

Privacy-Aware Spatial Task Assignment For Coordinated Participatory Sensing

Layla Pournajaf
Emory University
lpourna@emory.edu

Li Xiong
Emory University
lxiong@mathcs.emory.edu

Vaidy Sunderam
Emory University
vss@emory.edu

Slawomir Goryczka
Emory University
sgorycz@emory.edu

ABSTRACT

Participatory 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 software platforms to manage interactions between participants and applications, with a number of constraints. One such critical requirement is participant privacy. In this paper, we examine the problem of privacy-preserving spatial task assignment in coordinated participatory sensing when the participants do not share their exact location due to privacy concerns. We investigate methods for assigning participants to targets, efficiently managing location uncertainty and resource constraints. We formulate the problem as a two-stage optimization and propose greedy algorithms to approximate each stage. Simulation results show that our methods achieve high target coverage with low cost for applications while preserving privacy of the participants.

Categories and Subject Descriptors

H.2.8 [Database Management]: Database Applications—*Spatial databases and GIS*

General Terms

Algorithms, Security

Keywords

Spatial Task Assignment, Coordinated Participatory Sensing, Location Privacy, Spatial Uncertainty

1. INTRODUCTION

The widespread prevalence of smart devices has created an established platform for mobile sensing applications. An interesting and potentially valuable class of such applications is Participatory Sensing (PS) [14] in which devices sample and contribute data. Examples of these systems

include instant news coverage, trail condition updates after storms [11], collection of photos of recycling behavior at a university [17], and urban texture documentation [20]. In participatory sensing applications, the goal is to collect data through the participants from specific targets which could be objects, events, or phenomena during a time slice. To achieve this goal effectively, a spatial task management component might be used to distribute sensing tasks to the participants based on their locations. Each task includes application-based sensing requirements such as time frame or location of sensing, type of sensors, and sampling frequency [4]. The coordinated PS applications are sometimes called campaigns [17].

Several projects have focused on the optimization of task assignment to improve the sensing process [17, 20]. However, a major concern of participants in using a PS application is their privacy. While they can conceal their identity by using PS applications anonymously, their location is required information for effective spatial task assignment – which can reveal their identity. One of the promising approaches to preserve location privacy is spatial cloaking that has been widely used in location-based services [9]. However, cloaking in PS results in uncertain participant locations, challenging the task assignment process.

In this paper, we consider a coordinated data sensing approach in which a tasking server is responsible for managing and sharing sensing tasks among participants who do not share their exact locations. Our goal can be summarized as designing a spatial task assignment approach in PS to select the best set of qualified participants and efficiently assign sensing tasks to each selected participant based on their cloaked locations. Moreover, our method fulfills the PS application’s criteria and maximizes sensing coverage with minimized cost. Since participant locations are not disclosed, our method preserves their privacy against the tasking server who is the adversary in our model. The adversary is curious but not malicious, meaning that it follows the protocols and does not provide tampered or misleading information to the participants, but it might use the observed information from the participants to infer their identity or their location.

Our main contributions are summarized below. First, we propose a novel two-stage optimization approach for the privacy-aware spatial task assignment problem. In the first

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

The remainder of this article is organized as follows. In section 2 we give an account of previous work. In section 3 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 4. Our new findings and results are described in section 5. Finally, section 6 gives the conclusions.

2. PREVIOUS WORK

2.1 Task Management in Participatory Sensing

We categorize task management in PS into two major approaches: (i) Autonomous task selection, and (ii) Coordinated task assignment. In *autonomous task selection*, the 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 of their selected tasks. 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 [5, 19]. *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 PS task assignment include sensing costs, coverage of targets of interest, quality, and credibility of sensed data. Examples of this approach can be found in [17, 20, 18, 6].

Autonomous Task Selection

A survey of existing methods in which participants select a task autonomously without revealing their identity or location can be found in [4]. In [5] all tasks are downloaded by users in public places, so the tasking server can not identify them. Since this approach did not guarantee reliable and efficient sensing, the authors suggested another approach in which users connect to the server through an anonymizing network to conceal their location while downloading tasks from the tasking server [19]. Our approach is different from these works since none of them guarantee the efficiency of the selected tasks globally.

Coordinated Task Assignment

Reddy et al. [17] proposed a coverage-based task assessment that finds the least costly subset of participants to achieve the coverage goal. They also proposed a reputation-based assessment that finds the best set of participants who maximize the overall performance based on their participation

history. Shirani-Mehr et al. [20] also proposed a coverage-based task assignment method for assigning viewpoints to a group of moving participants. We propose a coverage-based task assignment, but our methods differ from the above-mentioned projects which assume no location privacy restriction.

In [18], the authors propose a data acquisition framework for PS applications that assessed sensing resources to answer queries from different PS applications efficiently. Their assignment criteria included sensing costs and quality of the query answers evaluated by the query initiators. However, their proposed model requires exact location of participants to assign tasks effectively, hence they protect privacy by adjusting the duration between consecutive location disclosure. Another approach [6] proposes a push method to upload tasks on to 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 [6]. 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 assessment, thereby ensuring that the server does not learn the exact location of the participants.

2.2 Location Privacy

To protect location privacy of individuals in location-based services, location obfuscation methods have been studied widely in the literature [7, 1, 15]. One major obfuscation method is spatial cloaking which hides the user's location inside a cloaked region using spatial transformations [12], or spatial k-anonymity [9, 16, 8, 3, 10]. Another method might hide the exact location of a user among a set of dummy locations [13]. In our work, we assume that the location of each participant is hidden in a cloaked spatial region without considering other details of the underlying obfuscation algorithm such as whether it satisfies k-anonymity or not. Therefore, our method can work with any cloaking method for location privacy.

