Accuracy-aware Aquatic Diffusion Process Profiling using Mobile Sensor Networks

Yu Wang*, Rui Tan*, Guoliang Xing*, Xiaobo Tan†
* Department of Computer Science and Engineering, Michigan State University, USA
† Department of Electrical and Computer Engineering, Michigan State University, USA

Abstract—Water resources and aquatic ecosystems are facing increasing threats from climate change, improper waste disposal, and oil spill incidents. It is of great interest to deploy mobile sensor networks to detect and monitor the (e.g., chemical pollutants or oil) diffusion processes that are harmful to aquatic environments. In this paper, we propose an accuracy-aware diffusion process profiling approach using smart aquatic mobile sensors such as robotic fish. In our approach, the mobile sensors collaboratively profile the characteristics of a diffusion including source location, pollution amount, and its evolution over time. In particular, the mobile sensors reposition themselves to progressively improve the profiling accuracy. We formally formulate a novel movement scheduling problem that aims to maximize the profiling accuracy subject to limited sensor mobility and energy budget. We develop a light-weight greedy algorithm as well as a more complex near-optimal algorithm to solve the problem. Extensive simulations under realistic settings validate the effectiveness of our approach. Moreover, a preliminary implementation on TelosB motes shows the feasibility of deploying our movement scheduling algorithms on mote-class sensor platforms.

I. INTRODUCTION

Water resources and aquatic ecosystems have been facing various physical, chemical, and biological threats from climate change, industrial pollution, and improper waste disposal. For instance, the last four decades witnessed more than a dozen of major oil spill incidents with each releasing more than 30 million US gallons of oil [1]. Other harmful diffusion processes like chemical or radiation leaks could also have disastrous impact on public health, ecosystem sustainability, and marine biology. When such a crisis arises, an immediate requirement is to profile the characteristics of the diffusion process, including the location of source, the amount of released substance, and how rapidly it spreads in space and evolves over time.

Manual sampling, via boat/ship or with handheld devices, is still a common practice in the monitoring of aquatic diffusion processes. This approach is labor-intensive and difficult to adapt to the dynamic evolution of diffusion. An alternative is in situ sensing with fixed or buoyed/moored sensors [2]. However, since buoyed sensors cannot move around, it could take a prohibitively large number of them to capture spatially inhomogeneous information. The past couple of decades have seen significant progress in developing robotic technologies for aquatic sensing. Autonomous underwater vehicles (AUVs) [3] and sea gliders [4] are notable examples of such technologies. However, because of their high cost (over 50,000 US dollars per unit [5]), weight (over 100 pounds), and size (1-2 meters long), it is difficult to deploy many AUVs or sea gliders for temporally and spatially resolved measurement of diffusion processes.

Recent advances in computing, communication, and smart materials have made it possible to create untethered robotic fish with onboard power, control, navigation, wireless communication, and sensing modules, which turns these robots into mobile sensing platforms in aquatic environments. Fig. 1(a) shows a prototype of robotic fish swimming in an inland lake. Fig. 1(b) shows the close-up of a robotic fish prototype, equipped with a dissolved oxygen (DO) sensor, GPS, and other electronic components, which has been developed for monitoring the DO level in water. Due to the low manufacturing cost, these platforms can be massively deployed to form a mobile sensor network that monitors harmful diffusion processes, providing significantly higher spatial and temporal sensing resolution. Moreover, a school of robotic fish can coordinate their sensing and movement to adapt to the dynamics of evolving diffusion processes.

Despite the aforementioned advantages, low-cost mobile sensing platforms introduce several challenges for aquatic sensing. First, due to the constraints on size and energy, they are typically equipped with low-end sensors whose measurements are subject to significant biases and noises. They must efficiently collaborate in data processing to achieve satisfactory accuracy in profiling diffusion processes. Second, practical aquatic mobile sensors are only capable of slow-speed movement. For instance, the typical speed of robotic fish is 1.8 to 6 meters per minute [6]. Therefore, the movement of sensors must be efficiently scheduled to achieve real-time profiling of the diffusion processes that may evolve rapidly over time. Third, due to the high power consumption of locomotion, the distance that mobile sensors move in a profiling process should be minimized to extend the network lifetime.

We make the following major contributions in this paper:

- We propose a novel accuracy-aware approach for aquatic diffusion profiling based on mobile sensor net-
Our approach features an iterative strategy for estimating the spatiotemporally evolving diffusion process. In each iteration, the sensors first collaboratively profile the diffusion based on the maximum-likelihood estimation (MLE) and then reposition themselves to maximize the expected profiling accuracy in the next iteration.

- We derive the analytical profiling accuracy based on the Cramér-Rao bound (CRB) and then formulate a movement scheduling problem for aquatic diffusion profiling. The system objective is to maximize the profiling accuracy, subject to limited sensor mobility, energy budget and a delay constraint of the profiling process. We develop novel greedy and dynamic-programming-based movement scheduling algorithms to solve the formulated problem.

- We conduct extensive simulations under realistic settings to evaluate the profiling accuracy of our approach. Moreover, we also implement our movement scheduling algorithms on TelosB mote [7] and evaluate the system overhead. The results show that our approach can accurately profile the diffusion and adapt to its evolution. The greedy algorithm yields satisfactory performance with linear complexity and the dynamic-programming-based algorithm can find the near-optimal movement schedule with a higher but still polynomial complexity.

The rest of this paper is organized as follows. Section II reviews related work. Section III introduces the preliminaries and Section IV presents the overview of our approach. Section V derives the analytical profiling accuracy and Section VI formulates the movement scheduling problem. Section VII presents the two movement scheduling algorithms. Section VIII presents evaluation results and Section IX concludes this paper.

II. RELATED WORK

Previous work on diffusion process localization and profiling using sensor networks has adopted several different estimation techniques, which include state space, statistical signal processing and geometric trilateration. The state space approaches [8] [9] use discrete state-space equations to approximate the partial differential equations that govern the diffusion process, and then apply filtering algorithms such as Kalman filters [8] [9] to localize and profile the diffusion process based on noisy measurements. In the statistical signal processing approaches, several estimation techniques such as MLE [10] [11] and Bayesian parameter estimation [12] are applied to deal with noisy measurements. For instance, in [10], an MLE-based diffusion characterization algorithm is designed based on binary sensor measurements to reduce the communication cost. In [12], the parametric probability distribution of the diffusion profile parameters is passed among sensors and updated with sensor measurements by a Bayesian estimation algorithm. In particular, the passing route is determined according to various estimation performance metrics including CRB. In geometric trilateration approaches [13] [14], the measurement of a sensor is mapped to the distance from the sensor to the diffusion source. The source location can then be estimated by trilateration among multiple sensors. Such approaches incur low computational complexities, but suffer lower estimation accuracy compared with more advanced approaches such as MLE [14]. Different from the above studies that are focused on static sensor networks, we aim to exploit sensor mobility to improve the accuracy in profiling diffusion processes.

Recently, sensor mobility has been exploited to enhance the adaptability and sensing capability of sensor networks. For instance, heuristic movement scheduling algorithms are proposed in [15] to estimate the contours of a physical field. In [16], path planning schemes are proposed for mobile sensors to reconstruct a spatial map of environmental factors of interest. Our previous works exploit sensor mobility to improve the detection performance of a sensor network [17] [18]. Several studies are focused on using robots to improve the profiling accuracy of diffusion process. As an extension to [11], the gradient of CRB is used to schedule the movement of a single sensor. Similarly, a robot motion control algorithm is proposed in [19] to maximize the determinant of the Fisher matrix. However, these studies are only applicable to a small number (e.g., 3 in [19]) of powerful robots. In contrast, this paper focuses on developing movement scheduling algorithms for moderate- or large-scale sensor networks that are composed of inexpensive mobile sensors.

