an alternative way to design a pricing mechanism is to maximize the best-case utility as in Eq. (1), assuming no expired or rejected offers. In this simulation, we consider single-batch and multi-batch offering based on utility maximization as baseline mechanisms. That is, instead of passing $EU_k^*()$ into USM in each batch (Line 3 of Algorithm 2), we pass $\hat{u}_k()$ (a nonnegative submodular function) into USM, where $\hat{u}_k(A) = u_k(A) + p_\gamma(S)$ and $u_k(A) = v_k(A) - p_\gamma(A)$. Compared to $EU_k^*()$, the objective function $\hat{u}_k()$ does not require the MC method and is easy to evaluate.

For convenience, we refer to EU-maximization-based mechanisms by feeding $EU_k^*()$ to USM as USM-EU, and baseline mechanisms by feeding $\hat{u}_k()$ to USM as USM-u.

B. USM-EU vs. USM-u in Single-Batch Offering

In this experiment, we compare the performance of USM-EU and USM-u in single-batch offering. $EU_k^*()$ is obtained by averaging over 50 iterations of MC simulations. We randomly select 30 or 60 users, and set $\gamma$ in $v()$ (Eq. (17)) to 4 or 8, and $\Delta c$ to 0.1 or 0.5. We assume no expired offers by setting $\rho_i = 1$ for each user $i$, and will study its impact later in Section VI-C3. A total of 30 iterations are conducted, and a different seed is used for generating users and cost distributions in each iteration. In the $i$-th iteration, however, the same set of users and cost distributions are used across different $\gamma$ and mechanisms for fair comparison. Results are provided in Fig. 3.

![Average EU](image3)

Fig. 3: Average EU achieved by USM-u and USM-EU for UN and TN cost distributions under different settings.

In Fig. 3a, we first observe that $\gamma^*$ achieving the maximum averaged EU is less than 1 for both USM-EU and USM-u under both UN and TN distributions. Intuitively, with a smaller $\gamma$, the platform can save more money per user and send out more offers. Although each offer is less likely to be accepted, the platform achieves a greater EU on average. Second, at $\gamma = \gamma^*$, USM-EU achieves a higher EU than USM-u, especially for UN. Besides, the fact that more money is saved per user under TN than UN with the same $\gamma$ explains the observation that both USM-EU and USM-u achieve a larger EU under TN than UN.

Similar behaviors are observed in Fig. 3b and 3d. But in Fig. 3c, when $\Delta c$ is changed from 0.5 to 0.1, the uncertainty in opportunity costs is reduced and energy costs become more dominant. In this case, the platform will not save much per user with a small $\gamma$ and should choose a larger $\gamma$. Besides, we do not observe the advantage of USM-EU.
Furthermore, as shown in Fig. 3a, USM-EU is slightly better than USM-u with γ between 0.3 and 0.6 under UN, but their performance is very close for γ ≤ 0.2 under UN and for all γ values under TN. This is mainly because in those cases, both mechanisms will send out offers to more or all users (30 max. in this setting) and thus achieve very close EU. When there are more users (Fig. 3b), USM-EU is better than USM-u under both UN and TN for γ < 0.6. With smaller Δc (Fig. 3c), USM-EU is less advantageous. With smaller κ (Fig. 3d), each user is less valuable and USM-u selects fewer users, since it assumes no expired or rejected offers. In contrast, USM-EU considers the average-case utility and is more aggressive in user selection, which explains its better performance than USM-u with the same small γ.

C. Batched Offering vs. Sequential Offering

In this simulation, we compare the following mechanisms:
- SB-u/EU: single-batch offering with USM-u or USM-EU;
- MB-u/EU: multi-batch offering with USM-u or USM-EU;
- SE: sequential offering.

