Optimization Choice in Wireless Sensor Network: Air Dispersion Sensing and Reconstruct

Shuo-Yang Wang, Yun-Da Tsai

Department of Electronic Engineering
National Taiwan University
Taipei, R.O.C
b97504026@ntu.edu.tw b97502049@ntu.edu.tw

I. Introduction

When we face some dangerous situation, such as a factory explosion. The time to evacuate people around the factory is very important. In this paper, we will introduce an algorithm to select which sensors to send information to the cluster head, this algorithm can minimize the square mean error between reconstruction value and simulation value. After receiving information from sensors, the cluster head can reconstruct the entire map according to these information, then we will find out which area may be polluted most heavily in the near future and should be evacuated as soon as possible.

II. Overview

A. Atmospheric dispersion modeling

In order to simulate the air pollution, we find out a well known air dispersion model which is a partial differential equation based on the law of conservation of mass. The solution of this differential equation can give us a function coordination to calculate the mass concentration of every point in the space.

B. Scenario

In the atmospheric dispersion model we assumed that the wind speed and direction is already known. However, in more realize case this assumption is over ideal assumption. Hence, based on this model, we relax the restriction of information about wind and make the wind speed as a random variable with some resonable distribution. Then we set up the pollutant concentration sensor which distributed uniformly as the cellular base station.

C. Algorithm

When the pollutant is spreading in the space, the sensor start sensing the pollutant concentration nearby itself, and re-transmitted the data back to the cluster head. After that, server will reconstruct the pollutant distribution map which can point out that some certain area suffer from the pollutant seriously. On the other hand, our wireless sensor might be restricted by the channel capacity, and not all of the sensor could transmit data to cluster head. According to the scenario, we can derive a optimization problem with the entropy as the objective function, in order to solve this problem, we apply the cross entropy algorithm to find the approximate solution.

D. Performance analysis

After the approximate solution is derived, we can conclude that the mean square error of reconstruction map will be bounded through theoretical result of information theory. Meanwhile, in order to validate the theoretical bound, we keep the air dispersion modeling result as the theoretical value and the reconstruction value as the measure value, and we can calculate the the mean square error from these two part of result. Considering the mean square result, we can verify if the experiment result is consist with the theoretical bound.

III. Model

The air dispersion model in this paper is based on , which simulates air dispersion all in matlab code. At first, it provides a governing equation and some simplifying assumptions that will permit us to derive a close-form analytical solution.

A. Governing Equation

We restrict our attention at the outset to the transport of a single contaminant whose mass concentration at location $\vec{x} = (x, y, z) \in \mathbb{R}^3$ and time $t \in \mathbb{R}$ can be described by a smooth function $C(\vec{x}, t)$. The law of conservation of mass for C may be in differential form as

$$\frac{\partial C}{\partial t} + \nabla \cdot \vec{J} = S, \tag{1}$$

where $S(\vec{x}, t)$ is a source or sink term, and the vector function $\vec{J}(\vec{x}, t)$ represents the *mass flux* of contaminant owing to the combined effects of diffusion and advection. The diffusive contribution to the flux arises from turbulent eddy motion in the atmosphere. The main result is that atmospheric diffusion may be assumed to follow Fick's law, which states that the diffusive flux is proportional to the concentration gradient, or $\vec{J}_D = -\mathsf{K}\nabla C$. The negative sign ensures that the contaminant will flow from regions of high concentration to regions of low concentration, and the diffusion coefficient $\mathsf{K}(\vec{x}) = diag(K_x, K_y, K_z)$ is a diagonal matrix whose entries are the turbulent eddy diffusivities and are in general functions of position. The second contribution to the flux is due to simple linear advection by the wind, which can be expressed as $\vec{J}_A = C\vec{u}$ where $\vec{u}$ is the wind velocity. By adding these two contributions together, we obtain the total flux $\vec{J} = \vec{J}_D + \vec{J}_A = C\vec{u} - \mathsf{K}\nabla C$, which after substituting into the equation of conservation of mass (1) yields the three-dimensional advection-diffusion equation

$$\frac{\partial C}{\partial t} + \nabla \cdot (C\vec{u}) = \nabla \cdot (\mathsf{K}\nabla C) + S. \tag{2}$$

