Technical Report

TR-2014-003

Spatial Task Assignment for Crowd Sensing with Cloaked Locations

by

Layla Pournajaf, Li Xiong, Vaidy Sunderam, Slawomir Goryczka

Mathematics and Computer Science

EMORY UNIVERSITY

Spatial Task Assignment for Crowd Sensing with Cloaked Locations

Layla Pournajaf, Li Xiong, Vaidy Sunderam, Slawomir Goryczka
Department of Mathematics and Computer Science
Emory University, Atlanta, Georgia 30322
lpourna|lxiong|vss|sgorycz@emory.edu

***Abstract*—Distributed mobile crowd sensing is becoming a valuable paradigm, enabling a variety of novel applications built on mobile networks and smart devices. However, this trend brings several challenges, including the need for crowdsourcing platforms to manage interactions between applications and the crowd (participants or workers). One of the key functions of such platforms is spatial task assignment which assigns sensing tasks to participants based on their locations. Task assignment becomes critical when participants are hesitant to share their locations due to privacy concerns. In this paper, we examine the problem of spatial task assignment in crowd sensing when participants utilize spatial cloaking to obfuscate their locations. We investigate methods for assigning sensing tasks to participants, efficiently managing location uncertainty and resource constraints. We propose a novel two-stage optimization approach which consists of global optimization using cloaked locations followed by a local optimization using participants' precise locations without breaching privacy. Experimental results using both synthetic and real data show that our methods achieve high sensing coverage with low cost using cloaked locations.**

I. Introduction

The widespread prevalence of smart devices has created an established platform for mobile crowd sensing [1]. Individuals with sensing and computing devices are able to collect and contribute valuable data about events or phenomena of interest. An interesting and valuable class of crowd sensing applications is Participatory Sensing (PS) [2] in which participants are actively involved (as versus being autonomous and minimally involved in opportunistic sensing) to collect and contribute data as workers[1]. The goal is to collect data through the participants about specific targets that could be objects, events, or phenomena at particular locations during a time period. Participants might need to travel to the location of the assigned targets to collect data using their smart devices or phones. Examples of these systems include crowd-contributed instant news coverage, trail condition updates after storms [3], and urban texture documentation [4]. This can be also considered as one type of location-aware crowdsourcing applications [5]–[7] in which tasks are distributed to participants with regard to their locations.

To maximize the coverage of data collection in such crowdsourcing systems, a spatial task management server might be used to distribute sensing tasks to the participants based on their locations effectively. Several projects have focused on the optimization of spatial task assignment to improve the sensing process [4], [9]. However, a major concern of all these systems is the location privacy of participants. While participants can conceal their identity by anonymous contribution, their location is a required piece of information for effective spatial task assignment – disclosing which can reveal their identity or other private attributes. One promising approach to preserve location privacy is spatial cloaking that has been widely used in location-based services [10]–[13]. However, spatial cloaking results in uncertain locations, challenging the task assignment process.

In this paper, we consider the spatial task assignment problem in a coordinated crowd sensing setting in which a tasking server is responsible for managing sensing tasks among participants who share their cloaked locations rather than their exact locations. Our goal is to efficiently assign sensing tasks to participants based on their cloaked locations to achieve a desired coverage goal with minimized cost, i.e. the total distance that participants have to travel for their assigned tasks.

Our main contributions are summarized below. First, we propose a novel two-stage optimization approach for the spatial task assignment problem in the presence of cloaked locations. In the first stage, a global optimization problem is solved at the task assignment server using cloaked locations. Our approach addresses location uncertainty and can work with different spatial cloaking methods. In the second stage, participants individually fine-tune their assignments using their own exact locations. We formulate formal optimization objectives for each stage and further show the optimization problems at each stage are NP-hard. Second, we propose efficient greedy algorithms to solve the optimization problem at each stage. Finally, we present extensive experiments using real and synthetic data and show the impact of various parameters on our algorithms and demonstrate the feasibility and benefit of our approach.

The remainder of this paper is organized as follows. In Section II we give an account of previous work. In Section III we present a comprehensive definition of the problem. This also includes formal objectives for the problem and computational complexity analysis for each objective. Our proposed methods and efficient algorithms to solve the problem are presented in Section IV. Our new findings and results are described in Section V. Finally, Section VI gives the conclusions.

II. Previous work

A. Task Management in Mobile Sensing

We categorize task management in mobile crowd sensing into two major approaches: (i) Autonomous task selection, and (ii) Coordinated task assignment. In *autonomous task selection*,

[1]In this paper, we use participants and workers interchangeably since our methods can be adopted for both voluntary or incentive-based models.

participants select their tasks autonomously from a set of existing tasks received from a task distribution entity. They might or might not inform the distributor about their selection choices. Since the selected tasks are not optimized globally, these approaches might not be efficient with respect to sensing cost or global utility. Examples of these approaches may be found in [14]–[16]. A survey of existing methods in which participants select a task autonomously without revealing their identity or location can be found in [17]. Our approach is different from these works since none of them guarantee the efficiency of the selected tasks globally.

Coordinated task assignment aims at optimizing the process of data sensing by efficient assessment of available sensing resources to meet the requirements of applications. The criteria for optimization of task assignment include sensing costs, coverage of targets of interest, quality, and credibility of sensed data. Examples of this approach can be found in [4], [9], [18], [19]. Reddy et al. [9] proposed a coverage-based task assessment that finds the least costly subset of participants to achieve the coverage goal. Shirani-Mehr et al. [4] also proposed a coverage-based task assignment method for assigning viewpoints to a group of moving participants. None of these studies considers location privacy restrictions. In [18], the authors proposed a data acquisition framework for PS applications that assess sensing resources to answer queries from different PS applications efficiently. Their assignment criteria include sensing costs and quality of the query answers evaluated by the query initiators. However, their proposed model requires knowledge of the exact location of participants to assign tasks effectively, hence they only mitigate the privacy problem by adjusting the duration between consecutive location disclosures. Another work [19] proposes a push method to upload tasks onto mobile phones selectively. Since the tasking server learns the locations of the participants during registration, the server is able to track the mobile phones for a limited time. Hence, participants are required to wait for a random amount of time before registering again. Our work differs from these approaches since we use cloaked locations of participants for task assignments, thereby ensuring that the server does not learn the exact location of the participants. We also propose a novel two-stage optimization method to handle uncertainty.

B. Location Privacy

To protect location privacy of individuals in location-based services, location obfuscation methods have been studied widely in the literature (see recent surveys in [13], [20], [21]). One typical obfuscation method is spatial cloaking or perturbation which hides the user's location inside a cloaked region using spatial transformations [22] or a set of dummy locations [23] in order to achieve uncertainty based privacy [24] or anonymity based privacy [10], [25]–[29]. Most recently, the work [11] proposed a location perturbation method based on a rigorous notion of indistinguishability, which is similar to the differential privacy concept [30]. In our work, we assume that the location of each participant is hidden in a cloaked spatial region with certain probability distribution (can be inferred probabilities based on perturbed locations) without considering other details of the underlying obfuscation algorithm. Therefore, our method can work with any cloaking or perturbation methods for location privacy. We note that differential privacy has also been recently applied to anonymize aggregate location or trajectory data [13], however, these works are not applicable in location based services setting when individual locations (albeit can be uncertain) are needed.