III. PRELIMINARIES

In this section, we describe the preliminaries of this paper, which include the diffusion process and sensor models.

A. Diffusion Process Model

The diffusion process, by which molecules spread from areas of high concentration to areas of low concentration, follows Fick’s law [20]. By denoting $t$ as the time elapsed from the discharge of substance and $c$ as the substance concentration at a particular position and time $t$, Fick’s law can be described by

$$\frac{\partial c}{\partial t} = D_x \cdot \frac{\partial^2 c}{\partial x^2} + D_y \cdot \frac{\partial^2 c}{\partial y^2} + D_z \cdot \frac{\partial^2 c}{\partial z^2},$$

where $D_x$, $D_y$ and $D_z$ are the diffusion coefficients in the directions of $x$-, $y$-, and $z$-axis, respectively. The diffusion coefficients characterize the speed of diffusion and depend on the species of diffuser and solvent as well as other environmental factors such as temperature. The above Fickian diffusion model has been widely adopted to study the spreading of gaseous substances [10] [11] and buoyant fluid pollutants such as oil slick on the sea [21] [22]. For many buoyant fluid pollutants, the two horizontal diffusion coefficients, i.e., $D_x$ and $D_y$, are identical, while the vertical diffusion coefficient, i.e., $D_z$, is insignificant. For instance, in a field experiment [23] where diesel oil was discharged into sea, the estimated $D_x$ is $2,000 \text{ cm}^2/\text{s}$ while $D_z$ is only $10 \text{ cm}^2/\text{s}$. Therefore,
the vertical diffusion coefficient can be safely ignored and the diffusion can be simplified as a 2-dimensional process. In the rest of this paper, we denote the horizontal diffusion coefficient as $D$, i.e., $D = D_x = D_y$. Suppose a total of $A$ cm$^3$ of substance is discharged at location $(x_0, y_0)$. Denote $d$ as the distance from any location $(x, y)$ to the source location $(x_0, y_0)$, i.e., $d = \sqrt{(x - x_0)^2 + (y - y_0)^2}$. As $D_y = D_y$, the diffusion is isotropic and the concentration can be denoted as $c(d, t)$. The initial condition for Eq. (1) is an impulse source, which can be represented by the Dirac delta function, i.e., $c(d, 0) = A \cdot \delta(d)$. With this initial condition, the closed-form solution of Eq. (1) is given as follows [10]:

$$c(d, t) = \alpha \cdot \exp\left(-\beta \cdot d^2\right), \quad d \geq 0, t > 0, \tag{2}$$

where $\alpha = \frac{A}{4\pi D t}$ and $\beta = \frac{1}{4 \pi D t}$. From Eq. (2), for a fixed time instance, the concentration distribution is described by the Gaussian function that centers at the source location. As the time elapses, the concentration distribution becomes flatter. In this paper, we represent the diffusion profile as $\Theta = \{x_0, y_0, \alpha, \beta\}$.

B. Sensor Model

Our approach leverages a cluster of aquatic mobile nodes (e.g., robotic fish [6]) to collaboratively profile the aquatic diffusion process. Many aquatic mobile platforms are battery-powered and hence have limited mobility and energy budget. For instance, the maximum movement speed of the robotic fish designed in [6] is 6 m/min. We assume that the mobile nodes are equipped with pollutant concentration sensors. Commercial off-the-shelf sensors are available to measure the concentration of various pollutants, e.g., the crude oil sensor Cyclops-7 [24]. The measurements of most sensors are subject to biases and additive random noises from sensor circuit and environment. Specifically, the reading of sensor $i$, denoted by $r_i$, is given by $r_i = c(d_i, t) + b_i + n_i$, where $d_i$ is the distance from sensor $i$ to the diffusion source, $b_i$ and $n_i$ are the bias and random noise experienced by sensor $i$, respectively. The bias $b_i$ represents the sensor’s response to foreign substances. We assume that the noise experienced by sensor $i$ follows the zero-mean normal distribution, i.e., $n_i \sim \mathcal{N}(0, \sigma^2)$, where $\sigma^2$ is the variance of $n_i$. We assume that the noises, i.e., $\{n_i|\forall i\}$, are independent across sensors. The bias and noise variance are often given in the specification from the manufacturer of the sensor. They may also be measured in offline lab experiments. For instance, by placing the sensor in the solvent (e.g., sea water) without pollutant, the bias and noise variance can be estimated by the sample mean and variance over a number of readings. The above measurement model has been widely adopted for various chemical sensors [10]–[12].

In this paper, we adopt a temporal sampling scheme to mitigate the impact of noise. Specifically, when sensor $i$ measures the concentration, it continuously takes $K$ samples in a short time, and computes the average as its measurement.

Therefore, the measurement, denoted by $z_i$, follows the normal distribution, i.e., $z_i \sim \mathcal{N}(c(d_i, t) + b, \sigma^2)$, where $\sigma^2 = \sigma^2 / K$.

Table I summarizes the notation used in this paper.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>$D$</td>
<td>diffusion coefficient</td>
</tr>
<tr>
<td>$A$</td>
<td>total amount of discharged substance in cm$^3$</td>
</tr>
<tr>
<td>$t$</td>
<td>time from the discharge of substance</td>
</tr>
<tr>
<td>$\alpha, \beta$</td>
<td>diffusion process profile parameters</td>
</tr>
<tr>
<td>$(x_0, y_0)$</td>
<td>coordinates of the diffusion source</td>
</tr>
<tr>
<td>$(x_i + x_0, y_i + y_0)$</td>
<td>coordinates of sensor $i$</td>
</tr>
<tr>
<td>$d_i$</td>
<td>distance from the diffusion source, $d_i^2 = x_i^2 + y_i^2$</td>
</tr>
<tr>
<td>$c(d_i, t)$</td>
<td>concentration at the position of sensor $i$ and time $t$</td>
</tr>
<tr>
<td>$b_i, \sigma^2$</td>
<td>sensor bias and noise variance</td>
</tr>
<tr>
<td>$z_i$</td>
<td>sensor measurement, $z_i \sim \mathcal{N}(c(d_i, t) + b, \sigma^2)$</td>
</tr>
<tr>
<td>$K$</td>
<td>the number of samples for computing a measurement</td>
</tr>
<tr>
<td>$N$</td>
<td>total number of sensors</td>
</tr>
<tr>
<td>$\mathbf{z}$</td>
<td>normalized observation, $\mathbf{z} = [z_1, z_2, \ldots, z_N]^T$</td>
</tr>
<tr>
<td>$\omega$</td>
<td>diffusion process profiling accuracy metric</td>
</tr>
<tr>
<td>$m_i$</td>
<td>number of movement steps in a profiling iteration</td>
</tr>
<tr>
<td>$\phi_i$</td>
<td>movement orientation in a profiling iteration</td>
</tr>
<tr>
<td>$M$</td>
<td>number of allocatable steps in a profiling iteration</td>
</tr>
<tr>
<td>$\tau$</td>
<td>time delay of a profiling iteration</td>
</tr>
<tr>
<td>$\nu$</td>
<td>sensor movement speed</td>
</tr>
<tr>
<td>$L$</td>
<td>upper bound of moving distance in a profiling iteration</td>
</tr>
</tbody>
</table>

* The symbols with subscript $i$ refer to the notation of sensor $i$.