We randomly select n out of 60 users, and generate a set of noise variances and cost distributions, which are used in all iterations for each n. In the i-th iteration, a different set of sensing costs is independently generated from the cost distributions and used across different mechanisms for fair comparison. We set Δc = 0.5, and varied n, κ or \{ρ_i\} to study their impacts on the average utility achieved by the platform. All results are averaged over 50 iterations. Due to limited space, only results for UN distributions are reported, but similar observations exist for TN distributions.

1) Impact of n (number of users): We first set κ = 4 and ρ_i = 1 for each user i (i.e., no expired offers). n is varied from 10 to 60, and results are provided in Fig. 4a. First, we observe that all mechanisms achieve a higher utility on average as n increases. Second, if only one batch/round is allowed, USM-EU achieves the highest utility, since it accounts for possible expiration and rejection and thus makes more offers in the first batch. The improvement of USM-EU (i.e., SB-EU) over USM-u (i.e., SB-u) varies from 8.5% with n = 10 to 40.5% with n = 60 (Fig. 4a), which means the advantage of SB-EU (over SB-u) is more obvious with more users. SE performs the worst, since it only sends out one offer in the first round.

When more batches are allowed (Fig. 4b), all mechanisms perform better, since it is always beneficial to send out more batches to make up for expired or rejected offers in the previous batch. Since USM-EU is very generous in making offers in the first batch, following batches become less profitable. If the maximum number of batches is unlimited, USM-u (i.e., MB-u) eventually achieves very close performance with USM-EU (i.e., MB-EU), but the price is a much larger cumulative delay. For instance, when n = 60, MB-EU and MB-u make 2.5 and 7.7 batches of offers on average, but the number is 24.9 for SE, which is the worst.

2) Impact of κ (currency in Eq. (17)): We then fix n = 30 and vary κ from 1 to 6. We set ρ_i = 1 for each user i. As shown in Fig. 4c, the average utility obtained by each mechanism increases as κ increases, because the platform values per unit MI (log-scaled) more and is able to recruit more users. Besides, SB-EU is still better than SB-u for different κ, but its advantage is less obvious when κ gets larger. For instance, the improvement is 127.1% with κ = 1, but reduces to 13.4% with κ = 6. Moreover, multi-batch offering is better than single-batch offering for both USM-EU and USM-u, but the improvement is more significant for USM-u.

3) Impact of ρ (probability of unexpired offers): We set n = 30 and κ = 4. For simplicity, we assume ρ_i = ρ for each user i and vary ρ from 0.2 to 1. Results are provided in Fig. 4d. First, we observe that all mechanisms are adversely affected when ρ decreases. Even though the platform knows ρ and adjusts the price as in Eq. (30) to achieve γ, i.e., increasing the acceptance probability to offset the high expiration probability, it implies higher prices for users and consequently reduced utility. As mentioned earlier in Section VII, the best price for each user in SE does not depend on ρ, but the resulting marginal EU does. Nevertheless, SE observes the outcome of the previous offer and continues offering, which explains it (as well as multi-batch offering) is more robust against offer expiration than single-batch offering.

VII. CONCLUSION AND FUTURE WORK

In this work, we designed a crowd-sensing system for spatial-statistics-based radio mapping and developed pricing mechanisms, i.e., sequential and batched offering, based on EU
maximization. We conducted extensive simulations to evaluate proposed mechanisms. Our results show that if only one batch is allowed, the proposed mechanism based on EU maximization is significantly better than the utility-maximization-based baseline mechanism. If multiple batches are permitted (and the number of batches is unlimited), the proposed mechanism achieves close performance with the baseline mechanism, but requires much fewer batches and thus a much smaller delay. Sequential offering works better than the single-batch baseline mechanism, but has a much larger cumulative delay. Offer expiration adversely affected all mechanisms, but sequential and batched offering are relatively more robust.