III. Two-Stage Optimization Approach

In this section, we first define the spatial task assignment (STA) problem and then we formulate a version of STA which deals with cloaked locations (STAC) as a two-stage optimization problem.

A. Problem Definition

Figure 1 illustrates a high-level design for task management in a crowd sensing architecture. In our work, we focus on three main components of this architecture including participants, applications and the tasking server. The applications are requesters of the data acquired via sensors carried/operated by participants. Our task management service referred to as the tasking server recruits suitable participants for applications. To this end, the applications upload their required tasks to the tasking service. A task includes a set of targets of interest and required sensing specifications such as type of sensing, required equipment, and sampling frequencies. Similarly, participants who are registered to this service via a trusted third-party anonymizer, provide their attributes including their capabilities such as their smart-device specifications, their spatial availability as cloaked areas, their temporal availability, and other restrictions such as their mobility limitations. In this section, we provide a more formal description of the spatial task assignment problem with cloaked locations. The summary of notations is presented in Table I. First, we formally define,

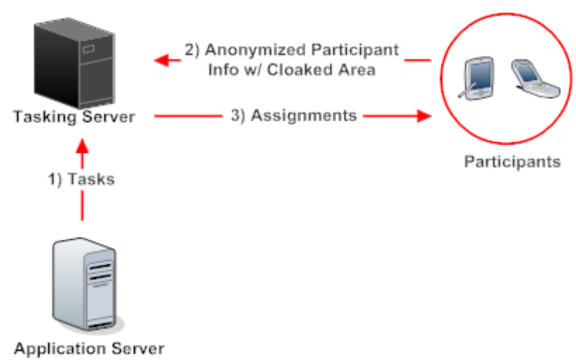


Fig. 1: Task assignment in a crowd sensing architecture.

who is a participant, and what is a cloaked area.

Definition 1: (Participant) A participant p_i is an anonymously registered user who has a limited travel budget b_i, i.e. the maximum distance a participant can travel. The participant shares this information as well as her cloaked area a_i (defined later in this section) and her desired sensing time with the tasking server. The participant's true location l_i is considered private and is not shared with the server.

Definition 2: (Cloaked Area) A cloaked area for a participant is a pair $\langle a_i, f_i \rangle$, where a_i is a spatial region and f_i is the probability density function of the participant at each point in

TABLE I: Notations

p_i	Participant i
t_j	Target j
n	Number of participants
m	Number of targets
b_i	Travel distance budget of participant i
l_i	Location of the participant i
a_i	Cloaked area of participant i
k_j	Required coverage for target j
g	Required fraction of task coverage
$d_{i,j}$	Distance between the participant i and target j
$\mathbf{x}$	First stage assignment matrix
$\mathbf{y}$	Second stage assignment matrix
$\hat{\mathbf{d}}$	Expected distance matrix

a_i. For simplicity, we refer to the cloaked area as a_i in this paper.

Each participant is able to perform tasks that meet their restrictions. The following definitions describe what is a task, its assignment, and its coverage.

Definition 3: (Task) A task specifies a set of targets for data collection, the location of each target, the required coverage for each target (k_j) (i.e. the number of participants to cover target t_j), and the overall coverage goal g (the required portion of the task coverage defined later in this section).

Definition 4: (Task Assignment) Task assignment is a mapping of participants to targets in a task shown by a matrix **x** where $x_{i,j} = 1$ if target $j \in M$ is assigned to participant $i \in N$, otherwise $x_{i,j} = 0$. $N := \{1, \ldots, n\}$ is a collection of row indexes (or participants), $M := \{1, \ldots, m\}$ is a collection of column indexes (or targets).

Definition 5: (Task Coverage) Coverage for a target is defined as the number of participants assigned to it, normalized by the required coverage of the target k_j. Task coverage (TU) is defined as the sum of coverage for all the targets in the task shown in (1). The maximum value of TU for full coverage is m. Coverage goal for a task denoted as g indicates the required fraction of task coverage with $g \in (0, 1]$.

$$TU = \sum_{j \in M} \frac{\sum_{i \in N} x_{i,j}}{k_j} \tag{1}$$

Definition 6: (Task Cost) The sensing cost for a pair of participant and target can be defined based on the travel distance, sensing duration or the complexity of each sensing. Since the participants may need to travel to the target location, we define a cost model which is simply the Euclidean distance between the participant's original location and assigned targets shown as a matrix **d**. Task cost (TC) is defined as the sum of all sensing costs for all targets in the task. Our cost model can be substituted by any other distance-based cost model without affecting the problem definition.

$$TC = \sum_{j \in M} \sum_{i \in N} x_{i,j} d_{i,j} \tag{2}$$

Given a set of participants and a task, we can define the task assignment problems as follows.

Definition 7: (STA: Spatial Task Assignment) For a set of participants and the set of targets in a task, the spatial task assignment problem (STA) formulated below aims at achieving the task coverage goal with the minimum cost by assigning targets to qualified participants using their exact locations.

$$\begin{aligned} \min_{x} \; & \sum_{i \in N} \sum_{j \in M} d_{i,j} x_{i,j} \\ \text{s.t.} \; & \sum_{j \in M} \frac{\sum_{i \in N} x_{i,j}}{k_j} \geq gm \\ & \sum_{j \in M} x_{i,j} d_{i,j} \leq b_i \end{aligned} \tag{3}$$

where the minimization objective is to minimize the task cost TC, defined in (2). The first constraint indicates that the task coverage TU, defined in (1), has to be greater than or equal to the required task coverage gm. The second constraint represents the travel budget of each participant (i.e. the total travel distance for participant p_i can not exceed her travel budget b_i).

Definition 8: (STAC: Spatial Task Assignment with Cloaked Locations) For a set of participants and the set of targets in a task, STAC aims at achieving the task coverage goal with minimum cost by assigning targets to the qualified participants using their cloaked locations. We formulate this problem as a two-stage optimization objective in section III-B.

B. *Formal Two-stage Optimization Objective*

In spatial task assignment with cloaked locations (STAC), since exact locations of the participants are not provided to the server, the distance between targets and participants described by the matrix **d**, used as the sensing cost matrix, is unavailable to the tasking server. Therefore the server is required to deal with location uncertainty and estimate the values of **d** as an expected distance matrix $\hat{\mathbf{d}}$. Then, the server can utilize these expected values to perform the task assignment. However, this uncertainty introduces inaccuracy in distance estimations and subsequently in task assignments. Hence, we propose a two-stage optimization solution to solve STAC. The first stage optimization problem is a global task assignment problem (G-STAC) based on uncertain locations solved at the tasking server, while the second is a local task assignment problem (L-STAC) solved by each participant. Dividing the assignment task into two separate problems utilizes participant location data locally while preserving participant privacy. The goal of the second stage is to refine and optimize task assignment results of the first stage by each participant using her exact location. In this section, we describe each stage in detail and then propose a formal objective for each problem.