3. TWO-STAGE OPTIMIZATION APPROACH

In this section, we first define the privacy-preserving spatial task assignment problem in coordinated PS and then we formulate it as a two-stage optimization problem.

3.1 Problem Definition

Figure 1 illustrates a high-level design for task management in a PS architecture. In our work, we focus on three main components of this architecture including participants, applications and the tasking server. The applications are consumers of the data which are acquired via the sensors carried/operated by participants. Our PS task management service referred to as the tasking server recruits suitable participants for PS 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 privacy-preserving spatial task assignment problem for coordinated PS applications. The summary of notations is presented in Table 1.

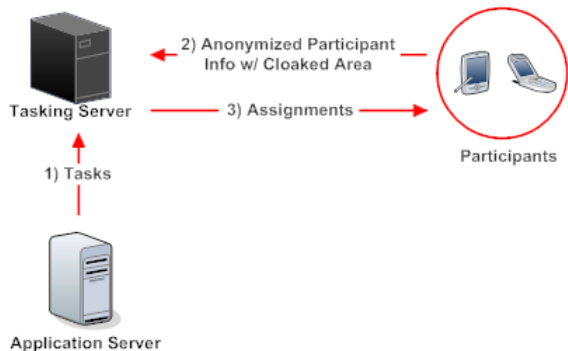


Figure 1: Task Assignment in a PS architecture.

Table 1: Notations

p_i	Participant i
t_j	Target j
n	Number of participants
m	Number of targets
r_i	Sensing range of participant i
b_i	Sensing energy of participant i
l_i	Location of the participant i
a_i	Cloaked area of participant i
k_j	Required coverage for target j
$d_{i,j}$	Distance between the participant i and target j
u_j	Coverage of target j by one participant
\mathbf{x}	First stage assignment matrix
\mathbf{y}	Second stage assignment matrix
$\hat{\mathbf{d}}$	Expected distance matrix

Definition 1. (Participant) A participant is an anonymously registered user who is modeled as a classic sensor which means she has a sensing range r and a limited energy budget b . The participant shares this sensing information with the tasking server as well as her cloaked area a (defined later in this section) and her desired sensing time. The participant’s location l is considered private and is not shared with the server.

Definition 2. (Cloaked Area) A cloaked area for a participant is a pair $\langle a, f \rangle$, where a is an spatial region and f is the probability density function of the participant at each point in a . For simplicity, we refer to the cloaked area as a in this paper.

Definition 3. (Task) A task includes a set of targets for data collection with several attributes including the location of the target, the desired time of sensing, other data collection instruction/requirements, and the required amount of

coverage for each target (k) i.e. the required number of participants to cover the target.

Definition 4. (Task Coverage) Coverage for a target is defined as the number of participants assigned to it, normalized by the required coverage of the target. Task coverage (TU) is defined as the sum of coverage for all the targets in the task.

Definition 5. (Task Cost) An assignment cost for a pair of participant and target includes the sensing cost which is calculated as the euclidean distance between the participant and the assigned target. Task cost (TC) is defined as the sum of all assignment 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.

Definition 6. (PTA: Privacy-Aware Spatial Task Assignment Problem) For a set of participants P and the set of targets T in a task, task assignment problem aims to achieve the maximum task coverage with minimum cost by assigning targets to the qualified participants using their cloaked areas instead of their exact location.

3.2 Formal Two-stage Optimization Objective

Suppose we have a set of participants P and targets T in an area of interest. At each time snapshot, every participant $p_i \in P$ submits a query to the server for possible tasks providing her cloaked area a_i , her sensing range r_i and sensing energy budget b_i . Since exact locations of the participants are not provided to the server, the distance between targets and participants described by a matrix \mathbf{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 \mathbf{d} as $\hat{\mathbf{d}}$. Then, the server can utilize the expected distances $\hat{\mathbf{d}}$ 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 the privacy-aware task assignment problem (PTA). The first stage optimization problem which is solved in the tasking server is a global task assignment problem (GPTA) is based on uncertain locations, while the second is a local task assignment problem (LPTA) which is solved locally by each participant. Dividing the assignment task into two separate problems utilizes participant location data with privacy. The goal of the second stage is to refine and optimize task assignment results of the first stage by the participant in the context of her exact location. In this section, we describe each stage in detail and then propose a formal objective for each problem.

3.2.1 GPTA : 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 Equation (1) with two objectives to

maximize coverage and minimize distance.

$$\begin{aligned}
& \min_x \sum_{i \in N} \sum_{j \in M} d_{i,j} x_{i,j} & (1) \\
& \max_x \sum_{i \in N} \sum_{j \in M} u_j x_{i,j} \\
\text{s.t.} \quad & x_{i,j} d_{i,j} \leq r_i, \quad i \in N, \quad j \in M \\
& \sum_{j \in M} x_{i,j} d_{i,j} \leq b_i
\end{aligned}$$

where $N := \{1, \dots, n\}$ is a collection of row indexes (or participants), $M := \{1, \dots, m\}$ is a collection of column indexes (or targets), matrix \mathbf{x} shows all the assignments (i.e., $x_{i,j} = 1$ iff Target j is assigned to Participant i otherwise 0), \mathbf{d} is the distance matrix of participants to targets, \mathbf{u} is the vector of coverage values which is calculated based on the required coverage of targets \mathbf{k} showing the portion of the coverage that can be offered by any participant.

$$\forall j \in M : u_j = \frac{1}{k_j}$$

Two constraints represent sensing range and energy limitations based on the distance matrix. In absence of the exact locations, we need to estimate distances $\hat{\mathbf{d}}$. We discuss the estimation process with more details in section 4.1.

