Contents lists available at ScienceDirect

Computers & Geosciences

journal homepage: www.elsevier.com/locate/cageo

# Monte Carlo source detection of atmospheric emissions and error functions analysis

Guido Cervone [a,b,*], Pasquale Franzese [b]

[a] Department of Geography and Geoinformation Science, George Mason University, 4400 University Dr., Fairfax, VA 22030, USA
[b] Center for Earth Observing and Space Research, George Mason University, 4400 University Dr., Fairfax, VA 22030, USA

ARTICLE INFO

*Article history:*
Received 11 June 2009
Received in revised form
24 December 2009
Accepted 19 January 2010

*Keywords:*
Monte Carlo
Atmospheric emissions
Source detection
Dispersion model
Error functions

ABSTRACT

A Monte Carlo algorithm is iteratively run to identify candidate sources for atmospheric releases. The values of the ground measurements of concentration are synthetically generated by a benchmark simulation of a Gaussian dispersion model. At each iteration, a Gaussian reflected plume model is applied to compute the dispersion from a candidate source, and the resulting concentrations are compared with the measurements at fixed points on the ground. Iterative algorithms for detection of atmospheric release sources are based on the optimization of an error function between numerical simulations and observations. However, the definition of error between observations and simulations by an atmospheric dispersion model is not univocal. In this paper, the comparisons are made using various error functions. The characteristics of different error functions between model predictions and sensor measurements are investigated, with a statistical analysis of the results. Sensitivity to domain size and addition of random noise to the measurements are also investigated.

## 1. Introduction

Airborne toxic contaminants are transported by the wind and dispersed by atmospheric turbulence. Potential atmospheric hazards include industrial chemical spills, forest fires, intentional or accidental releases of chemical, biological or radiological agents. Risk assessment of contamination from a known source can be performed by running multiple forward transport and dispersion numerical simulations for different meteorological conditions, and by analyzing the simulated contaminant clouds with clustering and classification algorithms to generate probabilistic risk maps (Cervone et al., 2008). However, often the source is unknown, and must be identified from limited concentration measurements observed on the ground.

There are currently no established methodologies for detecting the sources of atmospheric releases, and there is a great degree of uncertainty towards the effectiveness and applicability of existing techniques. One line of research focuses on the adjoint transport modeling (Pudykiewicz, 1998; Hourdin and Issartel, 2000), where the advection–diffusion equation is inverted to establish source–receptor relationships, which are then used to perform backward simulations of the particles from the receptor to the source. These methods assume a steady-state system and linear processes, i.e., do not include dispersion of chemically reactive agents (Enting, 2002). The optimization techniques employed to solve the problem of inverse simulation often give a single solution, or assume a Gaussian distribution to account for uncertainties. In general, these techniques do not perform satisfactorily in the cases of scarce and ill-distributed data, or of large amounts of data from different sensors.

A more general and powerful methodology for source detection is based on Bayesian inference coupled with stochastic sampling (Gelman et al., 2003). Bayesian methods aim at an efficient ensemble run of forward simulations, where statistical comparisons between simulated and observed data are used to improve the estimates of the unknown source location (Chow et al., 2006). The Bayesian approach is independent of the type of model used, the type and amount of data, and can be applied to non-linear processes as well. Senocak et al. (2008) use a Bayesian inference methodology to reconstruct atmospheric contaminant dispersion, pairing the Bayesian paradigm with Markov-chain Monte Carlo (MCMC) to iteratively identify potential candidate sources. A reflected Gaussian plume model is run for each candidate source, and the resulting concentrations are compared to ground observations. The goal of the algorithm is to minimize the error between simulated and measured concentrations. Delle Monache et al. (2008) paired the Bayesian inference process with a Metropolis-Hasting MCMC. The dispersion simulations are conducted using a Lagrangian particle dispersion model. The methodology is applied to find the source characteristics for the

[*] Corresponding author at: Department of Geography and Geoinformation Science, George Mason University, 4400 University Dr., Fairfax, VA 22015, USA. Tel.: +1 703 993 1799; fax: +1 703 993 1865.
*E-mail addresses:* gcervone@gmu.edu (G. Cervone), pfranzes@gmu.edu (P. Franzese).

0098-3004/$ - see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cageo.2010.01.007

1988 accidental release of radioactive material at Algeciras, Spain, using the available ground concentrations measurements and meteorological observations. The excellent results obtained by Delle Monache et al. (2008) show the operational potential of stochastic algorithms to identify the source characteristics of an atmospheric release.

Haupt (2005), Haupt et al. (2007), and Allen et al. (2007) use an evolutionary algorithms to maintain a population of candidate solutions from multiple forward dispersion simulations, which converge iteratively toward the real source. These algorithms are also model- and domain-independent, but unlike traditional Monte Carlo methods they maintain a population of candidate solutions. They are designed to try to avoid local optima, and are therefore particularly useful in problem domains with complex fitness landscapes such as the source detection problem.