1) G-STAC : First stage optimization objective: The first stage deals with uncertain locations which leads to uncertain distances for participant-target pairs. Assuming we had exact locations, the first-stage optimization objective would be as shown in (3). However, in absence of exact locations, we need to estimate distances as $\hat{\mathbf{d}}$. We discuss the estimation process with more details in Section IV-A.

2) L-STAC : Second stage optimization objective: Our second stage optimization runs in the participant's device locally using the given assignment from the first stage. Since new information is introduced in the second stage (i.e., exact locations available in each participant's device), these assignments can be adjusted and refined for more coverage and/or lower distance/cost. The reason is that (i) some targets might have been assigned to the participant by the server based on the estimated distance, but they are not actually accessible to the participant as the exact distance may exceed her travel budget; (ii) some targets are very close to the participant but have been estimated as being farther and not assigned. If each participant simply selects the closest targets in the second stage, however, over-coverage may occur for some of the targets meaning they might be covered more than required. Therefore, in addition to minimize the total travel distance with the exact location in the second stage, we would like to keep the assignments of the first stage unchanged as much as possible because they have been globally optimized for the global coverage goal and cost. The objectives of second stage assignment optimization of each participant $p_i, i \in N$ is shown in (4).

$$\begin{aligned} & \min_{y} \sum_{j \in M} d_{i,j} y_{i,j} \\ \text{s.t.} \quad & |\mathbf{y}_i - \mathbf{x}_i| < \epsilon \\ & \sum_{j \in M} \frac{y_{i,j}}{k_j} \geq \sum_{j \in M} \frac{x_{i,j}}{k_j} \\ & \sum_{j \in M} y_{i,j} d_{i,j} \leq b_i \end{aligned} \tag{4}$$

where for each participant p_i, $\mathbf{x}_i$ is the first stage assignment vector, $\mathbf{y}_i$ is the second stage assignment vector, $\mathbf{d}_i$ is the distance vector, b_i is the participant's travel budget, $|\mathbf{y}_i - \mathbf{x}_i|$ is the Hamming distance between two binary vectors $\mathbf{x}_i$ and $\mathbf{y}_i$ which is constrained using a small threshold ϵ in favor of keeping the first-stage assignments unchanged as much as possible. The second constraint guarantees that p_i's contribution to the task coverage is at least equal to the coverage share assigned to her in the first stage. The last constraint guarantees that her travel distance is within her budget.

C. Complexity Analysis

In this section we show that our global and local problems are NP-hard, by reducing the minimum set cover problem to G-STAC and L-STAC. The minimum set cover problem is a well studied NP-hard problem defined as follows.

Definition 9: (Minimum Set Cover Problem [31]) Given a universe W, a collection S of subsets of W, and a cost function $c : S \to \mathbb{R}_+$ find a minimum cost sub-collection of S that covers each element of W.

Theorem 1: The G-STAC is an NP-hard optimization problem.

Proof: To prove that G-STAC is NP-hard we show a polynomial reduction of the minimum set cover problem (Definition 9) to our problem.

Consider a minimum set cover problem with $W = \{p_1, \ldots, p_n, p_\emptyset, t_1, \ldots, t_m, t_\emptyset\}$ and S being a set of two-element subsets of W, i.e., $S = \{\{p_i, t_j\} : p_i \in W, t_j \in W\}$. Let $k > 0$ and $c : S \to \mathbb{R}_+$ be a cost function such that $c(\{p_i, t_j\}) = \hat{d}_{i,j}$ ($p_i \neq p_\emptyset$ and $t_j \neq t_\emptyset$) is an expected distance between t_j and p_i. For remaining elements of S the cost function is defined as follows: $c(\{p_i, t_\emptyset\}) = 0$ and $c(\{p_\emptyset, t_j\}) = D$, where $t_j \neq t_\emptyset$ and $D > \sum_{i \in N, j \in M} c(\{p_i, t_j\})$.

We reduce such instance of the minimal set cover problem to the G-STAC problem as follows. Let $P = \{p_i : i = 1, \ldots, n\}$ be a set of participants and $T = \{t_j : j = 1, \ldots, m\}$ be a set of targets. A distance between t_j and p_i is equal to $d_{i,j} = c(\{p_i, t_j\})$. We assume G-STAC has an optimal solution $\mathbf{x}_{OPT}$ with minimum cost and full coverage (when setting $g = 100\%$ and $k = 1$). We derive $S_{OPT} \subset S$ from $\mathbf{x}_{OPT}$ for the minimal partial set cover problem as follows. If t_j is assigned to p_i in $\mathbf{x}_{OPT}$, i.e. $x_{i,j} = 1$, then $\{p_i, t_j\} \in S_{OPT}$. If t_j is not assigned to any participant in $\mathbf{x}_{OPT}$, then $\{p_\emptyset, t_j\} \in S_{OPT}$. If p_i has no target assigned to it in $\mathbf{x}_{OPT}$, then $\{p_i, t_\emptyset\} \in S_{OPT}$. If all targets have been assigned and each participant has at least one target assigned to it in $\mathbf{x}_{OPT}$, then $\{p_\emptyset, t_\emptyset\} \in S_{OPT}$.

We show by contradiction that S_{OPT} covers set W with the minimal cost, i.e., any other solution would not have lower cost. Let us assume by contradiction that there is S' that covers W with lower cost. Elements of S' can be mapped to assignment pairs of participants to targets, therefore they define a solution $\mathbf{x}'$ for the G-STAC problem. Since S has a lower cost than S_{OPT}, then $\mathbf{x}'$ has a lower cost than $\mathbf{x}_{OPT}$. This is a contradiction with $\mathbf{x}_{OPT}$ being the optimal solution of G-STAC. ■

Similarly, we can show that L-STAC is NP-hard by a polynomial reduction of the minimum set cover problem (Definition 9) to it.

IV. Algorithms

In this section, we propose efficient greedy algorithms to approximate the optimization objectives for both G-STAC and L-STAC.

A. First Stage: G-STAC

We first present two methods to deal with location uncertainty in the first stage, then we propose an efficient greedy algorithm to approximate the optimization objective for G-STAC based on the greedy solution proposed in [32] for partial set cover problem.