3.2.2 LPTA : 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 less distance/cost. The reason is that after knowing the exact distances of participant-target pairs: (i) Some targets might have been assigned to the participant, but they are not actually within the sensing range of the participant; (ii) Some targets are very close to the participant and can be assigned with a very low distance cost, but have been estimated as being farther and not assigned. However, this might cause over-coverage for some of the targets meaning they might be covered more than required, therefore we would like to keep the assignments of the first stage unchanged as much as possible because they have been globally optimized for the goals of the application. The suggested objective of second stage assignment optimization of each participant $p_i, i \in N$ is shown in (2).

$$\begin{aligned}
& \min_y \sum_{j \in M} d_{i,j} y_{i,j} & (2) \\
& \max_y \sum_{j \in M} u_j y_{i,j} \\
& \min_y |\mathbf{y}_i - \mathbf{x}_i| \\
\text{s.t.} \quad & y_{i,j} d_{i,j} \leq r_i \\
& \sum_{j \in M} y_{i,j} d_{i,j} \leq b_i
\end{aligned}$$

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, \mathbf{u} is the utility vector, r_i is the participant's sensing range, b_i is the participant's energy budget,

and $|\mathbf{y}_i - \mathbf{x}_i|$ is the Hamming distance between two binary vectors \mathbf{x}_i and \mathbf{y}_i .

3.3 Complexity Analysis

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

Definition 7. (Minimum Set Cover Problem [21]) Given a universe W , a collection S of subsets of W , and a cost function $c : S \rightarrow \mathbb{R}_+$ find a minimum cost subcollection of S that covers each element of W k times.

THEOREM 3.1. *The GPTA is an NP-hard optimization problem.*

PROOF. To prove that GPTA is NP-hard we show a polynomial reduction of the minimum set cover problem (Definition 7) to our problem.

Let $W = \{p_1, \dots, p_n, p_0, t_1, \dots, t_m, t_0\}$ and S be 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 $c : S \rightarrow \mathbb{R}_+$ be a cost function such that $c(\{p_i, t_j\}) = \hat{d}_{i,j}$ ($p_i \neq p_0$ and $t_j \neq t_0$) is an expected distance between t_j and p_i . For remaining elements of S the cost function is defined as follows: $c(\{p_i, t_0\}) = 0$ and $c(\{p_0, t_j\}) = D$, where $t_j \neq t_0$ and $D > \sum_{i \in N, j \in M} c(\{p_i, t_j\})$.

We reduce such minimal set cover problem to the GPTA problem as follows. Let $P = \{p_i : i = 1, \dots, n\}$ be a set of participants and $T = \{t_j : j = 1, \dots, m\}$ be a set of targets. A distance between t_j and p_i is equal to $d_{i,j} = c(\{p_i, t_j\})$.

The GPTA is a multi-objective optimization problem, therefore it has many Pareto-optimal solutions. Among them we choose a solution \mathbf{x}_{OPT} with the maximal coverage, for which we define $S_{OPT} \subset S$ as follows. If t_j is assigned to p_i in \mathbf{x}_{OPT} , then $\{p_i, t_j\} \in S_{OPT}$. If t_j is not assigned to any participant in \mathbf{x}_{OPT} , then $\{p_0, t_j\} \in S_{OPT}$. If p_i has no target assigned to it in \mathbf{x}_{OPT} , then $\{p_i, t_0\} \in S_{OPT}$. If all targets have been assigned and each participant has at least one target assigned to it in \mathbf{x}_{OPT} , then $\{p_0, t_0\} \in S_{OPT}$.

We show by contradiction that S_{OPT} covers k times the set W with the minimal cost, i.e., any other Pareto-optimal solution would not have lower cost. Let us assume by contradiction that there is S' that covers W k times with lower cost. Elements of S' can be interpreted as assignments of participants to targets, therefore they define a solution \mathbf{x}' of the GPTA problem. Note that all targets assigned to p_0 in S' are unassigned in \mathbf{x}' . Such solution has lower task cost than \mathbf{x}_{OPT} and its task coverage can be:

- equal to the task coverage of \mathbf{x}_{OPT} . Then, the solution \mathbf{x}_{OPT} is not the minimal task cost solution, which is a contradiction.
- greater than the task coverage of \mathbf{x}_{OPT} . Then, \mathbf{x}_{OPT} could be improved (task coverage increased and task

cost decreased) and therefore is not a Pareto-optimal solution, which is a contradiction.

- less than the task coverage of \mathbf{x}_{OPT} . Let q_{OPT} be a coverage of the solution \mathbf{x}_{OPT} , q' be a coverage of the solution \mathbf{x}' , and $k = \max_{j \in M} k_j$ be the maximum number of participants requested to cover a single target. Then, $q' \leq q_{OPT} - 1/k$ and from the definition of D we have $c(S_{OPT}) < (m - q_{OPT})D + D$ and $c(S') \geq (m - q')D$. Therefore, $c(S_{OPT}) < (m - q_{OPT} + 1)D \leq (m - q')D - \frac{k-1}{k}D \leq c(S')$ and $c(S_{OPT}) < c(S')$, which contradicts our assumption that S' has the cost lower than S_{OPT} .