IV. OVERVIEW OF APPROACH

In this section, we provide an overview of our accuracy-aware approach to profiling aquatic diffusion process using a mobile sensor network. Our approach is designed to meet two key objectives. First, the noisy measurements of sensors are jointly processed to improve the accuracy of profiling the diffusion. Second, sensors can actively move according to existing measurements to maximize the profiling accuracy subject to energy consumption budget. With the estimated profile $\Theta = \{x_0, y_0, \alpha, \beta\}$, we can compute the contour of any given concentration at the current time instance. As long as the diffusion coefficient $D$ is known, we can also calculate the elapsed time and the total amount of discharged substance with $t = (4D)^{-1}$ and $A = \pi \sigma^2 \beta^{-1}$, respectively. Hence, we can predict the evolution of the diffusion in the future, which is often important for emergency management in the cases of harmful substance diffusion.

We assume that the mobile sensors are deployed at randomly chosen locations in the region of interest that contains the source of an active diffusion process. For instance, the sensors can be dropped off from an unmanned aerial vehicle or placed by an aquatic vessel at randomly chosen positions. We assume that each sensor knows its position (e.g., through a mounted GPS unit). After the deployment, sensors begin a diffusion profiling process consisting of multiple profiling iterations. The iterative profiling process is illustrated in Fig. 2. In a profiling iteration, sensors first synchronously take concentration measurements and send to the cluster head. The cluster head then adopts the MLE to estimate the diffusion profile $\Theta$ from the noisy concentration measurements of all sensors. With the estimated diffusion...
profile, the cluster head schedules the movements of sensors such that the expected profiling accuracy in the next profiling iteration is maximized, subject to the limited sensor mobility and energy budget. Finally, the movement schedule including moving distances and orientations is sent to sensors for directing their movements.

Our accuracy-aware diffusion profiling approach features the following key novelties. First, our approach starts with little prior knowledge about the diffusion and progressively learns the profile of the diffusion with improved accuracy along the profiling iterations. Such an iterative profiling strategy adapts to the dynamics of the diffusion process while reducing energy consumption of mobile sensors. Second, we analyze the CRB of the MLE-based diffusion profiling algorithm and propose a novel CRB-based profiling accuracy metric, which is used to direct the movement scheduling of mobile sensors. Third, we propose two novel movement scheduling algorithms, which include a gradient-descent-based greedy algorithm and a dynamic-programming-based near-optimal algorithm. The greedy algorithm only incurs linear complexity, while the dynamic-programming-based algorithm can find the near-optimal movement schedule with a higher but still polynomial complexity.

V. DIFFUSION PROFILING ACCURACY ANALYSIS

In this section, we will analyze the theoretical accuracy of the diffusion process profiling using distributed sensors. Specifically, we will derive the Cramér-Rao bound (CRB), which gives the theoretical accuracy bound of any estimation algorithm. We then propose a novel CRB-based profiling accuracy metric and its approximation, which will be used to guide the movements of sensors in Section VI.

A. Cramér-Rao Bound for Diffusion Process Profiling

CRB provides a theoretical lower bound on the accuracy of parameter estimators [25]. If an efficient estimation algorithm is employed, the estimation performance can be characterized by the CRB. CRB has been widely adopted to guide the design and evaluate the performance of estimation algorithms [11] [12]. In this section, we derive the CRB of estimating the diffusion profile $\Theta$.

We assume that $N$ aquatic sensors are deployed in the region of interest. In each profiling iteration, we first remove sensor biases and normalize the measurements to construct an observation vector, denoted by $z$, i.e., $z = [z_1 - b_1, \ldots, z_n - b_n]^T$. By denoting

$$H = [\sigma^{-1}e^{-\beta d_1^2}, \ldots, \sigma^{-1}e^{-\beta d_N^2}]^T,$$

$z$ follows the $N$-dimensional normal distribution, i.e., $z \sim N(\alpha H, I)$, where $I$ is an $N \times N$ unit matrix. The log-likelihood of $\Theta$ regarding an observation $z$, which can be derived as the logarithm of the probability density function of $z$, is given by [25]:

$$l(\Theta|z) = -(z - \alpha H)^T(z - \alpha H).$$  (3)

CRB is given by the inverse of the Fisher matrix [25]. For the diffusion profiling, the Fisher matrix is defined by $J = -\mathbb{E}_z \left( \frac{\partial^2}{\partial \Theta^2} l(\Theta|z) \right) = \alpha^2 \beta^2 H H^T$, where the expectation $\mathbb{E}_z[.]$ is taken over all possible observations $z$. For clarity of presentation, we denote $\hat{\Theta}$ as the estimate of $\Theta$ in the rest of this paper. The $k^{th}$ diagonal element of the inverse of $J$ (denoted by $J_{k,k}^{-1}$) provides a lower bound on the variance of the $k^{th}$ element of $\hat{\Theta}$ (denoted by $\hat{\Theta}_k$) [25]. Formally, $\text{Var}(\hat{\Theta}_k) \geq J_{k,k}^{-1}$. The $J_{k,k}$ is the CRB of $\Theta_k$, which is denoted as $\text{CRB}(\Theta_k)$ in this paper. Note that the CRBs of the parameters to be estimated, i.e., $x_0, y_0, \alpha, \beta$, can be easily computed via numerical methods. However, in order to guide the movements of sensors, we will derive the closed-form CRBs.

Even though $J$ is just a $4 \times 4$ matrix, deriving $J^{-1}$ is challenging. To simplify the discussion, we set up a Cartesian coordinate system with the origin at the diffusion source and let $(x_i, y_i)$ denote the coordinates of sensor $i$. Note that the coordinates of the diffusion source and sensor $i$ in the global coordinate system are $(x_0, y_0)$ and $(x_i + x_0, y_i + y_0)$, respectively. We first apply matrix calculus to derive the closed-form $J$ and then derive $J^{-1}$ by block matrix manipulations. Due to space limit, the details of the derivations are omitted here and can be found in [26]. To facilitate the expression of CRB, we first define several notations, i.e., $\tilde{x}_i$, $\tilde{y}_i$, $L_{X_1}$, $L_{X_2}$, $L_{Y_1}$, and $L_{Y_2}$. First, $\tilde{x}_i$ is given by

$$\tilde{x}_i = \frac{\sum_{j=1}^{N} x_j (d_j^2 - d_i^2) e^{-2\beta d_j^2}}{\sqrt{\sum_{m=1}^{N} \sum_{n=1}^{N} (d_m^2 - d_n^2)^2 e^{-2\beta (d_m^2 + d_n^2)}}}.$$  (4)

By replacing $x_j$ in Eq. (4) with $y_j$, we have $\tilde{y}_i$. Moreover, $L_{X_1}$, $L_{Y_1}$ are $1 \times N$ vectors, and $L_{X_2}$, $L_{Y_2}$ are $N \times 1$ vectors, where the $i^{th}$ elements of them are given by

$$L_{X_1}(i) = \sigma^{-1}e^{-\beta d_i^2}(x_i + \tilde{x}_i), \quad L_{Y_1}(i) = \sigma^{-1}e^{-\beta d_i^2}(y_i + \tilde{y}_i),$$

$$L_{X_2}(i) = \sigma^{-1}e^{-\beta d_i^2}(x_i - \tilde{x}_i), \quad L_{Y_2}(i) = \sigma^{-1}e^{-\beta d_i^2}(y_i - \tilde{y}_i).$$

With the above notations, the CRBs for $x_0$ and $y_0$ can be expressed by

$$\text{CRB}(x_0) = J_{1,1}^{-1} = \frac{(4\sigma^2 \beta^2)^{-1}}{L_{X_1} L_{X_2}},$$

$$\text{CRB}(y_0) = J_{2,2}^{-1} = \frac{(4\sigma^2 \beta^2)^{-1}}{L_{Y_1} L_{Y_2}}.$$  (5, 6)

Note that by applying the same matrix manipulations, we can also derive the CRBs of $\alpha$ and $\beta$. 