The statistical comparison between simulated and observed concentrations which drives the iterations from tentative sources is based on the definition of error, or uncertainty, in the model predictions. The accuracy of a transport and dispersion simulation depends on a number of factors, such as the scale of the phenomenon, the accuracy of the source term, the availability and representativeness of meteorological data, the coverage of the sensor network and the averaging times of its measurements, and the approximations inherent in the numerical model in order to perform the simulation in a realistic time frame and with a realistic data storage capability. Even in controlled field experiments, model simulations can at best give only an approximate representation of the evolution of a released contaminant. Several measures of accuracy of a dispersion calculation are available (Hanna et al., 1993). In some cases, only the peak concentration over the sensor network is compared with the simulated peak concentration, regardless of the time and location of occurrence of the peak. In other cases, the only realistic expectation is to calculate the predictions that fall within a factor 2 of the observations, or even a factor 10. In order to determine whether some model predictions are closer to the observations than others, several performance metrics need to be considered (Chang et al., 2003; Chang and Hanna, 2004).

The goal of this paper is an investigation of several error functions that can be used to drive the iterative processes. Each error function is a different definition of error between simulated and observed concentration data at the receptors. We investigate the responses of several error functions in a controlled environment, to ultimately help to identify the measures most suitable for efficient optimizations. We use a Monte Carlo based algorithm because of its simplicity and regularity, which allows to better isolate the behavior of the different functions and to compute statistics of their convergence (Metropolis and Ulam, 1949; Hammersley and Handscomb, 1964; Cervone et al., 2000; Robert and Casella, 2004; Rubinstein and Kroese, 2007; Delle Monache et al., 2008).

Numerical experiments simulate domains of varying size, with an increasing number of sensors, to assess the performance of each function under different conditions. Different measures used by the atmospheric modeling community to match simulations with measurements were considered and analyzed varying the domain size and the number of sensors, and adding artificial noise to simulate the uncertainty associated with sensors measurements. The algorithm has been tested with data from a release conducted during the Prairie Grass field experiments of dispersion (Barad and Haugen, 1958).

The paper is structured as follows: Section 2 discusses the methodology used, including the different measures used, and the stochastic search algorithm employed; Section 3 reports the results of the simulations performed, and the statistical assessment of the performance for each measure used; results for the test run against the Prairie Grass data are also reported. Finally, Section 4 summarizes the results obtained.

## 2. Methodology

The proposed methodology is based on numerical transport and dispersion simulations, ground measurements, and a stochastic Monte Carlo algorithm. We assume that the observations of the actual concentrations of the atmospheric release at fixed points in the space are available. A stochastic Monte Carlo algorithm is used to iteratively refine an initial random guess by varying the $x_0$, $y_0$ and $z_0$ coordinates of the best solution found. The score for each candidate solution is a metric of how well the simulated concentrations match the synthetic observations. The analysis of the metrics obtained from the different functions, which indicate their performance, is the main goal of the paper.

### 2.1. Dispersion simulations

The dispersion simulations were performed using a reflected three-dimensional Gaussian dispersion model (Arya, 1999). The mean concentration $c$ at a point $(x, y, z)$ from a source located at $(x_0, y_0, z_0)$ is defined by

$$c=\frac{Q}{2\pi U\sqrt{(\sigma_{y_0}^2+\sigma_y^2)(\sigma_{z_0}^2+\sigma_z^2)}}\exp\left[-\frac{(y-y_0)^2}{2(\sigma_{y_0}^2+\sigma_y^2)}\right]$$
$$\times\left\{\exp\left[-\frac{(z-z_0)^2}{2(\sigma_{z_0}^2+\sigma_z^2)}\right]+\exp\left[-\frac{(z+z_0)^2}{2(\sigma_{z_0}^2+\sigma_z^2)}\right]\right\} \quad (1)$$

where $x$, $y$ and $z$ are the coordinates in the alongwind, cross-wind and vertical directions, respectively, $U$ is the mean wind speed, $\sigma_{y_0}$ and $\sigma_{z_0}$ are the cross-wind and vertical source sizes. The dispersion coefficients $\sigma_y$ and $\sigma_z$ were chosen assuming slightly unstable atmospheric stability (Pasquill's 'C' stability class) and using Briggs' curves for rural conditions (Arya, 1999):

$$\sigma_y=0.11(x-x_0)[1+0.0001(x-x_0)]^{-1/2} \quad (2)$$

$$\sigma_z=0.08(x-x_0)[1+0.0002(x-x_0)]^{-1/2} \quad (3)$$

Part of the difficulties in a source detection algorithm originate from the large differences in concentration measurements around and downwind of the source, compared to off-centerline locations. Since far from the centerline the concentration gradient is small, local search methods might converge on local optima failing to identify the global optimum solution, especially for non-monotonic gradient or in the presence of errors. The error function selected may play an important role in the correct convergence toward the global optimum, as well as in the rate of convergence.