B. Assumptions

1) The contaminant is emitted at a constant rate Q from a single point source $\vec{x} = (0, 0, H)$ located at height H above the ground surface, as depicted in Fig. 1. Then the source term may be written as $S(\vec{x}) = Q\delta(x)\delta(y)\delta(z - H)$, where $\delta(\cdot)$ is the Dirac delta function.
2) The wind velocity is constant and aligned with the positive x-axis so that $\vec{u} = (u, 0, 0)$ for some constant $u \geqslant 0$.
3) The solution is steady state, which is reasonable if the wind velocity and all other parameters are independent of time and the time scale of interest is long enough.
4) The eddy diffusivities are functions of the downwind distance x only, and diffusion is isotropic so that $K_x(x) = K_y(x) = K_z(x) =: K(x)$.
5) The wind velocity is sufficiently large that diffusion in the x-direction is much smaller than advection; then the term $K_x\partial_x^2 C$ can be neglected.
6) Variations in topography are negligible so that the ground surface can be taken as the plane $z = 0$.
7) The contaminant does not penetrate the ground.

Making use of Assumptions, Eq. (2) reduces to

$$u\frac{\partial C}{\partial x} = K\frac{\partial^2 C}{\partial y^2} + K\frac{\partial^2 C}{\partial z^2} + Q\delta(x)\delta(y)\delta(z - H), \tag{3}$$

after solving this PDE with an appropriate set of boundary conditions, the result we derive is

$$r = \frac{1}{u}\int_0^x K(\xi)d\xi \tag{4}$$

$$c(r, y, z) = \frac{Q}{4\pi u r}\exp\left(-\frac{y^2}{4r}\right) \times \left[\exp\left(-\frac{(z-H)^2}{4r}\right) + \exp\left(-\frac{(z+H)^2}{4r}\right)\right] \tag{5}$$

C. The Ermak solution

Next, we consider deposition and settling. In many practical situations, contaminant particles are more massive than air and so they tend to settle out of the atmosphere at a well-defined rate known as the *settling velocity*, w_{set}. For spherical particles of uniform size, the settling velocity can be approximated using Stokes' law, $w_{set} = 2\rho g R^2/(9\mu)$, where ρ is the particle density, R is the particle radius, μ is the dynamic viscosity of air, and g is the gravitational acceleration. To incorporate the effect of settling, we supplement the advection velocity with a vertical component, $\vec{\mu} = (\mu, 0, -w_{set})$, which means that the advection-diffusion equation becomes

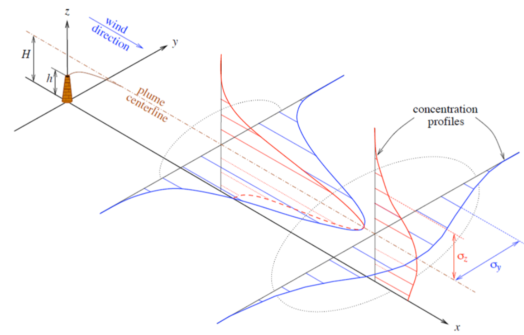

Fig. 1. A contaminant plume emitted from a continuous point source, with direction aligned with the x-axis. Profiles of concentration are given at two downwind locations, and the Gaussian shape of the plume cross-sections are shown relative to the plume centerline.

$$\frac{\partial c}{\partial r} - \frac{w_{set}}{D}\frac{\partial c}{\partial z} = \frac{\partial^2 c}{\partial y^2} + \frac{\partial^2 c}{\partial z^2}. \tag{6}$$

In addition to vertical settling within the atmosphere, observations suggest that taking a no-flux condition at the ground surface is not a reasonable approximation; instead, some portion of particles that reach the surface actually deposit on the ground and are absorbed. Ermak, who was the first person to consider pollutant dispersion with both deposition and settling, applied Laplace transform methods to Eq. (6) and obtained the solution

$$c(r, y, z) = \frac{Q}{4\pi\mu r}\exp\left(-\frac{y^2}{4r}\right)\exp\left(-\frac{w_{set}(z-H)}{2K} - \frac{w_{set}^2 r}{4K^2}\right) \times \left[\exp\left(-\frac{(z-H)^2}{4r}\right) + \exp\left(-\frac{(z+H)^2}{4r}\right) - \frac{2w_o\sqrt{\pi r}}{K}\exp\left(\frac{w_o(z+H)}{K} + \frac{w_o^2 r}{K^2}\right)\operatorname{erfc}\left(\frac{z+H}{2\sqrt{r}} + \frac{w_o\sqrt{r}}{K}\right)\right] \tag{7}$$

where $w_o := w_{dep} - \frac{1}{2}w_{set}$ and $\operatorname{erfc}(x) = 1 - \operatorname{erf}(x)$ is the *complementary error function*. In the following section, we will use Eq. (7) in our matlab simulation.