---

Figure 2: The iterative profiling.
B. Profiling Accuracy Metric

In this section, we propose a novel diffusion profiling accuracy metric based on the CRBs derived in Section V-A, which will be used to guide the movements of sensors in Section VI. As the parameters $x_0$, $y_0$, $\alpha$, and $\beta$ have different physical units, it is difficult to define an accuracy metric that jointly accounts for all the parameters. Several previous works [19] adopt the determinant of the Fisher matrix as the accuracy metric, which jointly considers all the parameters. Such a metric requires that the parameters are properly normalized to avoid biases. However, normalizing the parameters with different physical meanings is highly problem-dependent. Moreover, as the closed-form determinant of the Fisher matrix is extremely complicated, the resulted sensor movement scheduling has to rely on the numerical methods with high computational complexities [19]. In this paper, we propose a new profiling accuracy metric, denoted by $\omega$, which is defined according to the sum of reciprocals of CRB($x_0$) and CRB($y_0$). Formally,

$$\omega = \frac{1}{\text{CRB}(x_0)} + \frac{1}{\text{CRB}(y_0)} = (1-\epsilon)(L_{X_1} L_{X_2} + L_{Y_1} L_{Y_2}),$$

where $\epsilon = \frac{(L_{X_1} L_{Y_2} + L_{X_2} L_{Y_1})^2}{4 L_{X_1} L_{X_2} L_{Y_1} L_{Y_2}}$. By adopting reciprocals, the accuracy analysis can be largely simplified. Note that as $\alpha$ and $\beta$ are unknown but fixed in a particular profiling iteration, $4 \alpha^2 \beta^2$ in the denominator of Eq. (7) is a scaling factor. Therefore, optimizing $\frac{1}{\text{CRB}(x_0)} + \frac{1}{\text{CRB}(y_0)}$ is equivalent to optimizing $\omega$. In our approach, we adopt the MLE to estimate $\Theta$. As discussed in Section VI-A, the estimation accuracy of MLE converges to CRB and hence a larger $\omega$ indicates more accurate estimation of $x_0$ and $y_0$. With the metric $\omega$, the movements of sensors will be directed according to the accuracy of localizing the diffusion source. In the rest of this paper, profiling accuracy refers to the metric $\omega$ defined in Eq. (7). Note that our approach can also be applied to the CRBs of the elapsed time $t$ and substance amount $A$.

According to the derivations in Section V-A, $L_{X_1}$, $L_{X_2}$, $L_{Y_1}$, and $L_{Y_2}$ depend on $x_0$, $y_0$, $x_i$, $y_i$. Therefore, $\omega$ is a function of the positions of the sensors and the diffusion source. This observation has two important implications. First, as the profiling accuracy depends on the sensors’ positions, it is desirable to reposition sensors to achieve better profiling accuracy. Second, as the position of the diffusion source, i.e., $(x_0, y_0)$, is unknown to the network, it is impossible to compute the true profiling accuracy. In our approach, we compute $\omega$ based on the estimated position of the diffusion source, i.e., $(\hat{x}_0, \hat{y}_0)$. As sensors are repositioned in each profiling iteration, the discrepancy between the true and estimated profiles is expected to be reduced along with the iterations.

C. Approximation to Profiling Accuracy

In this section, we derive the approximation to the profiling accuracy $\omega$, which provides insights into how each individual sensor contributes to the overall profiling accuracy. In Section VII, we will develop efficient movement scheduling algorithms based on the approximated profiling accuracy. Our extensive numerical studies show that if sensors are uniformly and randomly distributed in the deployment region that covers the diffusion source, $\epsilon$ is close to zero. For instance, when $\sigma^2 = 1$ and 30 sensors are uniformly and randomly deployed in a circular region with a radius of 80 meters, the average value of $\epsilon$ is 0.019 with a standard deviation of $7.9 \times 10^{-4}$ when the diffusion source is at the center of the circular region. More extensive evaluations for the impacts of network density, deployment region size, and diffusion source location on $\epsilon$ can be found in [26]. After setting $\epsilon = 0$, the profiling accuracy can be safely approximated by $\omega \approx L_{X_1} L_{X_2} + L_{Y_1} L_{Y_2} = \sigma^{-2} e^{-2\beta d_i^2} (d_i^2 - (\hat{x}_i^2 + \hat{y}_i^2))$.

By defining $\omega_i = \sigma^{-2} e^{-2\beta d_i^2} (d_i^2 - (\hat{x}_i^2 + \hat{y}_i^2))$, the approximated profiling accuracy can be expressed as $\omega \approx \sum_{i=1}^{N} \omega_i$. Therefore, $\omega_i$ can be regarded as the contribution of sensor $i$ to the overall profiling accuracy. However, from Eq. (4), both $\hat{x}_i$ and $\hat{y}_i$ depend on all sensors’ distance from the source. Therefore, $\omega_i$ correlates with all sensors’ positions. Such inter-node dependence will lead to highly complex movement scheduling. We now propose an approximation to $\hat{x}_i^2 + \hat{y}_i^2$, which reduces the inter-node dependence among sensors. Our extensive numerical studies show that $\hat{x}_i^2 + \hat{y}_i^2$ can be well approximated by the minimum squared distance between the source and the sensors excluding sensor $i$, i.e., $\hat{x}_i^2 + \hat{y}_i^2 \approx \min_{j \in [1, N], j \neq i} d_j^2$. For instance, if 35 sensors are uniformly deployed in a circular region centered at the source and with a radius of 100 meters, the average relative error of the approximation is 4.4%. More evaluation results about the approximation can be found in [26]. By this approximation, the contribution of sensor $i$ only depends on the minimum distance of other sensors. The approximations in this section are summarized as follows:

$$\omega \approx \sum_{i=1}^{N} \omega_i, \quad \omega_i \approx \sigma^{-2} e^{-2\beta d_i^2} \left( d_i^2 - \min_{j \in [1, N], j \neq i} d_j^2 \right).$$

VI. DIFFUSION PROCESS PROFILING USING MOBILE SENSORS

In this section, we first present our MLE-based diffusion process profiling algorithm, which estimates the parameters of the diffusion process, i.e., $\Theta$, in each profiling iteration. We then formally formulate the movement scheduling problem for mobile sensors, which aims to maximize the profiling accuracy in the subsequent profiling iteration.

A. MLE-based Diffusion Process Profiling Algorithm

MLE and Bayesian parameter estimation are two typical parameter estimation approaches [25]. The Bayesian parameter estimation relies on the prior probability distribution of the parameters, which is often unknown and difficult to model in practice. For source localization problems, geometric trilateration approaches have also been widely adopted [13] [14]. However, as trilateration approaches do not explicitly deal with the random measurement noises,
they often yield inferior accuracy compared with MLE [14]. Therefore, in this paper, we adopt MLE to estimate the profile of the diffusion process.