There are several interesting and important future directions that could be explored. First, it would be very interesting to apply spatio-temporal statistics to model temporal correlations and capture temporal variations in radio environments. Second, it would be of practical importance to investigate the problem in an online setting, where users arrive in random order and one-time price offer has to be made upon each user’s arrival. Such an online pricing mechanism would better handle high user mobility and minimize actual displacements between reported and sensing locations. Last but not the least, we would like to consider security and user privacy of our system.

REFERENCES

[1] X. Ying, S. Roy, and R. Poovendran, “Pricing mechanism for quality-based radio mapping via crowdsourcing,” in 2016 IEEE Global Communications Conference (GLOBECOM), Dec 2016, pp. 1–6.

[2] “In the matter of unlicensed operation in the TV broadcast bands: Second report and order and memorandum opinion and order,” no. FCC Std. 08-260, Nov 2008.

[3] K. Harrison et al., “How much white-space capacity is there?” in New Frontiers in Dynamic Spectrum, 2010 IEEE Symposium on. IEEE, 2010, pp. 1–10.

[4] D. Makris et al., “Quantifying tv white space capacity: A geolocation-based approach,” IEEE Communications Magazine, 2012.

[5] “In the matter of unlicensed operation in the TV broadcast bands: Second memorandum opinion and order,” no. FCC Std. 10-174, Sep 2010.

[6] F. Hessar and S. Roy, “Capacity considerations for secondary networks in tv white space,” IEEE Transactions on Mobile Computing, vol. 14, no. 9, pp. 1780–1793, 2015.

[7] C. Phillips et al., “Bounding the error of path loss models,” in New Frontiers in Dynamic Spectrum Access Networks (DisSPAN), 2011 IEEE Symposium on. IEEE, 2011, pp. 71–82.

[8] A. Achtzehn, J. Riihijarvi, and P. Mahonen, “Improving accuracy for TVWS geolocation databases: Results from measurement-driven estimation approaches,” in Dynamic Spectrum Access Networks (DisSPAN), 2014 IEEE International Symposium on. IEEE, 2014, pp. 392–403.

[9] X. Yang, C. W. Kim, and S. Roy, “Revisiting TV coverage estimation with measurement-based statistical interpolation,” in IEEE COMSNETS. IEEE, 2015, pp. 1–8.

[10] N. Cressie, Statistics for spatial data. John Wiley & Sons, 2015.

[11] A. Achtzehn et al., “Improving coverage prediction for primary multi-transmitter networks operating in the TV whitespaces,” in IEEE SECON 2012.

[12] A. Deshpande et al., “Model-driven data acquisition in sensor networks,” in Proceedings of the Thirteenth international conference on Very large data bases-Volume 30. VLDB Endowment, 2004, pp. 588–599.

[13] A. Krause et al., “Near-optimal sensor placements in gaussian processes: Theory, efficient algorithms and empirical studies,” Journal of Machine Learning Research, vol. 9, no. Feb, pp. 235–284, 2008.

[14] D. Yang et al., “Crowdsourcing to smartphones: incentive mechanism design for mobile phone sensing,” in ACM MobiCom, 2012, pp. 173–184.

[15] I. Koutsopoulos, “Optimal incentive-driven design of participatory sensing systems,” in IEEE INFOCOM. IEEE, 2013, pp. 1402–1410.

[16] D. Peng, F. Wu, and G. Chen, “Pay as how well you do: A quality based incentive mechanism for crowdsensing,” in Proceedings of the 16th ACM International Symposium on Mobile Ad Hoc Networking and Computing. ACM, 2015, pp. 177–186.

[17] A. Singla et al., “Truthful incentives in crowdsourcing tasks using regret minimization mechanisms,” in Proceedings of the 22nd international conference on World Wide Web. ACM, 2013.

[18] S. He, D.-H. Shin, J. Zhang, and J. Chen, “Toward optimal allocation of location dependent tasks in crowdsensing,” in INFOCOM, 2014 Proceedings IEEE. IEEE, 2014, pp. 745–753.