Thus, S_{OPT} is the minimal cost coverage of W , which completes the proof. \square

THEOREM 3.2. *The LPTA is an NP-hard optimization problem.*

PROOF. To prove that LPTA is NP-hard we show a polynomial reduction of the GPTA to the LPTA.

Let $P = \{p_1, \dots, p_n\}$ be a set of participants, $T = \{t_1, \dots, t_m\}$ be a set of targets, and Y be a set of all possible assignments, i.e., $Y = P \times T$. Let $d : Y \rightarrow \mathbb{R}_+$ be a distance between t_j and p_i , such that $d_{i,j} = \hat{d}_{i,j}$. We use such GPTA to define an LPTA with the initial solution $Y_0 = \emptyset$. Note that for such Y_0 any final solution will not have any reassigned targets and value of the objective function related to reassigned targets will be always equal to zero.

The set of all Pareto-optimal solutions \mathcal{Y}_{OPT} of the LPTA is also a set of solutions of the GPTA, i.e., $\mathcal{X}_{OPT} = \mathcal{Y}_{OPT}$. We show by contradiction that \mathcal{X}_{OPT} is also a set of all Pareto-optimal solutions of the GPTA. Let us assume by contradiction that \mathcal{X}' are all Pareto-optimal solutions of GPTA and $\mathcal{X}' \neq \mathcal{X}_{OPT}$. Thus, $\mathcal{Y}' = \mathcal{X}'$ are all Pareto-optimal solutions of the LPTA and $\mathcal{Y}' \neq \mathcal{Y}_{OPT}$. Note that in the LPTA the objective function related to reassigning targets is constant and equals to 0, and all possible solutions have the same value of this objective function. Therefore, \mathcal{Y}_{OPT} is the set of all Pareto-optimal solutions of the LPTA and $\mathcal{Y}' = \mathcal{Y}_{OPT}$, which is a contradiction that completes the proof. \square

4. PRIVACY-AWARE TASK ASSIGNMENT ALGORITHMS

In this section, we propose efficient greedy algorithms to approximate the optimization objectives for both GPTA and LPTA.

4.1 First Stage: GPTA

In this section, 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 GPTA.

4.1.1 Distance Estimation

As mentioned earlier, we use a distance-based cost model in our work which defines the cost of sensing 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. Knowing the cloaked areas (as the pair of the area and the probabilistic density function $\langle a, f \rangle$), we propose two methods to calculate the expected distances.

i) *Centroid-point*: In this 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 $\hat{\mathbf{d}}$.

$$\hat{d}_{i,j} = \text{dist} \left(\int_{z \in a} z f(z) dz, l_j \right)$$

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 the target j is in the sensing range of the participant i). To calculate this probability, we apply a simple pruning approach for each participant-target pair which shrinks the cloaked area a to a' which is the intersection area of a circle centered in the target j with the radius of r_i (i.e., the sensing range of the participant i) and the cloaked area. Then, having the probability density function f , we calculate the probability of the participant being in a' which is equal to the probability of the target j being in the sensing range of participant i ($\rho_{i,j}$).

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

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

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

Different cloaking methods can effect distance estimation. However, without loss of generality, we assume each participant's location is cloaked in a circular region with uniform probability distribution. Figure 2 illustrates the estimation approaches.

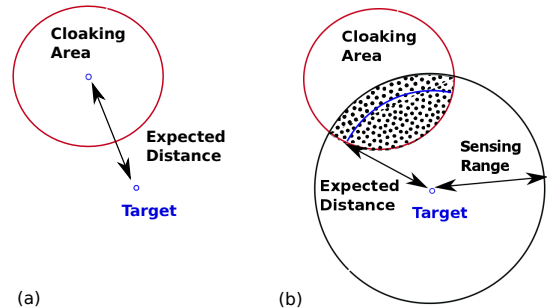


Figure 2: (a) Centroid-point method, (b) Expected-probabilistic method.

4.1.2 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 and removes the covered portion of the targets it covers from the remaining required coverage, until either all targets are covered or all energy budgets of participants are exhausted. The algorithm stops when no more update is possible. Since both the number of targets to be assigned and all energy budgets do not increase in time and have always non-negative values, the number of updates is finite. Hence, our algorithm eventually terminates. In each iteration, the algorithm finds the most cost-effective pair of participant-target 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 $\phi_{i,j}^{(1)}$.

$$\phi_{i,j}^{(1)} = \frac{\hat{d}_{i,j}}{\min(1 - u_j^+, u_j) + \epsilon}$$

which is the fraction of expected distance $\hat{d}_{i,j}$ to the coverage introduced by this participant. \mathbf{u}^+ is the vector of already 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. Finding the minimum in the denominator aims at preventing over-coverage. The small positive value ϵ is added to avoid overflow when the offered coverage by the participant is zero. ϵ should be selected smaller than the minimum value of vector \mathbf{u} . 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 the probability $\rho_{i,j}$. For Centroid-point method, these probabilities are calculated as

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

while for the probabilistic method, $\rho_{i,j}$ is calculated as described in section 4.1.1. At the end of the first stage, the covered portions of targets is calculated 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 along with her first stage assignment and the set of her accessible targets.