IV. Scenario

1) Although we can simulate multiple sources, we simulate single source to simplify our problem. The source is located at $\vec{x} = (0, 0, H)$ as mentioned in the Model section.
2) The wind direction is fixed to the positive x-axis. However, the wind speed u is a Gaussian random variable, which we will select before simulation, and this value may be known or unknown due to how we calculate the covariance matrix, for more detail information, you can read the 5) part of this section and the Algorithm section.
3) The emission rate Q is also a Gaussian random variable, which we will select before simulation, and this value

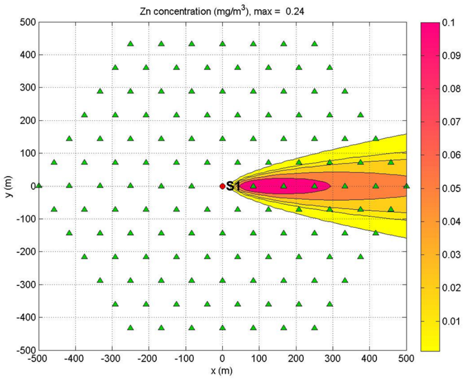

Fig. 2. Single source and honeycomb like scattered sensors

may also be known or unknown due to how we calculate the covariance matrix, for more detail information, you can read the 5) part of this section and the Algorithm section.

4) The sensors are scattered in a honeycomb like pattern and their center focuses at the source. Fig. 2 shows an example of how sensors are located.
5) There are two ways to calculate the covariance matrix(How we calculate covariance matrix is mentioned in the Algorithm section).
   a) We use the wind speed u and emission rate Q we selected before simulation to calculate covariance matrix.
   b) We use the mean and variance of the wind speed u and emission rate Q to calculate covariance matrix, not the indeed value of wind speed u and emission rate Q we selected before simulation.
6) After calculating the covariance matrix, we will use the iteration method to find out which sensors are selected to reconstruct the entire dispersion map, then calculate the mean square error between the reconstructed map and the indeed value simulated.

V. Algorithm

A. Introduction to cross entropy algorithm

a) Motivation: Cross entropy is already applied to solve some complicate optimization problems, such as travelling salesman problem(TSP), and the max-cut problem. On the other hand, cross entropy can also be used to perform rare event-simulation. Consider the property of our problem, we can figure out that if we want to determine a certain node should be chosen or not to be chosen base on the probability indicate by the algorithm output, this output must be fairly approach to 0 or 1. And this property is consist of the cross entropy in rare-event simulation. Hence we would like to use cross entropy algorithm to find the optimization solution of our sensor selection problem.

TABLE I
CHARACTERISTICS

Quantization step	Δ
Channel gain	G_i
TX power	P_i
Bandwidth	W
Noise Power density	N_0
Total time length	T

b) Brief description: To apply the cross entropy algorithm. The constraint and objective must be specified, which will be propose in the following subsection. After that, the CE method involves an iterative procedure where each iteration can be broken down into two phases.

1) Generate a random data sample (trajectories, vectors, etc.) according to a specified mechanism.
2) Update the parameters of the random mechanism based on the data to produce better sample in the next iteration.

After iteration which is large enough, we can make the parameter as our optimization solution.