[19] K. Han et al., “Posted pricing for robust crowdsensing,” in Proceedings of the 17th ACM International Symposium on Mobile Ad Hoc Networking and Computing. ACM, 2016.

[20] C. Phillips et al., “Practical radio environment mapping with geostatistics,” in Dynamic Spectrum Access Networks (DisSPAN), 2012 IEEE International Symposium on. IEEE, 2012, pp. 422–433.

[21] H. B. Yilmaz, T. Tugcu, F. Alagöz, and S. Bayhan, “Radio environment map as enabler for practical cognitive radio networks.” IEEE Communications Magazine, vol. 51, no. 12, pp. 162–169, 2013.

[22] Z. Feng et al., “Trac: Truthful auction for location-aware collaborative sensing in mobile crowdsourcing,” in INFOCOM, 2014 Proceedings IEEE. IEEE, 2014, pp. 1231–1239.

[23] X. Ying, S. Roy, and R. Poovendran, “Incentivizing crowdsourcing for radio environment mapping with statistical interpolation,” in Dynamic Spectrum Access Networks (DisSPAN), 2015 IEEE International Symposium on. IEEE, 2015, pp. 365–374.

[24] T. Luo, S. K. Das, H. P. Tan, and L. Xia, “Incentive mechanism design for crowdsourcing: An all-pay auction approach,” ACM Transactions on Intelligent Systems and Technology (TIST), vol. 7, no. 3, p. 35, 2016.

[25] T. Luo, S. S. Kanhere, H.-P. Tan, F. Wu, and H. Wu, “Crowdsourcing with talkclock contests: A new perspective,” in Computer Communications (INFOCOM), 2015 IEEE Conference on. IEEE, 2015, pp. 2515–2523.

[26] Y. Singer and M. Mittal, “Pricing mechanisms for crowdsourcing markets,” in Proceedings of the 22nd international conference on World Wide Web. ACM, 2013, pp. 1157–1166.

[27] D. Zhao et al., “How to crowdsourse tasks truthfully without sacrificing utility: Online incentive mechanisms with budget constraint,” in IEEE INFOCOM 2014, 2014.

[28] K. Han, C. Zhang, and J. Luo, “Taming the uncertainty: budget limited robust crowdsensing through online learning,” Biological Cybernetics, vol. 24, no. 3, pp. 1462–1475, 2016.

[29] K. Han, C. Zhang, J. Luo, M. Hu, and B. Veeravalli, “Truthful scheduling mechanisms for powering mobile crowdsensing,” IEEE Transactions on Computers, vol. 65, no. 1, pp. 294–307, 2016.

[30] J. Scott, “Rational choice theory,” Understanding contemporary society: Theories of the present, vol. 129, 2000.

[31] G. J. Pottie and W. J. Kaiser, “Wireless integrated network sensors,” Communications of the ACM, vol. 43, no. 5, pp. 51–58, 2000.

[32] S. Fujishige, Submodular functions and optimization. Elsevier, 2005, vol. 58.

[33] U. Feige et al., “Maximizing non-monotone submodular functions,” SIAM Journal on Computing, 2011.

[34] N. Buchbinder, M. Feldman, J. Seffi, and R. Schwartz, “A tight linear time (1/2)-approximation for unconstrained submodular maximization,” SIAM Journal on Computing, vol. 44, no. 5, pp. 1384–1402, 2015.

[35] E. R. Hansen, “Global optimization using interval analysis: the one-dimensional case,” Journal of Optimization Theory and Applications, vol. 29, no. 3, pp. 331–344, 1979.

[36] P. Pardalos et al., Nonconvex Optimization and Its Applications. Springer, 1994, vol. 1.

[37] R. Y. Rubinstein and D. P. Kroese, Simulation and the Monte Carlo method. John Wiley & Sons, 2016.