MLE aims to maximize the log-likelihood given by Eq. (3). Formally, $\hat{\Theta}(x) = \arg\max_\Theta l(\Theta|x)$. The expectation maximization algorithm [27] has been widely adopted to solve this optimization problem. We now discuss the estimation accuracy of the above MLE-based profiling algorithm. In our problem, as the concentration decays with the distance from the source, sensor measurements, i.e., $z_1, z_2, \ldots, z_N$, are not identically distributed. The asymptotic efficiency of MLE based on non-identically distributed observations has been proved in [28]. In other words, the variance of an estimated parameter converges to its CRB when the number of sensors, i.e., $N$, approaches infinity. We now evaluate the efficiency of MLE with realistic setting of $N$. Fig. 3 shows several numerical results, where a number of sensors are uniformly and randomly deployed in a squared region. We can see that $\text{Var}(\tilde{x}_0)$ approaches the CRB as the number of sensors increases. Moreover, we evaluate the impact of sensor sampling scheme on the convergence of MLE. Note that sensor’s signal-to-noise ratio (SNR) increases with the number of samples $K$. We can see from Fig. 3 that the MLE shows better convergence if SNR is higher. For instance, if $K = 10$ and $N \geq 10$, the difference between $\text{Var}(\tilde{x}_0)$ and CRB($\tilde{x}_0$) is insignificant. Therefore, $\omega$ characterizes the performance of the MLE-based profiling algorithm.

B. Movement Scheduling Problem for Diffusion Profiling

Because of the limited mobility and energy budget of aquatic mobile sensors, the sensor movements must be efficiently scheduled in achieving the maximum profiling accuracy. As the power consumption of sensing, computation and radio transmission is significantly less than that of mechanical locomotion, in this paper, we only consider the locomotion energy. For instance, as discussed in [29], the energy consumed in moving one meter is sufficient for transmitting more than 100,000 TinyOS messages with default size. Moreover, as the locomotion energy is approximately proportional to the moving distance, we can employ moving distance to characterize the locomotion energy. To simplify the motion control of sensors, we assume that a sensor moves straightly in each profiling iteration and the moving distance is always multiple of $l$ meters, where $l$ is referred to as step. We note that this model is motivated by the fact that aquatic sensors are only capable of simple motion control due to mechanical and computational limitations. As the estimation and movement are performed in an iterative manner as discussed in Section IV, we only focus on the movement scheduling in one profiling iteration. Denote $m_i \in \mathbb{Z}^+$ and $\phi_i \in [0, 2\pi)$ as the number of steps and movement orientation of sensor $i$ in a profiling iteration, respectively. Our objective is to maximize the expected profiling accuracy after sensor movements, subject to total energy budget and individual energy budget for each sensor. The movement scheduling problem for diffusion profiling is formally formulated as follows:

Problem. Suppose a total of $M$ steps can be allocated to sensors and sensor $i$ can move at most $L_i$ meters in a profiling iteration. To find the allocation of steps and movement orientations for all $N$ sensors, i.e., $\{m_i, \phi_i|i \in [1, N]\}$, such that the profiling accuracy $\omega$ (defined by Eq. (7)) after sensor movements is maximized, subject to:

$$
\begin{align*}
\sum_{i=1}^{N} m_i &\leq M, \\
 m_i \cdot l &\leq L_i, \quad \forall i.
\end{align*}
$$

Eq. (9) upper-bounds the total locomotion energy in a profiling iteration. Eq. (10) can be used to constraint the energy consumption of individual sensors. For instance, $L_i$ can be specified according to sensors’ residual energies. Moreover, $L_i$ can also be specified to ensure the delay of a profiling iteration. If sensors move at a constant speed of $v$ m/s and a profiling iteration is required to be completed within $\tau$ seconds to achieve temporal resolution of profiling, $L_i$ can be set by $L_i = v \cdot \tau$. An exhaustive search to the above problem would yield an exponential complexity with respect to $N$, which is $O\left(\left(\frac{L_0}{\omega \cdot l} \cdot \frac{M}{L_0} \right)^N\right)$ where $\omega_0$ is the granularity in searching the movement orientation. Such a complexity is prohibitively high as the problem needs to be solved in each profiling iteration by the cluster head.

VII. SENSOR MOVEMENT SCHEDULING ALGORITHMS

In this section, we propose a greedy movement scheduling algorithm based on gradient descent and a near-optimal algorithm based on dynamic programming to solve the problem formulated in Section VI-B.

A. Greedy Movement Scheduling Algorithm

Gradient descent is a widely adopted algorithm that finds a local optimum of a function. In this paper, we propose a greedy movement scheduling algorithm based on the gradient descent approach. We first discuss how to determine the movement orientations of sensors. As the profiling accuracy $\omega$ given by Eq. (7) is a function of all sensors’ positions, we can compute the gradient of $\omega$ with respect to the position of sensor $i$ (denoted by $\nabla_i \omega$), which is formally given by $\nabla_i \omega = \left[\frac{\partial \omega_i}{\partial x_i}, \frac{\partial \omega_i}{\partial y_i}\right]^T$. When all sensors except sensor $i$ remain stationary, the metric $\omega$ will increase fastest if sensor $i$ moves in the orientation given by the argument of $\nabla_i \omega$. Therefore,
in the greedy movement scheduling algorithm, we let \( \phi_i = \angle(\nabla_i \omega) \). We note that sensors will move simultaneously when executing the movement schedule. We then discuss how to allocate the movement steps. The magnitude of \( \nabla_i \omega \), denoted by \( \|\nabla_i \omega\| \), quantifies the steepness of the metric \( \omega \) when sensor \( i \) moves in the orientation \( \angle(\nabla_i \omega) \) while other sensors remain stationary. Therefore, in the greedy algorithm, we propose to proportionally allocate the movement steps according to all sensors’ gradient magnitudes. Specifically, \( m_i \) is given by \( m_i = \min \left\{ \frac{\|\nabla_i \omega\|}{\sum_{j=1}^{N} \|\nabla_j \omega\|} \cdot M, \left\lfloor \frac{L}{\varphi_i(1)} \right\rfloor \right\} \). Note that the \( \left\lfloor \frac{L}{\varphi_i(1)} \right\rfloor \) in the \( \min \) operator ensures the constraint Eq. (10). This greedy algorithm has linear complexity, i.e., \( O(N) \), which is preferable for the cluster head with limited computation resource.

B. Radial Movement Scheduling Algorithm

In this section, we will propose a new movement scheduling algorithm based on the approximations discussed in Section V-C. In this algorithm, each sensor moves towards or outward from the estimated source location along the straight line connecting the estimated source location and the sensor’s current position. Hence, it is referred to as the radial algorithm. We first discuss how to determine sensors’ movement orientations and then present a dynamic-programming-based movement step allocation algorithm.

1) Determining movement orientations: From Eq. (8), the contribution of sensor \( i \), i.e., \( \omega_i \), depends on the minimum distance of other sensors. Because of such inter-node dependency, it is difficult to derive the optimal distance for each sensor that maximizes the overall profiling accuracy \( \omega \). Therefore, the problem is a non-linear and non-convex (due to the \( \min \) operator in Eq. (8)) constrained optimization. Several stochastic search algorithms such as simulated annealing can find the optimal solution. However, these algorithms often have prohibitively high complexities. In the radial algorithm, we fix the sensor closest to the estimated source location and only schedule the movements of other sensors in each profiling iteration. As the sensor closest to the source receives the highest SNR, repositioning other sensors will likely yield more performance gain. Moreover, this sensor can serve as the cluster head that receives measurements from sensors and computes the movement schedule. It is hence desirable to keep it stationary due to its higher energy consumption in computation and communication. We note that the sensor closest to the source may be different in each iteration after sensors’ movements, resulting in rotation of cluster head among sensors. By fixing the sensor closest to the source, the distance \( d_i \) that maximizes the approximated \( \omega_i \), denoted by \( d^*_i \), can be directly calculated by

\[
d^*_i = \sqrt{\frac{1}{2\beta} + \min_{j \in [1,N]} \frac{d^2_j}{d^2_j}, \forall i \neq \arg \min_{j \in [1,N]} d_j.}
\]

(11)

Note that as \( \beta \) is a time-dependent variable, \( d^*_i \) also changes with time and hence should be updated in each profiling iteration. Eq. (11) allows us to easily determine the movement orientation of sensor \( i \). Specifically, if \( d_i > d^*_i \), sensor \( i \) will move towards the estimated source location; Otherwise, sensor \( i \) will move in the inverse direction. Formally, by defining \( \delta = \text{sgn}(d^*_i - d_i) \), we can express the movement orientation of sensor \( i \) as \( \phi_i = \angle(\delta \cdot x_i \cdot \delta \cdot y_i) \).