### 2.2. Error functions

The quantitative comparison of synthetic observations and predicted concentration is performed by applying statistical measures of error. Each measure reflects different aspects of the spatial distribution of mean concentration, and the appropriate use of one measure rather than another depends largely on the characteristics of the concentration field. The importance of the information provided by a specific error measure is determined by factors such as the number of sensors, the number of outliers, the range of measured and simulated values. We considered the following functions: the fractional bias FB, the normalized mean square error NRMSE, the geometric variance VG and the correlation function CORR which have been often used in model validation exercises (e.g., Hanna et al., 1993; Chang et al., 2003), the mean square error RMSE, two new functions defined in Haupt et al. (2006) and Allen et al. (2007) which are referred to in this paper as AHY1, AHY2, and a modified version of AHY2 which is designated as AHY2MOD:

$$\mathrm{FB} = 2\left|\frac{\overline{C_o} - \overline{C_s}}{\overline{C_o} + \overline{C_s}}\right| \tag{4}$$

$$\mathrm{NRMSE} = \sqrt{\frac{\overline{(C_o - C_s)^2}}{\overline{C_o}\ \overline{C_s}}} \tag{5}$$

$$\mathrm{VG} = \exp[\overline{(\log c_o - \log c_s)^2}] \tag{6}$$

$$\mathrm{CORR} = 1 - \frac{\overline{(C_o - \overline{C_o})(C_s - \overline{C_s})}}{\sigma^2_{C_o}\sigma^2_{C_s}} \tag{7}$$

$$\mathrm{RMSE} = \sqrt{\frac{\overline{(C_o - C_s)^2}}{\overline{C_o}^2}} \tag{8}$$

$$\mathrm{AHY1} = \sqrt{\frac{\overline{(C_o - C_s)^2}}{\overline{C_o^2}}} \tag{9}$$

$$\mathrm{AHY2} = \sqrt{\frac{\overline{[\log_{10}(C_o + 1) - \log_{10}(C_s + 1)]^2}}{\overline{[\log_{10}(C_o + 1)]^2}}} \tag{10}$$

$$\mathrm{AHY2MOD} = \sqrt{\frac{\overline{(\log_{10} C_o - \log_{10} C_s)^2}}{\overline{(\log_{10} C_o)^2}}} \tag{11}$$

where $c_o$ and $c_s$ are the observed and simulated concentration at the sensors, respectively.

FB is a normalized mean bias, therefore it is mainly a measure of the systematic over- or underprediction of the simulations with respect to measurements. FB is not a local measure. Very low values of FB indicate good quantitative agreement in terms of overall mean concentration, but provide no information on the spatial accuracy of the simulation, as observed and simulated values are not paired in space. For example, the simulation could be out of phase with the observations. FB is strongly influenced by occurrences of large over- or underprediction of high concentration measurements.

NMSE and VG are both expressed in terms of variances, reflecting both systematic bias and relative random errors, which are estimated on a linear scale by NMSE and on a logarithmic scale by VG. NMSE is strongly affected by infrequently occurring large overprediction or large observed outliers. By contrast, VG is rather insensitive to both high and low values, and is in general preferred when both predicted and observed values span several orders of magnitude. However, note that VG is strongly affected by extremely low values and is undefined for zero values, which are not uncommon in dispersion modeling. Therefore, zero and near zero values require a special treatment when evaluated using VG. For example,the limit of detection of the samplers can be assumed as minimum threshold. CORR is the standard correlation function, which is based on mean concentration and variance.

We also analyzed the functions AHY1 and AHY2, defined in Haupt et al. (2006) and Allen et al. (2007) as metrics for the cost function of a genetic algorithm for source detection. AHY1 is a variation of RMSE with a different normalization coefficient; in AHY1 the normalization is over the average of $c_o^2$, whereas in RMSE the normalization is over the square of the average $c_o$. AH2MOD is a variation of AHY2, where the logarithm is taken over the observed value, rather than observed value plus one. The main difference is the value of the function value for very small observed concentrations.

### 2.3. Monte Carlo

A specific Monte Carlo algorithm was implemented to identify the optimal $x$,$y$ and $z$ location of the unknown source. Monte Carlo methods are a class of computational algorithms that rely on repeated random sampling to optimize a process.

The algorithm starts by generating a single candidate solution (a.k.a. initial guess) within the boundaries of the search space, and evaluates it according to one of the error functions defined in Section 2.2. The iterative search continues by applying random changes to the best solution found and evaluating the new solution. Specifically, at each iteration the $x$, $y$ and $z$ variables of the best solution are modified by a random increment. If the new solution has a better score (i.e., a lower error) as determined by the error function, it replaces the current best. The process continues until a termination condition is met. The termination condition was based on a fixed number of iterations. It is possible to use different criteria such as, e.g., a fixed elapsed time, or a solution with a given minimum error.