APPENDIX A

A. Proof of Lemma 7

Consider two sets $A, B$ such that $A \subseteq B \subseteq S$ and any $i \in S \setminus B$. Let $f(A) = 1 + MI(A)$. First, we know that $f(A)$ is submodular from the submodularity of $MI(A)$, i.e.,

$$f(A \cup \{i\}) - f(A) = MI(A \cup \{i\}) - MI(A) + \alpha \geq MI(B \cup \{i\}) - MI(B) + \alpha = f(B \cup \{i\}) - f(B).$$

(37)
We also know that \( f(A) \) is monotone, since \( MI(A) \) is \( \alpha \)-approximately monotone and thus
\[
f(A \cup \{i\}) - f(A) = MI(A \cup \{i\}) - MI(A) + \alpha \geq 0. \tag{38}
\]

Let \( a = f(A), b = f(A \cup \{i\}), c = f(B), \) and \( d = f(B \cup \{i\}) \). From the submodularity and monotonicity of \( f(\cdot) \), we have \( b - a \geq d - c \geq 0 \) and \( b \geq d \). Let \( a' = b - (d - c) \geq a \). Since \( \log(\cdot) \) is non-decreasing concave, we have \( \log(b) - \log(a) \geq \log(d) - \log(c) \). Hence, \( v(A \cup \{i\}) - v(A) \geq v(B \cup \{i\}) - v(B) \geq 0 \), and \( v(\cdot) \) is submodular monotone.

**B. Proof of Lemma 2**

We notice that \( u(A_y, p) \) is submodular in \( A_y \) given \( p \), since \( v(\cdot) \) is submodular (Lemma 1) and
\[
u(A_y \cup \{i\}, p) = v(A_y) - p_i \geq v(i) - p_i \geq u(B \cup \{i\}, p) - u(B, p)
\tag{39}
\]
for \( A_y \subseteq B \subseteq S \) and any \( i \in S \setminus B \). Since the class of submodular functions are closed under taking expectations, it follows that \( EU(A, p) \) is submodular in \( A \) given \( p \).

**C. Proof of Theorem 1**

Before proving Theorem 1 we first prove the following lemma.

**Lemma 3.** Given a nonnegative submodular function \( f : 2^S \rightarrow \mathbb{R}_+ \), and its estimate \( \hat{f} \) with \( |\hat{f}(A) - f(A)| \leq \epsilon \) for any \( A \subseteq S \) and some small \( \epsilon > 0 \), USM(S, \( \hat{f} \)) returns a solution \( A \) with the following performance guarantee,
\[
f(A) \geq \frac{1}{3} f(OPT) - \frac{1}{3}(2n + 2)\epsilon
\tag{40}
\]
where \( OPT = \arg \max_{A \subseteq S} f(A) \) and \( n = |S| \).

**Proof.** Our proof is inspired by the proof in [34]. Let us start with Lemma 4.

**Lemma 4.** For every \( 1 \leq i \leq n \), \( a_i + b_i \geq -4\epsilon \), where \( a_i = \hat{f}(A_i-1 \cup \{u_i\}) - \hat{f}(A_i-1) \) and \( b_i = \hat{f}(B_i-1 \setminus \{u_i\}) - \hat{f}(B_i-1) \).

**Proof.** Since \( |\hat{f}(A) - f(A)| \leq \epsilon, \forall A \subseteq S \), we have
\[
f(A) - \epsilon \leq \hat{f}(A) \leq f(A) + \epsilon, \forall A \subseteq S.
\tag{41}
\]
Notice that \((A_i-1 \cup \{u_i\}) \cap (B_i-1 \setminus \{u_i\}) = B_i-1, (A_i-1 \cup \{u_i\}) \cap (B_i-1 \setminus u_i) = A_i-1 \). Based on both observations and submodularity of \( f \), we get
\[
a_i + b_i = [\hat{f}(A_i-1 \cup \{u_i\}) - \hat{f}(A_i-1)] + [\hat{f}(B_i-1 \setminus \{u_i\}) - \hat{f}(B_i-1)] \geq [f(A_i-1 \cup \{u_i\}) + f(B_i-1 \setminus \{u_i\})] - 4\epsilon \geq -4\epsilon.
\tag{42}
\]