B. Set up

Here, given the channel capacity constraint and the objective function. First, we have already know that the total transmission time T and total bandwidth W is limited, and we assume that the way of uploading data is using time division multiple access(TDMA). Also the total transmission rate cannot be larger than the channel capacity. Hence, we can derive the follow two constraint equation:

$$
\begin{aligned}
Wlog_2\left(1+\frac{G_iP_i}{WN_0}\right) &\geq \frac{H(X_i)}{t_i} \\
&= \frac{1}{t_i}log_2\left(\frac{2\pi e\sigma_i^2}{\Delta_i^2}\right) \quad i \in S
\end{aligned} \tag{8}
$$

$$
\Sigma_{i\in S}t_i \leq T \tag{9}
$$

On the other hand, our objective is to maximize the total entropy transmitted. According to the correlated Gaussian random source, and combing with the definition of differential entropy, we can derive the total entropy transmitted as a function of covariance matrix :

$$
max_{S\subseteq V} = \frac{1}{2}log_2[(2\pi e)^{|S|}det(\Sigma_S)] - |S|log_2\Delta \tag{10}
$$

$$
= \frac{1}{2}log_2[\left(\frac{2\pi e}{\Delta^2}\right)det(\Sigma_s)] \tag{11}
$$

Where all the terms mentioned above is on the table I.

The covariance matrix can be derived from the air dispersion model. Assume that the emission rate Q of source variate as a stationary Gaussian random process, and the speed of wind is also variate as some probability distribution. Given the concentration field function:

$$c(x,y,z) = \frac{Q}{4\pi ur} exp\left(-\frac{y^2}{4r}\right) \times \left[exp\left(-\frac{(z-H)^2}{4r}\right) + exp\left(-\frac{(z+H)^2}{4r}\right)\right] \tag{12}$$

Then follow the definition of convariance matrix, and let $ur = k$, $z = 0$:

$$\begin{aligned} Cov\left[C_1, C_2\right] &= E\left[C_1 C_2\right] - E\left[C_1\right] E\left[C_2\right] \qquad (13) \\ &= E\left[\frac{Q^2}{16\pi^2 K_1 K_2} exp\left(-u\left(\frac{y_1^2}{4k_1} + \frac{y_2^2}{4k_2}\right)\right) \times 4exp\left(-uH^2\left(\frac{1}{4k_1} + \frac{1}{4k_2}\right)\right)\right] \\ &- E\left[\frac{Q}{4\pi k_1} exp\left(-\frac{y_1^2}{4k_1}\right) \cdot 2 \cdot exp\left(-\frac{H^2}{4k_1}u\right)\right] \times \\ &- E\left[\frac{Q}{4\pi k_2} exp\left(-\frac{y_2^2}{4k_2}\right) \cdot 2 \cdot exp\left(-\frac{H^2}{4k_2}u\right)\right] \times \end{aligned}$$

Let $S_1 = -\frac{y_1^2}{4k_1}$, $S_2 = -\frac{y_2^2}{4k_2}$, $t_1 = -\frac{H^2}{4k_1}$, $t_2 = -\frac{H^2}{4k_2}$, Hence, given the probability distribution of random variable Q and u, we can calculate any two point in the coordination.

$$E\left[\frac{Q^2}{4\pi^2 k_1 k_2} exp(-u\left(S_1 + S_2 + t_1 + t_2\right))\right] - E\left[\frac{Q}{2\pi k_1} exp(-u\left(S_1 + t_1\right))\right] \cdot E\left[\frac{Q}{2\pi k_2} exp(-u\left(S_2 + t_2\right))\right] \tag{14}$$

Now, given the group of sensor be chosen, we can calculate the total number.

C. Finding the optimization solution

Following is some steps of finding the optimization solution:

1) According to the (8), we can calculate the smallest transmission time for each sensor, $t_{mi} : t_m = t_{m1}, t_{m2}, ..., t_{mN}$
2) Initialize a vector of probability $p_{1\times N} = [1/2, 1/2, ..., 1/2]$, for each entry in this vector is mapping to the probability of each node being chosen.
3) Generate $8N$ binary sequences eg. $x = [1, 0, ..., 1]$, where 1 represent the node will be chosen, and 0 represent will not be chosen.
4) For each binary sequence $x^{(k)}$, if all the node be chosen satisfy $\Sigma_i x_i^(k) t_{mi} \leq T$, the sequence is a feasible solution. Otherwise, dropping the chosen node which needs the longest transmission time ($x_j = 0$) until this sequences $\Sigma_i x_i^(k) t_{mi} \leq T$. Operating all the $8N$ sequences, we can change these sequences into feasible solution.
5) Calculate each objective $\{H_1, H_2, ..., H_{8N}\}$, and sorting these sequences from the largest to the smallest $\{H_{l(1)}, H_{l(2)}, ..., H_{l(8N)}\}$, where l is the index of origin sequence. Maintain two variables HM, S to memorize the largest objective and the best solution sequences. Note that if the largest objective is larger than the previous one (HM), replace the values, $HM = H_{l(1)}$ and $S = x^{l(1)}$.
6) Choose the first 20% sequences, and update the probability vector p according to the following equation:
$$p = \alpha \cdot p + \frac{1-\alpha}{r} \sum_{j=1}^{r} x^I j \tag{15}$$
7) Repeat step 3 to 6 until every element in p vector converge to smaller than 0.05 or larger than 0.95 (usually need 30 100 iterations), and get the optimize solution.