1) Distance Estimation: As mentioned earlier, we use a distance-based cost model in our work which defines the sensing cost as the Euclidean distance between participants and targets. Therefore, our tasking server is required to deal with the location uncertainty of the participants to estimate distances. Queries over uncertain spatio-temporal data have been extensively studied with many algorithms to handle queries such as nearest neighbors, top-k, and range queries [33]. However, most of them aim at ranking the query answers and cannot be directly adopted in our work which requires actual distances to optimize the sensing cost. Knowing the cloaked areas (as the pair of the area and the probabilistic density function $\langle a, f \rangle$), we propose two simple methods to calculate the expected distances.

i) Centroid-point: In this baseline method, we calculate the centroid of all points in the cloaked area $z \in a$ as the expected location of the participant and use it to calculate the expected distances $\mathbf{d}_{i,j}^{\hat{}}$.

$$\hat{d}_{i,j} = dist(\int_{z \in a_i} z f_i(z) dz, l_j)$$

where l_j is the location of the target j and the function $dist$ is the Euclidean distance between two points.

ii) Expected-probabilistic: In this method, for each pair $\langle i, j\rangle$ of participant-target, we first calculate the probability of the target j being accessible by the participant i as $\rho_{i,j}$ (i.e., the probability that target j is within the travel budget of the participant i). To calculate this probability, we apply a simple pruning approach for each participant-target pair and shrinks the cloaked area a_i to a'_i which is the area of intersection between a_i and the circle centered at target j with radius b_i (i.e. the distance budget of participant i). Then, having the probability density function f_i, we calculate the probability of the participant being in a'_i which is equal to the probability of the target j being within the travel budget of participant i ($\rho_{i,j}$).

$$\rho_{i,j} = \int_{z \in a'_i} f_i(z)dz$$

Finally, we compute $\hat{d}_{i,j}$ as the expected distance between the target and the intersection area a'_i with probability $\rho_{i,j}$.

$$\hat{d}_{i,j} = \frac{\int_{z \in a'_i} dist(z, l_j) f_i(z)dz}{\int_{z \in a'_i} f_i(z)dz}$$

The above estimation methods can work with any cloaking area or can be discretized to work with a set of perturbed locations associated with probabilities. Figure 2 illustrates our estimation approaches when participant location is cloaked in a circular region with uniform probability distribution function.

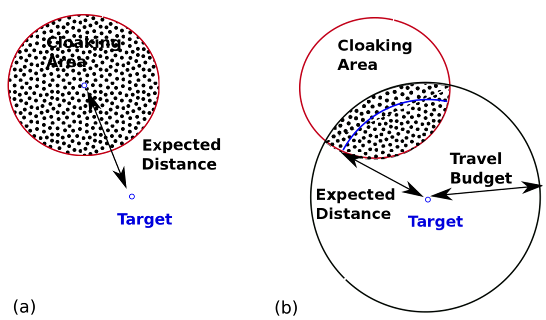


Fig. 2: (a) Centroid-point method, (b) Expected-probabilistic method.

2) Greedy Algorithm: Algorithm 1 represents the pseudocode for an efficient greedy algorithm to approximate the solution of our first stage objective. It iteratively picks the most cost-effective participant-target pair and updates the current coverage for the target, until either the coverage goal is met or all travel budgets of participants are exhausted. Since both the number of targets to be assigned and all travel budgets do not increase in time and always have non-negative values, the number of updates is finite. In each iteration, the algorithm finds the most cost-effective participant-target pair and assigns them to each other. For a participant $p_i, i \in N$ and target $t_j, j \in M$, the cost-effectiveness of assigning them to each-other is calculated as $\phi^{(1)}_{i,j}$.

$$\phi^{(1)}_{i,j} = \frac{\hat{d_{i,j}}}{\min(1 - u^+_j, \frac{1}{k_j}) + \epsilon}$$

which is the ratio of expected distance $\hat{d}_{i,j}$ (cost) to the expected coverage contributed by this participant. $\mathbf{u}^+$ is the vector of current covered portions of the targets which is initially all zero. If a target is fully covered, the corresponding value of this target in $\mathbf{u}^+$ becomes 1. The expected coverage contributed by participant p_i for target t_j is hence $\min(1 - u^+_j, \frac{1}{k_j})$, the minimum of remaining required coverage of t_j and the coverage p_i can offer for t_j. Finding the minimum aims at preventing over-coverage. The small positive value ϵ is added to avoid overflow when the expected coverage by the participant is zero.

Since one of our distance estimation methods is probabilistic, Algorithm 1 is designed to select the most cost-effective pair of participant-target $\langle i, j\rangle$ with probability $\rho_{i,j}$. For the Centroid-point method, these probabilities are calculated as

$$\rho_{i,j} = \begin{Bmatrix} 1 & \hat{d}_{i,j} \leq b_i \\ 0 & \hat{d}_{i,j} > b_i \end{Bmatrix}$$

For the probabilistic method, $\rho_{i,j}$ is calculated as described in section IV-A1. Algorithm 1 finds the answer in one pass through all participant-target pairs for the centroid-point method because the probabilities are either 0 or 1. For the expected-probabilistic method an upper-bound threshold R is used in the algorithm to stop the algorithm after R passes through all possible pairs. While the algorithm will converge after sufficient number of passes, we use R mainly for experiment purposes and enhanced efficiency. At the end of the first stage, the covered proportion of targets is updated in $\mathbf{u}^+$ based on the first stage assignments. Therefore, we refer to it as the expected coverage vector which is passed to the participant in the second stage along with her first stage assignment and the set of her accessible targets.

Time Complexity. The time complexity of our distance estimation methods are $O(nms)$ where n is the number of participants, m is the number of targets, and s is the number of points (sampling points when f_i is continuous) in each participant's cloaked area. Algorithm 1 runs in $O(nm)$ for the centroid-point method because the probabilities are either 0 or 1, so the algorithm finds the answer in one pass through all participant-target pairs. For the expected-probabilistic approach, the algorithm will run no more than R rounds for all participant-target pairs, so the the time complexity is $O(Rnm)$.

B. Second Stage: L-STAC

Due to the uncertainty of locations used in the first stage, a participant might be assigned to targets that are not accessible while not being assigned to targets that are accessible. The main goal of the second stage is for each participant to fine-tune the assignment using her exact location while maintaining the overall coverage goal. However, the main pitfall in the second stage is over-coverage of some targets at the cost of under-coverage of others, i.e. more participants are assigned to some targets than required, if each participant simply optimizes its cost by selecting the closest targets in the second stage. Hence, we have an additional constraint to bound the overall changes in the assignments implemented by several heuristics in order to maintain the overall coverage goal without incurring additional cost. The server provides the expected coverage vector, the final $\mathbf{u}^+$ at the end of G-STAC to all participants.

Algorithm 1 A greedy algorithm for the first stage task assignment problem

Input: P (set of participants), T (set of targets), **b** (vector of travel budgets), $\hat{d}$ (matrix of expected distances), k (vector of required coverage for targets), g (task coverage goal), ρ (matrix of the access probabilities), R (threshold on running rounds)
Output: **x** (matrix of task assignments), $\mathbf{u}^+$ (vector of covered portion of targets)

All elements of **x** and $\mathbf{u}^+$ are initialized to 0
$TC \longleftarrow 0$
$r \longleftarrow 0$
while $(TC \leq gm)$ **and** $(r < R)$ **do**
  if a remaining probable pair exists **then**
    Select the most cost-effective target-participant pair from the remaining pairs, say indexed at i and j with the probability $\rho_{i,j}$.
    if a pair is selected **then**
      Assign the selected pair as $x_{i,j} \longleftarrow 1$
      $TC \longleftarrow \min(1 - u_j^+, \frac{1}{k_j}) + TC$
      $u_j^+ \longleftarrow \min(1 - u_j^+, \frac{1}{k_j}) + u_j^+$
      $b_i \longleftarrow b_i - \hat{d}_{i,j}$
      if $u_j^+ = 1$ **then**
        $T \longleftarrow T \setminus T_j$
      end if
      if $b_i = 0$ **then**
        $P \longleftarrow P \setminus P_i$
      end if
    else
      $r \longleftarrow r + 1$
    end if
  else
    Break
  end if
end while

Hence, the $\mathbf{u}^+$ at the beginning of second stage optimization is initialized with the given values from the server.