4.2 Second Stage: LPTA

The main pitfall in the second stage is over-coverage, i.e. assigning more participants to targets than required. To avoid over-coverage, the server provides the expected coverage vector, the final \mathbf{u}^+ at the end of GPTA to all participants. Hence, the \mathbf{u}^+ at the beginning of second stage optimization is initialized with the given values from the server. Algorithm 2 represents 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 its exact location, and the information provided by the server, the set of the nearest targets τ , 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 second line of the algorithm 2 recalculates the \mathbf{u}^+ to access the covered portions of the targets without consideration of this participant.

Similar to Algorithm 1, the second stage algorithm itera-

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

Data: P (set of participants), T (set of targets), \mathbf{b} (vector of sensing energy budgets), \hat{d} (matrix of expected distances), \mathbf{u} vector of coverage values, ρ (matrix of the access probabilities)

Result: \mathbf{x} (matrix of task assignments), \mathbf{u}^+ (vector of covered portion of targets)

```

1 All elements of  $\mathbf{x}$  and  $\mathbf{u}^+$  is initialized to 0;
2 while All targets are not covered do
3   Select the most cost-effective target-participant pair
   from the remaining pairs, say indexed at  $i$  and  $j$  with
   the probability  $\rho_{i,j}$ .
4   if A remaining pair exists then
5     Assign the selected target to participant as
6      $x_{i,j} \leftarrow 1$ 
7      $u_j^+ \leftarrow \min(1 - u_j^+, u_j) + u_j^+$ 
8      $b_i \leftarrow b_i - \hat{d}_{i,j}$ 
9     if  $u_j^+ = 1$  then
10    | Mark  $t_j$  as covered;
11    end
12  else
13  | Break
14  end

```

tively picks the most cost-effective target and assigns it to p_i with some probability. One difference is, since we want to minimize the difference between the assignments of two stages for participants (as stated in the third line of (2)), we penalize each new assignment which is different from $x_{i,j}$. Therefore, the cost-effectiveness of each assignment in this stage is calculated as $\phi_{i,j}^{(2)}$.

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

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

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

Using this probability, we aim at avoiding over-coverage of the targets, but at the same time giving smaller chance to costly assignments. Without this probability, participants would keep assigning targets until their energy budget is exhausted. Completely expending the energy budget by all participants can result in over-coverage. This effect can be seen easily in the baseline method which is compared to our methods in the experiments section. Using $\phi^{(2)}$ in calculating this probability emphasizes the importance of a cost-effective selection by giving it a higher probability.

5. EXPERIMENTAL RESULTS

In this section, we evaluate our task assignment methods experimentally to show their efficiency and effectiveness. First

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

Data:

p_i , $i \in N$ (the participant), τ (set of nearest targets for p_i), \mathbf{x}_i (first stage assignments for p_i), r_i (p_i 's sensing range), b_i (p_i 's energy budget), \mathbf{u} (vector of coverage), \mathbf{u}^+ (vector of covered portions of targets)

Result: \mathbf{y}_i (vector of task assignments for p_i)

```

1 All elements of  $\mathbf{y}_i$  are initialized to 0;
2 for all the targets in  $\tau$  do
3   |  $u_j^+ \leftarrow u_j^+ - u_j \times x_{i,j}$ 
4 end
5 for all the possible targets in  $\tau$  sorted based on
  cost-effectiveness do
6   | Select the target  $j$  with the probability  $\rho_j$ 
7   | Assign the selected target to  $p_i$  as  $y_{i,j} \leftarrow 1$ 
8   |  $u_j^+ \leftarrow u_j^+ + \min(1 - u_j^+, u_j)$ 
9   |  $b_i \leftarrow b_i - d_{i,j}$ 
10  | if energy budget of the participant is exhausted then
11  |   | Break
12  | end
13 end

```

we discuss the details of our experiment settings, then we analyze the results.

5.1 Settings

We use Brinkhoff's Network-based Generator of Moving Objects [2] to create a set of moving objects in all of our experiments. The map of the city of Oldenburg in Germany is used as the input to the generator. In each time snapshot, the set of participants is chosen uniformly from the set of generated moving objects in the map. In the same way, targets are selected from the nodes of the road graph of the map. The rest of the parameters are randomly simulated for participants and targets. In our experiments, we study the effect of different parameters such as the number of participants/targets, crowd density, and cloaking size on the task cost and coverage. We also analyze the performance of all methods by studying their running time. Task cost (TC) and coverage (TU) are calculated as described in section 3. To combine these two values into one evaluation criterion we add the task cost (TC) and uncovered portion of the task ($m - TC$) and normalize the sum to the range of [0,1] using min-max method. We refer to this normalized value as combined cost (CC):

$$CC = \frac{(m - TU) + TC}{m + \sum_{i \in N} \frac{b_i}{r_i}}$$

where the number of targets m represents the maximum possible value for task coverage. 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. The smaller value of CC represents higher coverage and lower cost which is a better result. A weighted version of this evaluation criterion can also be defined to compare the results when high coverage or low cost is favored.

$$WCC = \frac{(1 - w)(m - TU) + w \times TC}{m + \sum_{i \in N} \frac{b_i}{r_i}}$$

where w is a real number in the range of [0,1].