Define \( OPT_i = \{OPT \cup A_i\} \cap (B_i) \). Thus, \( OPT_0 = OPT \) and the algorithm outputs \( OPT_n = A_n = B_n \). Examine the sequence \( f(OPT_0), ..., f(OPT_n) \), which starts with \( f(OPT) \) and ends with the \( f \) value of the output of the algorithm. The idea is to bound the total loss of value along this sequence.

**Lemma 5.** For every \( 1 \leq i \leq n \), we have
\[
f(OPT_{i-1}) - f(OPT_i) \leq [\hat{f}(A_i) - \hat{f}(A_i-1)] + [\hat{f}(B_i) - \hat{f}(B_i-1)] + 2\epsilon.
\tag{44}
\]

**Proof.** W.L.O.G., we assume that \( a_i \geq b_i \), i.e., \( A_i \leftarrow A_i-1 \cup \{u_i\}, B_i \leftarrow B_i-1 \) (the other case is analogous). Notice that in this case \( OPT_i = (OPT \cup A_i) \cap B_i = OPT_{i-1} \cup \{u_i\} \), \( B_i = B_{i-1} \) and \( \hat{f}(B_i) = \hat{f}(B_{i-1}) \). Hence, the inequality we need to prove is that
\[
f(OPT_{i-1}) - f(OPT_i \cup \{u_i\}) \leq [\hat{f}(A_i) - \hat{f}(A_i-1)] + 2\epsilon
\tag{45}
\]

We now consider two cases. If \( u_i \in OPT \), then the left-hand of the inequality is 0, and all we need to show is that \( a_i \geq -2\epsilon \). This is true since \( a_i + b_i \geq -4\epsilon \) by Lemma 4 and we assumed \( a_i \geq b_i \).

If \( u_i \notin OPT \), then also \( u_i \notin OPT_{i-1} \), and thus
\[
f(OPT_{i-1}) - f(OPT_{i-1} \cup \{u_i\}) \leq \hat{f}(B_i-1 \setminus \{u_i\}) - \hat{f}(B_i-1) + 2\epsilon = b_i + 2\epsilon \leq a_i + 2\epsilon.
\tag{46}
\]

The first inequality follows by submodularity: \( OPT_{i-1} = ((OPT \cup A_i) \cap B_i) \subseteq (B_i \setminus \{u_i\}) \) (recall that \( u_i \in B_i \) and \( u_i \notin OPT_{i-1} \)). The second inequality follows from our assumption that \( a_i \geq b_i \).

Summing up Lemma 5 for every \( 1 \leq i \leq n \), we have
\[
\sum_{i=1}^{n}[f(OPT_{i-1}) - f(OPT_i)] \leq \sum_{i=1}^{n}[\hat{f}(A_i) - \hat{f}(A_{i-1})] + \sum_{i=1}^{n}[\hat{f}(B_i) - \hat{f}(B_{i-1})] + 2n\epsilon.
\tag{47}
\]

The above sum is telescopic and we have
\[
f(OPT_0) - f(OPT_n) \leq [\hat{f}(A_n) - \hat{f}(A_0)] + [\hat{f}(B_n) - \hat{f}(B_0)] + 2n\epsilon
\leq f(A_n) + f(B_n) + 2n\epsilon \leq f(A_n) + f(B_n) + (2n + 2)\epsilon
\tag{48}
\]
By our definition, \( OPT_0 = OPT \) and \( OPT_n = A_n = B_n \). Then we obtain that \( f(OPT) \leq 3f(A_n) + (2n + 2)\epsilon \) and \( f(A_n) = f(B_n) \).