2) Allocating movement steps: We now discuss how to allocate the movement steps. In the rest of this section, when we refer to sensor \( i \), we assume \( i \neq \arg \min_{j \in [1,N]} d_j \). After sensor \( i \) moves \( m_i \) steps in the orientation of \( \phi_i \), its contribution to the overall profiling accuracy is given by

\[
\omega_i(m_i) = \frac{(d_i + \delta \cdot m_i \cdot \delta \cdot y_i)^2 - \min_{j \in [1,N]} d_j^2}{2\sigma^2 \bar{D}(d_i + \delta \cdot m_i \cdot \delta \cdot y_i)^4}, \quad (12)
\]

where \( \min_{j \in [1,N]} d_j^2 \) in Eq. (12) is a constant for sensor \( i \), and \( \beta \) can be predicted based on its current estimate to capture the temporal evolution of the diffusion, i.e., \( \beta = \frac{1}{1/S + D \cdot T} \). Given the radial movement orientations described earlier, the problem formulated in Section VI-B is equivalent to maximizing \( \sum_i \omega_i(m_i) \) subject to the constraints Eqs. (9) and (10), which can be solved by a dynamic programming algorithm as follows.

We number the sensors by \( 1, 2, \ldots, N - 1 \), excluding the sensor that is the closest to the estimated source location. Let \( \Omega(i, m) \) be the maximum \( \omega \) when the first \( i \) sensors are allocated with \( m \) steps. Formally, \( \Omega(i, m) = \max_{m_j} \sum_{j=1}^{i} \omega_j(m_j) \) subject to \( 0 \leq m_j \leq \lfloor L/l \rfloor \) and \( \sum_{j=1}^{i} m_j = m \). The dynamic programming recursion that computes \( \Omega(i, m) \) can be expressed as

\[
\Omega(i, m) = \max_{0 \leq m_i \leq \lfloor L/l \rfloor} \{ \Omega(i-1, m-m_i) + \omega_i(m_i) \}, \quad (13)
\]

The initial condition of the above recursion is \( \Omega(0, m) \) for \( m \in [0, M] \). According to Eq. (13), at the \( i^{th} \) iteration of the recursion, the optimal value of \( \Omega(i, m) \) is computed as the maximum value of \( \lfloor L/l \rfloor \) cases which have been computed in previous iterations of the recursion. Specifically, for the case where sensor \( i \) moves \( m_i \) steps, the maximum profiling accuracy \( \omega \) of the first \( i \) sensors allocated with \( m \) steps can be computed as \( \Omega(i-1, m-m_i) + \omega_i(m_i) \), where \( \Omega(i-1, m-m_i) \) is the maximum \( \omega \) of the first \( i - 1 \) sensors allocated with \( m - m_i \) steps. The maximum overall profiling accuracy is given by \( \omega^* = \max_{m \in [1,M]} \Omega(N-1, m) \).

We now describe how to construct the movement schedule using the above dynamic programming recursion. The movement schedule of sensor \( i \) is represented by a pair \( (i, m_i) \). For each \( \Omega(i, m) \), we define a movement schedule \( S(i, m) \) initialized to be an empty set. The \( S(i, m) \) is filled incrementally in each iteration when computing \( \Omega(i, m) \). Specifically, in the \( i^{th} \) iteration of the recursion, if \( \Omega(i-1, m-m_x) + \omega_x(m_x) \) gives the maximum value among all cases, we add a movement schedule \( (i, m_x) \) to \( S(i, m) \). Formally, \( S(i, m) = S(i-1, m-m_x) \cup \{(i, m_x)\} \), where \( m_x = \arg \max_{0 \leq m_x \leq \lfloor L/l \rfloor} \{ \Omega(i-1, m-m_x) \} \). The complexity of the dynamic programming is \( O((N-1)M^2) \), where \( N \) is the number of sensors.
and $M$ is the number of allocatable movement steps in a profiling iteration. Finally, we note that the above dynamic programming algorithm finds the optimal solution if the overall profiling accuracy is strictly the linear combination of sensors’ contributions. As the approximations are accurate as discussed in Section V-C, the overall radial algorithm can find the near-optimal movement schedule of mobile sensors.

VIII. PERFORMANCE EVALUATION

A. Evaluation Methodology and Settings

1) Methodology: We conduct extensive simulations to evaluate the proposed movement schedule algorithms. The simulation programs are written in Matlab. The diffusion source is at the origin of the coordinate system, i.e., $x_0 = y_0 = 0$. The sensors are deployed uniformly and randomly in the square region of $200 \times 200 \text{m}^2$ centered at the origin. The reading of a sensor is set to be the sum of the concentration of sensors’ contributions. As the approximations are accurate overall profiling accuracy is strictly the linear combination of energy. In practice, various approaches can be applied to initiate the profiling process, e.g., by comparing the average measurement to a threshold that ensures good SNRs. The sensor speed is set to $v = 2.5 \text{ m/min}$. The typical speed of real low-power aquatic mobile platforms is several meters per minute, e.g., 1.8 to 6 m/min in [6]. Other settings include $l = 0.5 \text{ m}$, $\tau = 60 \text{ s}$ and $K = 2$, unless otherwise specified.

2) Settings: In the simulations, the amount of discharged substance is set to be $A = 0.7 \times 10^6 \text{ cm}^3$ (i.e., $0.7 \text{ m}^3$) unless otherwise specified. The diffusion coefficient is set to be $D = 5,000 \text{ cm}^2/\text{s}$. Note that the settings of $A$ and $D$ are comparable to the real field experiments reported in [21] [23] where $2$ to $5 \text{ m}^3$ of diesel oil were discharged into sea and the estimated diffusion coefficient ranges from $2,000 \text{ cm}^2/\text{s}$ to $7,000 \text{ cm}^2/\text{s}$. The noise standard deviation is set to be $\sigma = 1 \text{ cm}^3/\text{m}^2$, i.e., $1 \text{ cm}^3$ diffusion substance per unit area.\(^1\) To easily compare various movement scheduling algorithms, we let the first profiling round always start at $t = 1800 \text{ s}$, i.e., half an hour after the discharge. At $t = 1800 \text{ s}$, the average received SNR is around 10:1. The rationale of this setting is that moving sensors too early (i.e., at low SNRs) leads to little improvement on profiling accuracy, resulting in waste of energy. In practice, various approaches can be applied to initiate the profiling process, e.g., by comparing the average measurement to a threshold that ensures good SNRs. The sensor speed is set to $v = 2.5 \text{ m/min}$. The typical speed of real low-power aquatic mobile platforms is several meters per minute, e.g., 1.8 to 6 m/min in [6]. Other settings include $l = 0.5 \text{ m}$, $\tau = 60 \text{ s}$ and $K = 2$, unless otherwise specified.