Algorithm 2 presents the pseudocode for our greedy approach to approximate the solution of our second stage objective. This algorithm runs locally on each participant's device $p_i \in P$, so it has access only to the corresponding participant's attributes including her exact location, and the information provided by the server including the set of candidate targets τ that may be accessible by the participant (the server can prune the targets that are not accessible by the participant if the minimum possible distance between a participant and a target is larger than b_i), and the result of the first stage assignment for this participant $\mathbf{x}_i$. The result of assignments in this algorithm is stored in $\mathbf{y}_i$.

The algorithm at participant p_i first initializes all elements of its assignment vector to 0 (no assignment) and updates $\mathbf{u}^+$ so it only contains the coverage contributed by all other participants, i.e. by removing the current targets assigned to p_i from the first stage (line 1-8). Similar to Algorithm 1, the algorithm then iteratively picks the most cost-effective target and assigns it to p_i with some probability which is designed to avoid over-coverage. In contrast to Algorithm 1, since we want to satisfy the the first constraint of (4), we penalize each new assignment which is different from $x_{i,j}$. Therefore, the cost-effectiveness score of each assignment in this stage is calculated as $\phi_{i,j}^{(2)}$.

$$\phi_{i,j}^{(2)} = \frac{\frac{d_{i,j}}{b_i} + |x_{i,j} - 1|}{\min(1 - u_j^+, \frac{1}{k_j}) + \epsilon}$$

which is the ratio of second stage cost (i.e., the sum of normalized distance and change penalty) to the expected coverage contributed by this participant for target $t_j \in \tau$. The expected coverage contributed by the participant is computed the same as in the first stage. The other difference of our second stage algorithm from the first stage concerns the probabilities which are used to assign targets to participants. For a target j, $\rho_{i,j}$ is calculated as follows,

$$\rho_{i,j} = 1 - \frac{\phi_{i,j}^{(2)}}{\max\left\{\phi_{i,j}{}^{(2)}\right\}}$$

Using this probability, we aim at avoiding over-coverage of the targets, but at the same time reducing the chances of costly assignments. Without this probability, participants would repeatedly assign targets until their travel budget is exhausted. Completely expending the travel budget by all participants can result in over-coverage with high cost. This effect can be seen easily in the autonomous task selection methods discussed in Section II. Using $\phi^{(2)}$ to calculate this probability favors more cost-effective assignments by giving them a higher probability. Similar to Algorithm 1, for efficiency purposes, an upper-bound threshold R' is used in the algorithm to stop the algorithm after R' passes through all targets in τ.

Algorithm 2 A greedy algorithm for the second stage task assignment problem

Input: p_i, $i \in N$ (the participant), τ (set of accessible targets for p_i), $\mathbf{x}_i$ (first stage assignments for p_i), b_i (p_i's travel budget), **k** (vector of required coverages for targets), g the task coverage goal, $\mathbf{u}^+$ (vector of covered portion of targets), R' (threshold on running rounds)
Output: $\mathbf{y}_i$ (vector of task assignments for p_i)

1: All elements of $\mathbf{y}_i$ are initialized to 0
2: $LC \longleftarrow 0$ (Local task coverage) $AC \longleftarrow 0$(Assigned task coverage to this participant)
3: $r \longleftarrow 0$
4: **for** all the targets in τ **do**
5:   $u_j^+ \longleftarrow u_j^+ - \frac{x_{i,j}}{k_j}$
6:   $AC \longleftarrow AC + \frac{x_{i,j}}{k_j}$
7: **end for**
8: **while** $(LC < AC)$**and**$(b_i > 0)$**and**$(r < R')$ **do**
9:   **if** a remaining probable target exists in τ **then**
10:     **if** target j is selected **then**
11:       **if** $d_{i,j} \leq b_i$ **then**
12:         Assign the selected target as $y_{i,j} \longleftarrow 1$
13:         $TC \longleftarrow TC + \min(1 - u_j^+, \frac{1}{k_j})$
14:         $u_j^+ \longleftarrow u_j^+ + \min(1 - u_j^+, \frac{1}{k_j})$
15:         $b_i \longleftarrow b_i - d_{i,j}$
16:         $\tau \longleftarrow \tau \setminus \tau_j$
17:       **end if**
18:     **else**
19:       $r \longleftarrow r + 1$
20:     **end if**
21:   **else**
22:     Break
23:   **end if**
24: **end while**

Time Complexity. Algorithm 2 runs no more than R' rounds of passing through all targets in τ, so having the number of all targets as m, the time complexity is $O(R'm)$.

V. EXPERIMENTAL RESULTS

In this section, we evaluate our task assignment methods experimentally using both real and synthetic datasets. First we discuss the details of our experiment settings, then we present and analyze the results.

A. Settings

Datasets. We used a real dataset from Gowalla [34] which contains check-in information of users of a location-based social network. The check-ins consist of time and location coordinates of users at different positions. For our experiments, we used user and position information during October 2010 in New York city. We used each day as a snapshot for our task assignment experiment. In all experiments, participants are selected uniformly from all Gowalla users on each given day, while targets are picked randomly from all the spots.

We also used Brinkhoff's Network-based Generator of Moving Objects [35] to create a set of synthetic dataset of moving objects (OLE) to test our methods and algorithms. The map of the city of Oldenburg in Germany is used as the input to the generator. In OLE, at each time snapshot, the set of participants is chosen uniformly from the set of moving objects in the map. In the same way, targets are selected from the nodes of the road graph of the map.

Evaluation Metrics. Task cost (TC) and task coverage (TU) are calculated as described in section III. In many settings, the desired coverage goal (g) may not be achievable, even given exact locations of the participants, due to the limited number of participants or travel budgets of the participants. Hence, we also present a combined cost metric that adds the task cost (TC) and uncovered targets ($gm - TU$) and normalizes the sum to the range of [0,1] using min-max method. We refer to this normalized value as penalized cost (PC):

$$PC = \frac{(gm - TU) + TC}{gm + \sum_i b_i}$$

The denominator is used for normalization and is equal to the maximum possible value for the sum of uncovered portion of the task and the task cost. Thus, a smaller value of PC represents higher coverage and lower cost, which is considered a better result. In our experiments, we study the effect of different parameters such as the number of participants/targets, coverage goal, and cloaking size on the task cost and coverage.

Parameters. Table II shows the parameter settings for our simulations with default settings highlighted. In all experiments, we selected the travel budget of the participants randomly in the range of 1%-3% of the map size. We experimented with different cloaking models such as circular and rectangular areas, continuous/discrete instance sampling, and uniform/normal probability distribution of the instances; however, since the results were very similar for different cloaking models, we only present results for a circular model with continuous uniform distribution of instances. The size of the cloaking area for each participant is roughly selected uniformly in the range of 0.01%-1% of the map area. For all of our experiments with the OLE dataset, we used a part of the map of Oldenburg highlighted in Figure 3. The reason of this selection is to provide a more crowded map, compared to Gowalla which is a sparse dataset with a maximum of 235 participants in each snapshot. The required coverage of each target (k) is set to one in all experiments because varying the required target coverage has a similar effect as varying number of targets. The coverage goal g is selected between 10%-100% of the number of targets, with a default value of 100%. R and R' vary depending on the size of the cloaking area, but in general they are selected in the range of [50,200]. We repeated each experiment for 100 times and obtained their average as our final results.