Table 2 shows default settings of our simulations. In all experiments, we select the sensing range of the participants randomly in the range of [100-300]. Sensing energy of the participants is always selected as a multiple of the sensing range with a default value of one. For all participants, we assume circular cloaking areas with uniform probability distribution. The radius of the cloaking area is selected randomly in the range of [500-1500] to match the range [1-10] percentage of the map roughly. For all of our experiments, we just use a part of the map of Oldenburg which is specified in figure 3. The required coverage of the targets is selected randomly as an integer number between 1 and 5 with the default value of one. Our proposed meth-

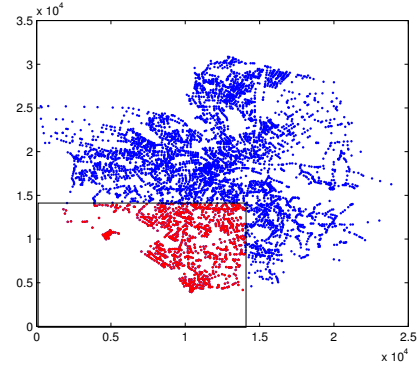


Figure 3: The map of Oldenburg, Germany generated by [2]. The employed section is framed.

ods are presented in four groups based on the distance estimation model of the GPTA (Centroid-point or Expected-point) and the optimization stages (one-stage GPTA-only or two-stage GPTA-LPTA combination). We refer to our approaches as CPA1, CPA2, EPA1, EPA2 to represent the one/two-stage Centroid-point, and one/two-stage Expected-probabilistic approaches respectively. We use two approaches to compare with our methods. We utilize our first stage optimization solution with zero level of privacy as an approach with no privacy constraint (NPA). In this approach, we assume the tasking server has access to exact locations of the participants, therefore it runs only in the server. As another approach we use an autonomous task selection method in which participants select targets from a global set of tasks locally while no globally optimized assignment is provided. We refer to this method as our baseline approach (BSA) which runs only on clients.

5.2 Results

We first examine the results of all methods for different densities of participants and targets to compare our proposed methods with each other and the other two methods BSA and NPA. Then, we compare our superior methods (two-stage methods) with BSA and NPA extensively to study the impact of crowd density, cloaking sizes, and evaluation weight on their combined cost and running time.

5.2.1 Impact of Numbers of Participants and Targets

Table 2: Default Settings

Parameter	Value
Number of Participants	200 - 1000 (Default Value 1000)
Number of Targets	100 - 500 (Default Value 100)
Sensing range	100 - 300
Sensing energy	[1-5] times sensing range (Default value 1)
Required Coverage of Targets	[1-5] (Default value 1)
Cloaking Size (Radius)	500 - 1500

In these experiments, we study the impact of increasing the number of participants and targets on combined cost and running time by: (a) varying the number of participants while the number of targets is fixed; (b) varying the number of targets while the number of participants is fixed; and (c) varying both numbers of participants and targets simultaneously while maintaining their ratio.

Figure 4 shows the combined cost for increasing number of participants with a fixed number of targets using the default settings. Increasing the number of participants increases the task coverage resulting in lower combined costs for all of the approaches. For all combinations of the participants and targets, the expected-probabilistic approaches perform better than their corresponding centroid-point approaches. The difference is more significant for CPA2 and EPA2 for larger number of participants. On the other hand, regardless of the distance estimation methods, both two-stage methods outperform the one-stage methods. EPA2 improves the results of EPA1 up to 1.4 times, while the CPA2 improves the results of CPA1 up to 1.2 times. The baseline approach (BSA) outperforms our one-stage methods for smaller participant/target ratios. However, as the number of participants and subsequently the participant/target ratio increases, both CPA1 and EPA1 outperform BSA significantly. As for our two-stage methods, both CPA2 and EPA2 outperform the baseline approach (BSA). Comparing EPA2 and BSA, the difference of the results increases as the the number of participants increases reaching up to 2.4 times lower combined cost for 1000 participants (i.e., 10 to 1 participant/target ratio). Our approach achieves comparable cost to the NPA approach which is without privacy, compared to the baseline approach BSA.

Figure 5 shows the running time for the same experiment. EPA1 and EPA2 which benefit from the probabilistic solution, include a running time overhead for calculating accessible probabilities and expected distances. This time overhead makes expected-probabilistic approaches more costly than centroid-point methods regardless of the number of stages. Moreover, the effect of the second stage on running time is trivial for both methods, therefore we have presented them together. Intuitively, since the second stage runs on clients in a distributed setting with a small number of targets, its running time is very small compared to the global optimization in the first stage. The same reasoning explains the small running time of the BSA approach. Increasing the number of participants has a linear impact on the running time of all methods.

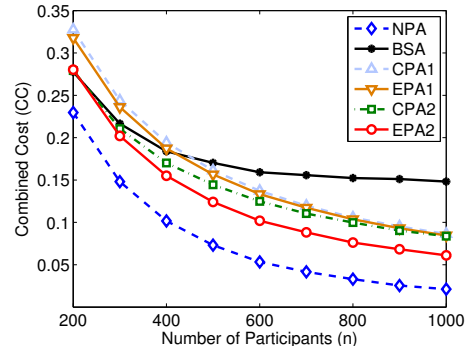
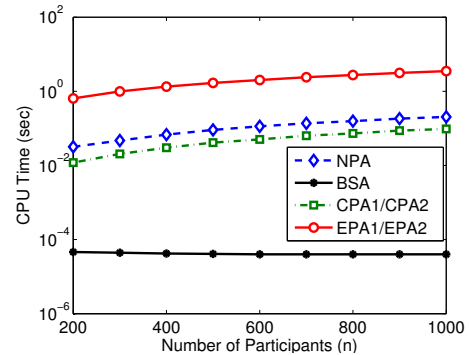
**Figure 4: Combined cost for different number of participants, and $m = 100$** **Figure 5: Running time for different number of participants, and $m = 100$**