B. Simulation Results

1) Sensor movement trajectories: We first visually compare the sensor movement trajectories computed by the greedy and radial movement scheduling algorithms. Total 20 sensors are deployed. Fig. 4 shows the movement trajectories of sensors in the first 15 profiling rounds. The arrow in Fig. 4 represents the movement orientation in the 15th profiling round. The numbers in the captions of sub-figures are the profiling accuracy after the first 15 profiling rounds.

\(^1\)As we adopt a 2-dimensional model to characterize the diffusion process, the physical unit of concentration is $\text{cm}^3/\text{m}^2$. As observed in the field experiments [21], diesel oil can penetrate down to several meters from the water surface. As a result, the equivalent $\sigma$ that accounts for the depth dimension ranges from $0.1 \text{ cm}^3/\text{m}^3$ to $1 \text{ cm}^3/\text{m}^3$. Our setting is consistent with the noise standard derivation of the crude oil sensor Cyclops-7 [24], which is $0.1 \text{ cm}^3/\text{m}^3$. 

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{figure4.png}
\caption{(a) greedy: $\omega = 15400$ (b) radial: $\omega = 16400$}
\end{figure}
to the maximum $\omega$ after the movements of sensors and the temporal evolution of the diffusion.

2) Profiling accuracy: In the second set of simulations, we evaluate the accuracy in estimating the diffusion profile $\Theta$. Total 10 sensors are deployed and our evaluation lasts for 15 profiling rounds. Fig. 5 plots the profiling accuracy $\omega$ (defined in Eq. (7)) based on estimated diffusion profile $\Theta$. The curve labeled with “stationary” is the result if all sensors always remain stationary. Nevertheless, we can see that the profiling accuracy improves over time because of the temporal evolution of the diffusion. Moreover, the radial algorithm outperforms the greedy and SNR-based algorithms by 16% and 50% in terms of $\omega$ at the 15th profiling round, respectively. Moreover, the accuracy performance of the radial algorithm is very close to the annealing algorithm that can find the near-optimal solution.

Fig. 6 plots the average of $\text{Var}(\tilde{x}_0)$ and $\text{Var}(\tilde{y}_0)$ in each profiling round under various settings of the discharged substance amount $A$. Note that in order to evaluate the variances in each profiling round, the sensors perform many rounds of MLE, where each round yields a pair of $(\tilde{x}_0, \tilde{y}_0)$. The variances $\text{Var}(\tilde{x}_0)$ and $\text{Var}(\tilde{y}_0)$ are calculated from all rounds. We can see from Fig. 6 that the variances decrease with $A$. As sensors receive higher SNRs in the case of higher $A$, our result is consistent with the intuition that the estimation error decreases with SNR. Compared with the SNR-based algorithm, the radial algorithm reduces the variance in estimating diffusion source location by 36% for $A = 0.7 \times 10^6 \text{cm}^3$. Compared with the greedy algorithm, the reductions are 12% and 18% for $A = 0.7 \times 10^6$ and $1.4 \times 10^6 \text{cm}^3$, respectively.

We have also evaluated the accuracy in estimating the substance amount $A$ and elapsed time $t$. Both the greedy and radial algorithms can achieve a high accuracy. For instance, with the radial algorithm, the relative error in estimating $A$ is within 1.4%. Due to space limit, detailed evaluation results are omitted here and can be found in [26].

3) Impact of sampling, source bias and network density: In this set of simulations, the profiling error is characterized by the average of $\text{Var}(\tilde{x}_0)$ and $\text{Var}(\tilde{y}_0)$ after 15 profiling rounds. Except for the evaluations on the network density, total 10 sensors are deployed.

In the temporal sampling scheme presented in Section III-B, a sensor yields the average of $K$ continuous samples as the measurement to reduce noise variance. Fig. 7 plots the profiling error versus $K$. We can see that the profiling error decreases with $K$. The relative reductions of profiling error by the radial algorithm with respect to the greedy and SNR-based algorithms are about 18% and 30%, respectively, when $K$ ranges from 2 to 20.

The approximations discussed in Section V-C assume that the sensors are uniformly and randomly deployed around the diffusion source. In this set of simulations, we evaluate the impact of source location bias on the profiling accuracy. Specifically, the diffusion source appears at $(\delta, 0)$, where $\delta$ is referred to as the source location bias. Fig. 8 plots the profiling error versus $\delta$. To jointly account for the impact of random sensor deployment, for each setting of $\delta$, we deploy a number of networks and show the error bars in Fig. 8. We can see that the radial algorithm is consistently better than other algorithms. Moreover, we can see that the radial algorithm is robust to the source location bias.

Fig. 9 plots the profiling error versus the number of sensors. When more sensors are deployed, the profiling error can be reduced. The radial algorithm is consistently better than other algorithms. For all algorithms, the profiling error is reduced by about 40% when the number of sensors increases from 10 to 15. Moreover, the relative reduction of profiling error decreases with the number of sensors.

4) Execution time: We have implemented the greedy and radial movement scheduling algorithms on TelosB mote [7] with a processor frequency of 8 MHz. When 10 sensors are to be scheduled, the two algorithms require 1 and 8.8 kilobytes RAM, respectively. Fig. 10 plots the average execution time versus the number of sensors to be scheduled. For both algorithms, the execution time linearly increases with the number of sensors, which is consistent with our complexity analysis. The execution time of the radial algorithm is three times of the greedy algorithm. For the radial algorithm, 30% execution time is devoted to computing a look-up table consisting of each sensor $i$’s contribution, i.e., $\omega_i(m_i)$, given all possible values of $m_i$. Note that in each iteration of the annealing algorithm, a new look-up table needs to be computed due to changed movement orientations. Therefore,
the execution time of the \textit{annealing} algorithm highly depends on the number of iterations that can be very large. As a result, the \textit{annealing} algorithm is infeasible on mote-class platforms. Note that our current implementation is a preliminary version, where the algorithms are carried out by floating-point computation. From our previous experiences, the fixed-point arithmetic can significantly reduce the execution time. Moreover, if more powerful mote platform is adopted, the delay due to the movement scheduling can be ignored. For instance, the projected execution time on Imote2 [7] with a processor frequency of 416 MHz is within 2 seconds for the results in Fig. 10.

IX. CONCLUSION

This paper proposes an accuracy-aware diffusion process profiling approach using mobile sensor networks. Our approach features an iterative learning process where the sensors reposition themselves to progressively improve the profiling accuracy along the iterations. We develop various movement scheduling algorithms for mobile sensors, including a light-weight greedy algorithm and a near-optimal algorithm with a higher complexity. Extensive simulations with realistic settings validate the effectiveness of our approach.

ACKNOWLEDGMENT

We thank Jianxun Wang for his constructive discussions in this work.