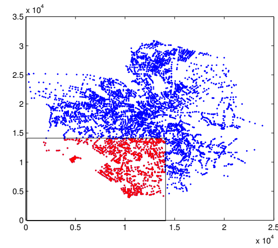


Fig. 3: The map of Oldenburg, Germany generated by [35]. The employed section is framed.

Comparisons. We report the results of the following methods based on the combination of different distance estimation model of the G-STAC (Centroid-point or Expected-probabilistic) and the optimization stages (one-stage G-STAC-only or two-stage G-STAC/L-STAC combination).

- CPA1 (baseline): one-stage centroid-point based approach as a baseline,
- CPA2: two-stage centroid-point based approach to demonstrate the benefit of the two-stage optimization compared to the baseline one-stage optimization approach,
- EPA1: one-stage expected-distance approach to demonstrate the benefit of probabilistic distance estimation over the baseline centroid approach,
- EPA2 (complete proposed solution): two-stage expected-probabilistic approach,
- NPA (reference solution): we utilized our first stage optimization solution with zero level of privacy as a reference solution with no privacy constraint (NPA). In NPA, we assume the tasking server has access to exact locations of the participants, therefore it runs only on the server.

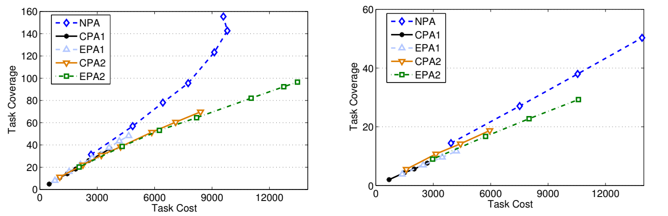
TABLE II: Experimental Settings with Highlighted Default Values

Parameter	Value
Number of Participants	50, 100, 150, **200**, 300, 400, 500
Number of Targets	50, 100, 150, **200**, 300, 400, 500
Travel Budget	1% - 3% of Map Size
Coverage Goal	10%-**100%**
Cloaking Model	**Circular**, Rectangle
Cloaking Area	0.01% - 1% of Map Area

B. Results

In this section, we report the results of each experiment for the two datasets in terms of task cost, coverage, and penalized cost. The scales for the task coverage and cost are different for the two datasets due to their different map sizes and level of sparsity.

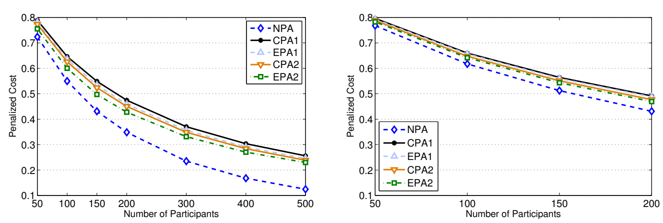
1) Impact of Numbers of Participants and Targets: In these experiments, we study the impact of increasing the number of participants and targets on the task cost and coverage by: (a) varying the number of participants while the number of targets is fixed; and (b) varying the number of targets while the number of participants is fixed.



(a) (b)

Fig. 4: Task coverage vs cost for different number of participants, and $m = 200$ using datasets (a) OLE (b) Gowalla

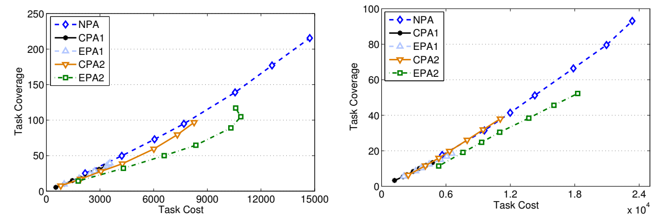
Figure 4 shows the task coverage versus cost in both datasets for increasing number of participants with a fixed number of targets using the default settings. Overlapping points in some approaches such as baseline indicate that increasing the number of participants does not affect the task cost or coverage in some cases. In both datasets, our two-stage approaches (CPA2 and EPA2) achieve a significantly higher coverage compared to the one-stage approaches (CPA1 and EPA1). Thanks to the local optimization at the second stage, they are able to get much closer to the coverage goal while the ratio of cost and coverage stays roughly the same. Moreover, the expected-probabilistic approaches outperform their corresponding centroid-point methods which is more apparent for CPA2 and EPA2. In OLE, EPA2 achieves more coverage with the same coverage/cost ratio as CPA2, but in Gowalla, this ratio is higher for CPA2, which means by using EPA2, more coverage is obtained at the expense of slightly higher cost per additional coverage due to the sparsity of the dataset.



(a) (b)

Fig. 5: Penalized cost for different number of participants, and $m = 200$ using datasets (a) OLE (b) Gowalla

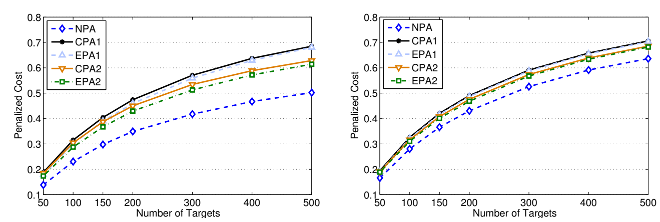
Figure 5 shows the penalized cost in both datasets for increasing number of participants with a fixed number of targets using the default settings. Increasing the number of participants results in higher task coverage which causes lower penalized costs for all of the approaches. In both datasets, for all combinations of the participants and targets, the expected-probabilistic approaches outperform their corresponding centroid-point approaches. However, this is more clear in the OLE dataset due to higher task coverage being possible. On the other hand, regardless of the distance estimation methods, both two-stage methods outperform the one-stage methods.



(a) (b)

Fig. 6: Task coverage vs cost for different number of targets, and $n = 200$ using datasets (a) OLE (b) Gowalla

Figure 6 shows task coverage versus cost in both datasets for increasing numbers of targets with a fixed number of participants using the default settings. Overlapping points in some approaches such as baseline indicate that increasing the number of targets does not affect the task cost or coverage in some cases. In both datasets, our two-stage approaches (CPA2 and EPA2) achieve higher coverage compared to the one-stage approaches (CPA1 and EPA1). Moreover, for the same number of targets, the expected-probabilistic approaches achieve higher coverage compared to their corresponding centroid-point methods. All methods are robust preserving a constant coverage/cost ratio, however, while CPA2 keeps a ratio comparable to NPA, EPA2 achieves more coverage at the expense of higher cost per additional coverage.