Figure 6 shows the combined cost for different number of targets and a fixed number of participants using the default settings. Our one-stage approaches outperform the BSA for most of the values of m , performing similar to it for larger number of targets (i.e., smaller participant/target ratios). For all numbers of targets, the EPA2 outperforms the CPA2 and BSA. The main difference between the figures 4 and 6 is how the combined cost decreases significantly by increasing the number of participants in figure 4 while it increases when adding more targets to the task in figure 6. The reason is, by increasing the number of targets with constant number of participants, the uncovered portion of the targets increases, resulting in higher combined cost. Figure 7 shows the running time for the same experiment. While the effect of the number of participants was linear on the running time, increasing the number of targets has a polynomial effect.

Since we observe that the two-stage approaches (EPA2 and CPA2) both outperform their corresponding one-stage approaches (EPA1 and CPA1), we will only show EPA2 and CPA2 for the remaining experiment results for better readability of the graphs.

Figure 8 shows the combined cost for increasing numbers of both participants and targets with a fixed ratio of two to one using the other default settings. For all numbers of participants and targets, our two-stage methods (CPA2 and EPA2) outperform the baseline approach (BSA). The difference becomes more significant as the number of participants

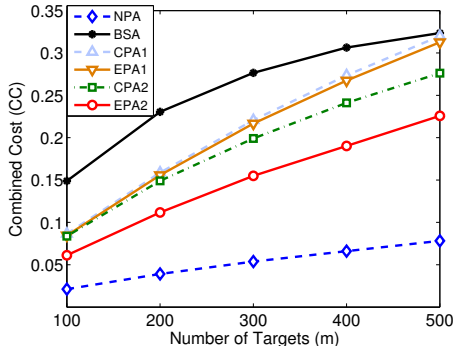


Figure 6: Combined cost for different number of targets, and $n = 1000$

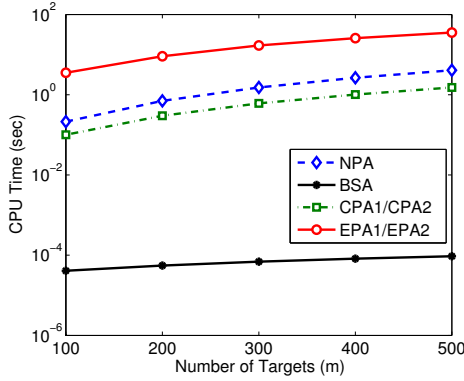


Figure 7: Running time for different number of targets, and $n = 1000$

and targets increases. EPA2 outperforms BSA by 1.4 times lower combined cost for 1000 participants and 500 targets. Figure 9 shows the running time for the same experiment.

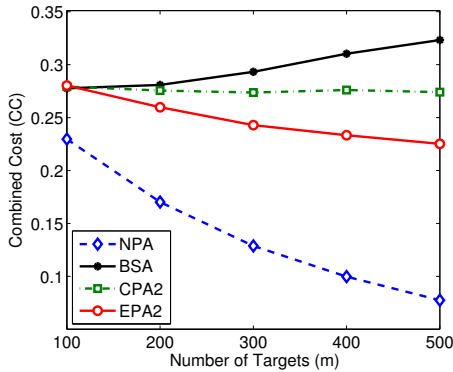


Figure 8: Combined cost for different number of targets and participants, and $m/n = 0.5$

5.2.2 Impact of Cloaking Size

Figure 10 shows the impact of cloaking size on combined cost for the fixed number of participants and targets using the default settings. The cloaking size is shown as the relative percentage of the map. By increasing the cloaked size, EPA2 shows more robustness compared to CPA2. BSA and NPA

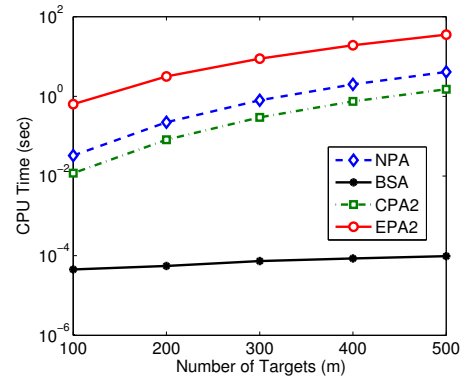


Figure 9: Running time for different number of targets and participants, and $m/n = 0.5$

are not affected by the cloaking size. Figure 11 shows the impact of cloaking size on running time for the same experiment. Unlike the combined cost, the running time of EPA2 is more affected compared to CPA2 due to the overhead of processing larger cloaked areas.