REFERENCES

APPENDIX

A. Derivation of Cramér-Rao bound

In this section, we present the derivation of CRB for diffusion process profiling. To get the closed-form of J, we apply the block matrix manipulations, and express it with block matrices:

\[
J = \begin{bmatrix}
J_{11} & J_{12} \\
J_{21} & J_{22}
\end{bmatrix},
\]

where the blocks \(J_{11}, \ J_{12}, \ J_{21}, \ \text{and} \ J_{22}\) are \(2 \times 2\) matrices that contain complicated expressions of \(x_i, y_i, d_i, \alpha, \ \text{and} \ \beta\), and their expressions are given by

\[
J_{11} = 4\alpha^2 \beta^2 \left[ \sum_{i=1}^{N} \frac{x_i^2}{\sigma_{x_i}^2 d_i^2} + \sum_{i=1}^{N} \frac{x_i y_i}{\sigma_{x_i} \sigma_{y_i} d_i^2} \right],
\]

\[
J_{12} = 2\alpha \beta \left[ \sum_{i=1}^{N} \frac{a d^2 x_i}{\sigma_{x_i}^2 (x_i d_i)^2} + \sum_{i=1}^{N} \frac{a d^2 y_i}{\sigma_{x_i} \sigma_{y_i} (x_i d_i)^2} \right],
\]

\[
J_{21} = 2\alpha \beta \left[ \sum_{i=1}^{N} \frac{a d^2 x_i}{\sigma_{x_i}^2 (x_i d_i)^2} + \sum_{i=1}^{N} \frac{a d^2 y_i}{\sigma_{x_i} \sigma_{y_i} (x_i d_i)^2} \right],
\]

\[
J_{22} = \sum_{i=1}^{N} \frac{a^2 d^4}{\sigma_{x_i}^4 (x_i d_i)^4} + \sum_{i=1}^{N} \frac{a^2 d^2 y_i}{\sigma_{x_i}^2 \sigma_{y_i} (x_i d_i)^2}.
\]

The inverse of \(J\) is given by

\[
J^{-1} = \begin{bmatrix}
F^{-1} - F J_{12} J_{22}^{-1} J_{12}^\top \left( I + J_{22} F^{-1} J_{12} \right)^{-1}
\end{bmatrix}.
\]

where \(F = J_{11} - J_{12} J_{22}^{-1} J_{21}\). The CRBs of the source location coordinates, i.e., \(\text{CRB}(x_0)\) and \(\text{CRB}(y_0)\), are given by the diagonal elements of \(F^{-1}\).

With the above notations, \(F\) can be expressed as

\[
F = 2\alpha^2 \beta^2 \begin{bmatrix}
2L_{X_1}L_{X_2}L_{Y_1}L_{Y_2} + L_{X_1}L_{Y_2} + L_{Y_1}L_{X_2} \\
2L_{Y_1}L_{Y_2}
\end{bmatrix}.
\]

Therefore, the CRBs for \(x_0\) and \(y_0\) are given by

\[
\text{CRB}(x_0) = F^{-1}_{1,1} = 1 = \frac{(4\alpha^2 \beta^2)^{-1}}{L_{X_1}L_{X_2} - \left( L_{X_1}L_{Y_2} + L_{X_2}L_{Y_1} \right)^2},
\]

\[
\text{CRB}(y_0) = F^{-1}_{2,2} = 1 = \frac{(4\alpha^2 \beta^2)^{-1}}{L_{Y_1}L_{Y_2} - \left( L_{X_1}L_{Y_2} + L_{X_2}L_{Y_1} \right)^2}.
\]

B. Profiling Accuracy Approximation

1) Evaluation of \(\epsilon\): This section evaluates the impacts of network density, deployment region size, and diffusion source location on \(\epsilon\), which is defined as \(\frac{(L_{X_1}L_{Y_2} + L_{X_2}L_{Y_1})^2}{4L_{X_1}L_{X_2}L_{Y_1}L_{Y_2}}\). For different configuration of the above mentioned settings, we calculate the expectation \(E(\epsilon)\) and standard derivation \(\sigma(\epsilon)\) over a large number (10,000 in our evaluation) of sensor deployments. In each deployment, sensors’ positions are generated randomly according to a uniform distribution over the deployment region.

First, we jointly evaluate the impact of network density and deployment region size under the centered diffusion source condition. Network density is indicated by the number of sensors deployed in the region of interest. We let the sensor amount \(N\) varies from 5 to 50, and the deployment region size \(R\) varies from 5 meters and 100 meters. For each combination of \(N\) and \(R\), we randomly generate 10,000 deployment settings to evaluate \(E(\epsilon)\) and \(\sigma(\epsilon)\). The selected results are shown in Table II.

Second, we evaluate the impact of diffusion source location, i.e., the bias of the source location to the region centroid. Specifically, we set the region of interest as a circle centered at \((0,0)\) with a radius of 100 meters, and let the source location bias \(\delta\) range from 0 to 50 meters. For each \(\delta\), \(N\) varies from 5 to 50. For each combination of \(\delta\) and \(N\), we randomly generate 10,000 deployment settings to evaluate \(E(\epsilon)\) and \(\sigma(\epsilon)\). The selected results are shown in Table III.

2) Evaluation of Profiling Accuracy Approximation: This section discusses the Monte Carlo method to evaluate the accuracy of the profiling accuracy approximation given in Eq. (8), by calculating the cumulative distribution function (CDF) versus the error rate. The Monte Carlo method computes CDF as follows. Define deployment set \(\mathbf{P} = \ldots\)
Table III
BIASED DIFFUSION SOURCE \((\varepsilon, \sigma(\varepsilon))\).

<table>
<thead>
<tr>
<th>(K = 100\text{ m})</th>
<th>(N = 30)</th>
<th>(N = 40)</th>
<th>(N = 50)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\delta = 5\text{ m})</td>
<td>0.0219, 0.0303</td>
<td>0.0164, 0.0230</td>
<td>0.0124, 0.0171</td>
</tr>
<tr>
<td>(\delta = 10\text{ m})</td>
<td>0.0229, 0.0322</td>
<td>0.0172, 0.0246</td>
<td>0.0128, 0.0177</td>
</tr>
<tr>
<td>(\delta = 15\text{ m})</td>
<td>0.0246, 0.0344</td>
<td>0.0178, 0.0256</td>
<td>0.0140, 0.0200</td>
</tr>
<tr>
<td>(\delta = 20\text{ m})</td>
<td>0.0274, 0.0383</td>
<td>0.0190, 0.0267</td>
<td>0.0148, 0.0208</td>
</tr>
<tr>
<td>(\delta = 25\text{ m})</td>
<td>0.0298, 0.0422</td>
<td>0.0219, 0.0305</td>
<td>0.0171, 0.0237</td>
</tr>
</tbody>
</table>

Figure 11. CDF of profiling accuracy approximation.

\(\{P_1, P_2, \ldots, P_n\}\), where \(n\) is the number of deployed sensors. Define \(B\) is the event the approximation error rate is no greater than \(\varepsilon\), where \(\varepsilon \in [0, 1]\). We draw \(T\) samples of \(P\) and count the number of occurrences of event \(B\), which is denoted as \(T'\). The ratio \(T'/T\) characterizes the CDF of \(B\) at \(\varepsilon\). Figure B2 shows the CDF vs. error rate curve, in which the radius of the deployment region is 100 meters and sensor density varies.

**C. Estimated Diffusion Profile**

In this section, we present the full estimated diffusion process profile. Note that our approaches not only effectively address the diffusion source localization, but also accurately estimate the initial substance amount \(\tilde{A}\) and the elapsed time \(\tilde{t}\). Fig. C plots the relative error of \(\tilde{A}\), which is calculated as \((\tilde{A} - A)/A\). We can see that all algorithms can accurately estimate \(A\) with a relative error less than 1.4%. Moreover, the relative errors decrease with the elapsed time. Fig. C plots the \(\tilde{t}\) versus the groundtruth time. We can see that the elapsed time can be accurately estimated. The relative error of \(\tilde{t}\) is 1.0%.

Figure 12. The estimated substance amount \(\tilde{A}\).

Figure 13. The estimated elapsed time \(\tilde{t}\).