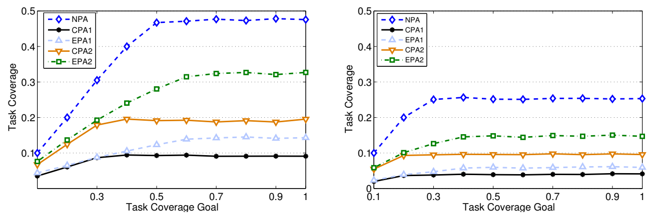


(a) (b)

Fig. 7: Penalized cost for different number of targets, and $n = 200$ using datasets (a) OLE (b) Gowalla

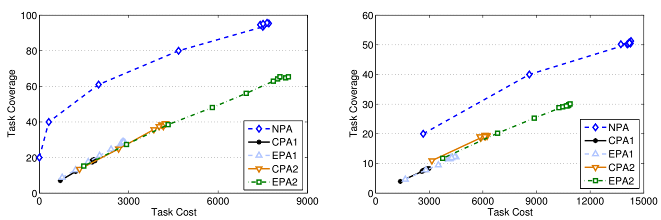
Figure 7 shows the penalized cost in both datasets for increasing numbers of targets with a fixed number of participants using the default settings. In both datasets, for all combinations of participants and targets, the expected-probabilistic approaches outperform their corresponding centroid-point approaches and similarly, both two-stage methods outperform the one-stage methods. For the same experiment settings in OLE and Gowalla, the difference between penalized cost of different methods including NPA is smaller in Gowalla.

2) Impact of Coverage Goal: Figure 8 shows task coverage in both datasets for increasing coverage goal with a fixed number of participants and targets using the default settings including 200 participants and 200 targets. In both datasets, our two-stage approaches (CPA2 and EPA2) achieve higher coverage compared to the one-stage approaches (CPA1 and EPA1). Moreover, for the same coverage goal, the expected-probabilistic approaches achieve higher coverage compared to their corresponding centroid-point methods. Comparing the real dataset Gowalla to synthetic OLE, all methods achieve higher coverage in OLE which can be explained by higher density and lower sparseness of participants.



(a) (b)

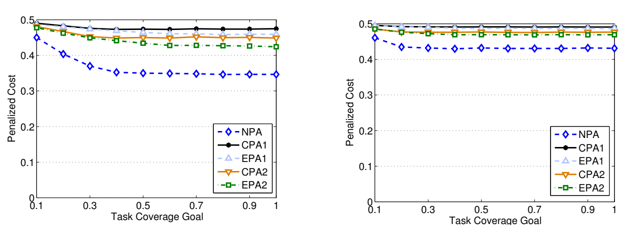
Fig. 8: Relative task coverage for different coverage goal (relative), $n = 200$, and $m = 200$ using datasets (a) OLE (b) Gowalla



(a) (b)

Fig. 9: Task coverage vs cost for different coverage goal (relative), $n = 200$, and $m = 200$ using datasets (a) OLE (b) Gowalla

Figure 9 shows task coverage versus cost for the same experiment setting. In both datasets, EPA2 achieves a higher task coverage for the same coverage goals, with a coverage/cost ratio very similar to other methods. Overlapping points in each approach indicate that changing the coverage goal does not affect the task cost or coverage in some cases.

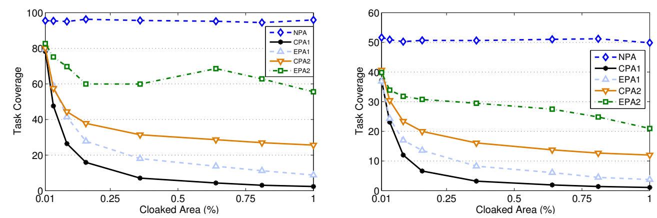


(a) (b)

Fig. 10: Penalized cost for different coverage goal (relative), $n = 200$, and $m = 200$ using datasets (a) OLE (b) Gowalla

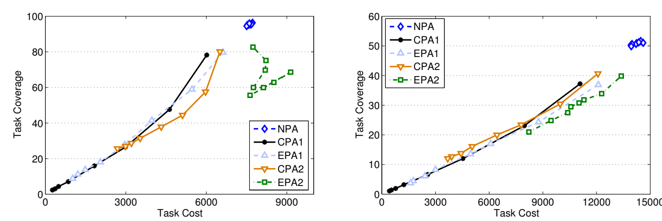
Finally, Figure 10 shows the impact of coverage goal on penalized cost. EPA2 outperforms the other methods for all values of coverage goal.

3) Impact of Cloaking Size: Figure 11 shows the impact of cloaking size on task coverage for a fixed number of participants and targets. The cloaking size is shown as a percentage of the map area. By increasing the cloaked size, in both datasets, EPA2 shows more robustness compared to CPA2 as well as the one-stage methods, indicating that EPA2 is not affected by cloaking size as much as the other methods. Figure 12 shows the impact of cloaking size on cost and coverage for the same experiment. Similarly, in both datasets, CPA1, EPA1, and CPA2 are more affected by cloaking size. Overlapping points in some approaches such as NPA indicate that changing the cloaking size does not affect the task cost or coverage in some cases. Finally, Figure 13 shows the impact of cloaking size on penalized cost. EPA2 outperforms the other methods for all cloaking sizes.



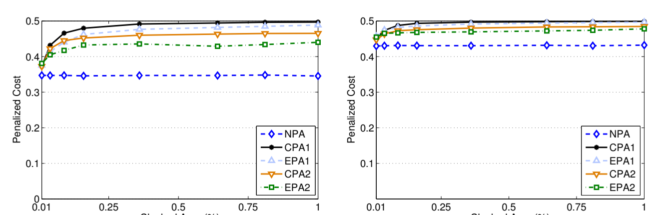
(a) (b)

Fig. 11: Task coverage for different sizes of cloaking area, $n = 200$, and $m = 200$ using datasets (a) OLE (b) Gowalla



(a) (b)

Fig. 12: Task coverage vs cost for different sizes of cloaking area, $n = 200$, and $m = 200$ using datasets (a) OLE (b) Gowalla



(a) (b)

Fig. 13: Penalized cost for different sizes of cloaking area, $n = 200$, and $m = 200$ using datasets (a) OLE (b) Gowalla

VI. Conclusions and Future Work

In this paper we defined and formulated the problem of spatial task assignment in crowd sensing with cloaked locations as a novel two-stage optimization problem. We showed the problem to be NP-hard in each stage, and proposed efficient greedy algorithms for each stage of the optimization problem. We studied the impact of parameter values including task size, participant size, coverage goal and cloaking size on our methods and showed their effectiveness and robustness. As a next step, we plan to find approximation algorithms with performance guarantees and evaluate our methods with real-time data by implementing a real-world crowd sensing application using our task assignment approach. We are also interested in extending our approach to take into account the trustworthiness of participants in terms of the quality of the data contributed by them [8].

ACKNOWLEDGMENTS

This research is supported by the Air Force Office of Scientific Research (AFOSR) DDDAS program under grant FA9550-12-1-0240. Many thanks to Daniel A. Garcia, of Emory University, for his helpful comments.

REFERENCES

[1] R. K. Ganti, F. Ye, and H. Lei, “Mobile crowdsensing: Current state and future challenges,” *Communications Magazine, IEEE*, vol. 49, no. 11, pp. 32–39, 2011.