D. Brief description

Our goal is to minimize the mean square error(MSE) between the reconstruct map and the origin field, this is not exactly the same as getting the most entropy uploading solution from cross entropy solution. Hence, we need to construct relation between optimized solution and smallest MSE.

c) Terminology: We use the received data $\hat{X}_S$ to reconstruct the whole data X_V as $\tilde{X_V}$ Note that since we introduced no extra information during the reconstructing process, if the reconstruction can be viewed as a kind of mapping function $f : \hat{X}_S \to \tilde{X}_V$ it can be shown that:

$$I\left(X_V; \hat{X}_S\right) \geq I\left(X_V; \tilde{X}_V\right) \tag{16}$$

Moreover, we can further illustrate

$$\begin{aligned} I\left(X_V; \hat{X}_S\right) &\geq \left(X_V; \tilde{X}_V\right) \\ &= h\left(X_V\right) - h\left(X_V | \tilde{X}_V\right) \\ &\geq h\left(X_V\right) - \sum_{i=1}^{N} h\left(X_i | \tilde{X}_V\right) \\ &= h\left(X_V\right) - \sum_{i=1}^{N} h\left(X_i - \tilde{X}_i | \tilde{X}_V\right) \\ &\geq h\left(X_V\right) - \sum_{i=1}^{N} h\left(X_i - \tilde{X}_i\right) \\ &\geq h\left(X_V\right) - \sum_{i=1}^{N} h\left(N\left(0, E\left(X_i - \tilde{X}_i\right)^2\right)\right) \\ &\geq h\left(X_V\right) - \sum_{i=1}^{N} log_2 \frac{1}{2} log_2\left[\left(2\pi e\right) MSE_i\right] \\ &= h\left(X_V\right) - \frac{1}{2}\left[\left(2\pi e\right)^N \prod_{i=1}^{N} MSE_i\right] \\ &\geq h\left(X_V\right) - \frac{1}{2}\{\left(2\pi e\right)^N \left[\frac{1}{2}\sum_{i=1}^{N} MSE_i\right]^N\} \end{aligned}$$

$$h\left(X_V\right) - \frac{1}{2}log_2\left[\left(2\pi e\right)^N MSE_V^N\right] \tag{17}$$

Substituting

$$h\left(X_V\right) = \frac{1}{2}log_2\left[\left(2\pi e\right)^N \left|\Sigma_V\right|\right] \tag{18}$$

Then we obtain

$$I_S = I\left(X_V; \hat{X}_S\right) \geq \frac{1}{2}log_2\left(\frac{\left|\Sigma_V\right|}{\left(MSE_V\right)^N}\right) \tag{19}$$

Note that observing $\hat{X}_V$ from X_V gives no more information. Hence, $h\left(\hat{X}_S|X_V\right) = 0$, then

$$\begin{aligned} I\left(X_V; \hat{X}_S\right) &= h\left(\hat{X}_S\right) - h\left(\hat{X}_S|X_V\right) \\ &= h\left(\hat{X}_S\right) \end{aligned} \tag{20}$$

Now, we can conclude that finding the maximum entropy can give a lower bound of mean square error as the following relation, which means that the more entropy we can achieve the more possible to make the MSE smaller:

$$h\left(\hat{X}_S\right) \geq \frac{1}{2}log_2\left(\frac{\left|\Sigma_V\right|}{\left(MSE_V\right)^N}\right) \tag{21}$$

$$MSE_V \geq \left(\frac{\left|\Sigma_V\right|}{2^{2h\left(\hat{X}_S\right)}}\right)^{\frac{1}{N}} \tag{22}$$