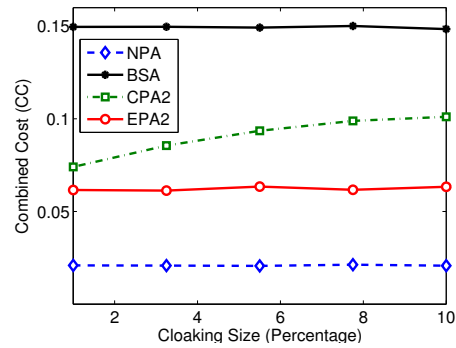


Figure 10: Combined cost for different cloaking size, $m = 100$, and $n = 1000$

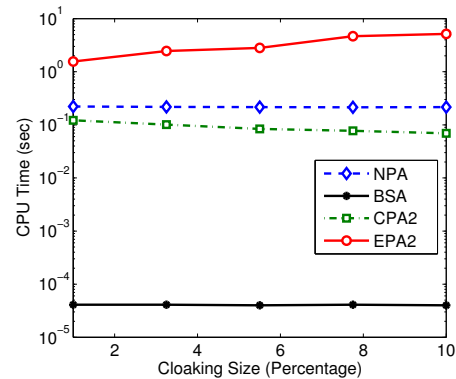


Figure 11: Running time for different cloaking size, $m = 100$, and $n = 1000$

5.2.3 Weighted Evaluation Criteria

Figure 12 presents how the combined costs of all methods are affected by different weights for evaluating task coverage and

cost. Both our methods show more robustness compared to BSA as w varies. For smaller values of w , i.e. more weight on coverage, BSA outperforms our methods. However, both CPA2 and EPA2 outperform BSA for most values of w .

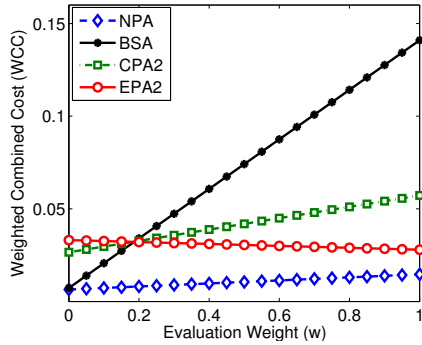


Figure 12: Weighted combined cost for different weights, $m = 100$, and $n = 1000$

6. CONCLUSIONS AND FUTURE WORK

In this paper we defined and formulated the problem of privacy-preserving spatial task assignment in coordinated Participatory Sensing (PS) as a novel two-stage optimization problem which maximizes coverage and minimizes cost for the tasks. We proved the problem to be NP-hard in each stage, therefore we proposed efficient greedy algorithms to approximate each stage of the optimization problem. We studied the impact of all the parameters affecting our methods and showed their efficiency and robustness. As our next step, we plan to evaluate our methods with real data by implementing a real-world PS application which uses our task assignment approach.

7. ACKNOWLEDGMENTS

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

8. REFERENCES

- [1] 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*, pages 47–60. Springer, 2007.
- [2] T. Brinkhoff. A framework for generating network-based moving objects. *GeoInformatica*, 6(2):153–180, 2002.
- [3] 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*, pages 171–178. ACM, 2006.
- [4] D. Christin, A. Reinhardt, S. S. Kanhere, and M. Hollick. A survey on privacy in mobile participatory sensing applications. *Journal of Systems and Software*, 84(11):1928–1946, 2011.
- [5] C. Cornelius, D. K. A. Kapadia, D. Peebles, M. Shin, and N. Triandopoulos. Anonymsense: Privacy-aware people-centric sensing. volume 5, pages 211–224. In *Proceedings of the International Conference on Mobile Systems, Applications and Services (MobiSys)*, ACM Press, Jun 2008.
- [6] T. Das, P. Mohan, V. N. Padmanabhan, R. Ramjee, and A. Sharma. Prism: Platform for remote sensing using smartphones. pages 63–76. in *Proceedings of the 8th ACM International Conference on Mobile Systems, Applications, and Services (MobiSys)*, 2010.
- [7] M. Duckham and L. Kulik. A formal model of obfuscation and negotiation for location privacy. In *Pervasive Computing*, pages 152–170. Springer, 2005.
- [8] B. Gedik and L. Liu. A customizable k-anonymity model for protecting location privacy. 2004.
- [9] 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.
- [10] 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*, 19(12):1719–1733, 2007.
- [11] A. Kansal, M. Goraczko, and F. Zhao. Building a sensor network of mobile phones. pages 547–548. in *Proceedings of the 6th International Conference on Information Processing in Sensor Networks (IPSN)*, 2007.
- [12] A. Khoshgozaran and C. Shahabi. Blind evaluation of nearest neighbor queries using space transformation to preserve location privacy. In *SSTD*, 2007.
- [13] 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.
- [14] N. D. Lane, S. B. Eisenman, M. Musolesi, E. Miluzzo, and A. T. Campbell. Urban sensing systems: opportunistic or participatory? pages 11–16. In *Proceedings of the Workshop on Mobile Computing Systems and Applications (HotMobile)*, ACM Press, 2008.
- [15] M. F. Mokbel. Towards privacy-aware location-based database servers. In *Data Engineering Workshops, 2006. Proceedings. 22nd International Conference on*, pages 93–93. IEEE, 2006.
- [16] 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.
- [17] S. Reddy, D. Estrin, and M. Srivastava. Recruitment framework for participatory sensing data collections. pages 138–155. in *Proceedings of the 8th International Conference on Pervasive Computing*, Springer Berlin Heidelberg, May 2010.
- [18] M. Riahi, T. G. Papaioannou, I. Trummer, and K. Aberer. Utility-driven data acquisition in participatory sensing. *EDBT/ICDT*, ACM, March 2013.
- [19] M. Shin, C. Cornelius, D. Peebles, A. Kapadia, D. Kotz, and N. Triandopoulos. Anonymsense: A system for anonymous opportunistic sensing. *Journal of Pervasive and Mobile Computing*, 7(1):16–30, 2010.
- [20] 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*, pages 62–71. ACM, 2009.
- [21] V. V. Vazirani. *Approximation algorithms*. 2001.