[2] N. D. Lane, S. B. Eisenman, M. Musolesi, E. Miluzzo, and A. T. Campbell, “Urban sensing systems: opportunistic or participatory?” In Proceedings of the Workshop on Mobile Computing Systems and Applications (HotMobile). ACM Press, 2008, pp. 11–16.

[3] A. Kansal, M. Goraczko, and F. Zhao, “Building a sensor network of mobile phones.” in Proceedings of the 6th International Conference on Information Processing in Sensor Networks (IPSN), 2007, pp. 547–548.

[4] H. Shirani-Mehr, F. Banaei-Kashani, and C. Shahabi, “Efficient viewpoint assignment for urban texture documentation,” in *Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. ACM, 2009, pp. 62–71.

[5] J. Howe, *Crowdsourcing: How the power of the crowd is driving the future of business*. Random House, 2008.

[6] F. Alt, A. S. Shirazi, A. Schmidt, U. Kramer, and Z. Nawaz, “Location-based crowdsourcing: extending crowdsourcing to the real world,” in *Proceedings of the 6th Nordic Conference on Human-Computer Interaction: Extending Boundaries*. ACM, 2010, pp. 13–22.

[7] L. Kazemi and C. Shahabi, “Geocrowd: enabling query answering with spatial crowdsourcing,” in *Proceedings of the 20th International Conference on Advances in Geographic Information Systems*. ACM, 2012, pp. 189–198.

[8] L. Kazemi, C. Shahabi, and L. Chen, “Geotrucrowd: trustworthy query answering with spatial crowdsourcing,” in *Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. ACM, 2013, pp. 304–313.

[9] S. Reddy, D. Estrin, and M. Srivastava, “Recruitment framework for participatory sensing data collections,” in Proceedings of the 8th International Conference on Pervasive Computing. Springer Berlin Heidelberg, May 2010, pp. 138–155.

[10] M. Gruteser and D. Grunwald, “Anonymous usage of location-based services through spatial and temporal cloaking.” In Proceedings of the International Conference on Mobile Systems, Applications, and Services (MobiSys), 2003.

[11] M. E. Andrés, N. E. Bordenabe, K. Chatzikokolakis, and C. Palamidessi, “Geo-indistinguishability: Differential privacy for location-based systems.” ACM, 2013, pp. 901–914.

[12] C. Bettini, S. Jajodia, and P. Samarati, *Privacy in location-based applications: research issues and emerging trends*. Springer, 2009, vol. 5599.

[13] G. Ghinita, *Privacy for Location-Based Services*, ser. Synthesis Lectures on Information Security, Privacy, and Tru. Morgan & Claypool, 2013. [Online]. Available: http://books.google.com/books?id=RB-fcnlloxoC

[14] C. Cornelius, D. K. A. Kapadia, D. Peebles, M. Shin, and N. Triandopoulos, “Anonysense: Privacy-aware people-centric sensing,” vol. 5, no. 8, in Proceedings of the International Conference on Mobile Systems, Applications and Services (MobiSys). ACM Press, Jun 2008, pp. 211–224.

[15] M. Shin, C. Cornelius, D. Peebles, A. Kapadia, D. Kotz, and N. Triandopoulos, “Anonysense: A system for anonymous opportunistic sensing,” *Journal of Pervasive and Mobile Computing*, vol. 7, no. 1, pp. 16–30, 2010.

[16] D. Deng, C. Shahabi, and U. Demiryurek, “Maximizing the number of worker’s self-selected tasks in spatial crowdsourcing,” in *Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. ACM, 2013, pp. 314–323.

[17] D. Christin, A. Reinhardt, S. S. Kanhere, and M. Hollick, “A survey on privacy in mobile participatory sensing applications,” *Journal of Systems and Software*, vol. 84, no. 11, pp. 1928–1946, 2011.

[18] M. Riahi, T. G. Papaioannou, I. Trummer, and K. Aberer, “Utility-driven data acquisition in participatory sensing,” EDBT/ICDT. ACM, March 2013.

[19] T. Das, P. Mohan, V. N. Padmanabhan, R. Ramjee, and A. Sharma, “Prism: Platform for remote sensing using smartphones.” in Proceedings of the 8th ACM International Conference on Mobile Systems, Applications, and Services (MobiSys), 2010, pp. 63–76.

[20] C. A. Ardagna, M. Cremonini, E. Damiani, S. D. C. di Vimercati, and P. Samarati, “Location privacy protection through obfuscation-based techniques,” in *Data and Applications Security XXI*. Springer, 2007, pp. 47–60.

[21] M. F. Mokbel, “Towards privacy-aware location-based database servers,” in *Data Engineering Workshops, 2006. Proceedings. 22nd International Conference on*. IEEE, 2006, pp. 93–93.

[22] A. Khoshgozaran and C. Shahabi, “Blind evaluation of nearest neighbor queries using space transformation to preserve location privacy.” In SSTD, 2007.

[23] H. Kido, Y. Yanagisawa, , and T. Satoh, “An anonymous communication technique using dummies for location-based services.” In Proceedings of the IEEE International Conference on Pervasive Services (ICPS), 2005.

[24] R. Shokri, G. Theodorakopoulos, J. Le Boudec, and J. Hubaux, “Quantifying location privacy,” in *Security and Privacy (SP), 2011 IEEE Symposium on*. IEEE, 2011, pp. 247–262.

[25] M. F. Mokbel, C. Y. Chow, and W. G. Aref, “The new casper:query processing for location services without compromising privacy.” In Proc. of VLDB, 2006.

[26] B. Gedik and L. Liu, “A customizable k-anonymity model for protecting location privacy,” 2004.

[27] C.-Y. Chow, M. F. Mokbel, and X. Liu, “A peer-to-peer spatial cloaking algorithm for anonymous location-based service,” in *Proceedings of the 14th annual ACM international symposium on Advances in geographic information systems*. ACM, 2006, pp. 171–178.

[28] P. Kalnis, G. Ghinita, K. Mouratidis, and D. Papadias, “Preventing location-based identity inference in anonymous spatial queries,” *Knowledge and Data Engineering, IEEE Transactions on*, vol. 19, no. 12, pp. 1719–1733, 2007.

[29] S. Wang and X. S. Wang, “In-device spatial cloaking for mobile user privacy assisted by the cloud,” in *Mobile Data Management (MDM), 2010 Eleventh International Conference on*. IEEE, 2010, pp. 381–386.

[30] C. Dwork, “Differential privacy,” in *Automata, languages and programming*. Springer, 2006, pp. 1–12.

[31] V. V. Vazirani, *Approximation algorithms*, 2001.

[32] P. Slavík, “Improved performance of the greedy algorithm for partial cover,” *Information Processing Letters*, vol. 64, no. 5, pp. 251–254, 1997.

[33] Y. Wang, X. Li, X. Li, and Y. Wang, “A survey of queries over uncertain data,” *Knowledge and Information Systems*, pp. 1–46, 2013.

[34] “Gowalla dataset, http://snap.stanford.edu/data/loc-gowalla.html.” [Online]. Available: http://snap.stanford.edu/data/loc-gowalla.html

[35] T. Brinkhoff, “A framework for generating network-based moving objects,” *GeoInformatica*, vol. 6, no. 2, pp. 153–180, 2002.