One of the central parameters governing the behavior of the Monte Carlo search is the step-size random increment which is used to generate new candidate solutions. The implemented algorithm uses an adaptive step-size based on a fourth-order polynomial curve. A different step-size is used for each of the three dimensions $x$, $y$ and $z$, as they have different domain ranges, and require independent adjustments. The search starts with a step-size equal to one-half of the domain for the particular variable, and decreases as new candidate solutions are generated. If after ten consecutive new solutions no improvement is made, the step-size is set to increase for the future iterations. The step-size then switches back and forth between increasing and decreasing depending on the progress of the algorithm. The rationale behind this behavior can be explained as switching between exploration and exploitation of the search space. In the first phases exploration of the space for potentially good candidate solutions is important in order not to remain localized to the area surrounding the initial guess. As the search progresses, the algorithm refines the solution by making smaller adjustments. If no new progress is detected, then either the perfect solution has been found, or the algorithm has converged to a local optimum. In order to sample new areas of the space, and potentially leave the area of local optimum, larger steps are required and are possible due to the adaptive step-size.

The Monte Carlo method was implemented using the R (www.r-project.org) statistical environment. A typical run of 1000 iterations completes in under 4 min. Experiments were performed on an Intel(R) Core(TM) 2 Duo CPU E8400 running at 3.00 GHz, with a Linux operating system.

## 3. Results

The experiments were carried on four different domains defined by their cross-wind extension $W$, along-wind extension $L$, and vertical height $H$. The domain areas ranged from a minimum of 0.5 km$^2$ to a maximum of 200 km$^2$. The same height $H = 100$ m was used for all domains. For all simulations, $Q = 0.1$ kg s$^{-1}$ and $U = 2$ m s$^{-1}$.

First a synthetic concentration field was generated by running the Gaussian model defined in Eq. (1) from a known source. A second synthetic concentration field was generated by adding noise to the original field as $c_i' = gc_i$, where $g$ is a uniform random process between 0 and 2, and $c_i$ is the original synthetic

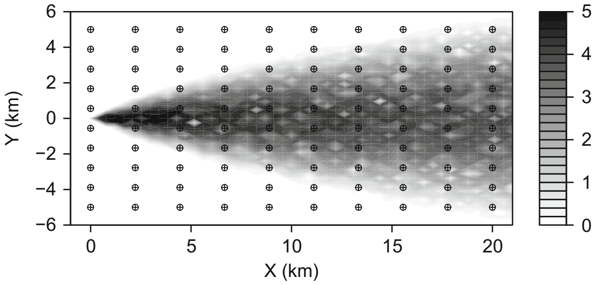

**Fig. 1.** Concentration field generated by Eq. (1) for $Q = 100\,\mathrm{g\,s^{-1}}$ and $U = 2\,\mathrm{m\,s^{-1}}$, for a source with size $\sigma_{y_0} = \sigma_{z_0} = 10\,\mathrm{m}$, and with superimposed random noise for a domain of $200\,\mathrm{km}^2$ with 100 sensors indicated by crossed circles. Side scale represents logarithm of concentration.

**Table 1**
Summary of domain configurations used.

| $W$ (km) | $L$ (km) | $H$ (km) | Number of sensors |
|---|---|---|---|
| 0.5 | 1 | 0.1 | 4, 9, 16, 25, 100, 900, 2500 |
| 1 | 2 | 0.1 | 4, 9, 16, 25, 100, 900, 2500 |
| 5 | 10 | 0.1 | 4, 9, 16, 25, 100, 900, 2500 |
| 10 | 20 | 0.1 | 4, 9, 16, 25, 100, 900, 2500 |

concentration at the sensor $i$. Fig. 1 shows the synthetic concentration field generated by Eq. (1) for a source $\sigma_{y_0} = \sigma_{z_0} = 10\,\mathrm{m}$, and with superimposed random uniform noise for an area of $200\,\mathrm{km}^2$ with 100 sensors, represented by crossed circles. The synthetic concentrations are an idealized representation of an observed concentration field.

The source detection procedure consists of the following steps:

- Run the dispersion model (1) from a tentative source location.
- Compare the calculated concentration field with the synthetic field at the sensor locations using one the error functions defined in Section 2.2.
- Compare the current value of the error function with its value at the previous iteration.
- Generate a new candidate source location by modifying the best solution found.

Several sets of simulations were run with a different number of sensors deployed in the domain, ranging from 4 to 2500. Details of the individual domain configurations are reported in Table 1. The number of sensors is important as they effectively determine the amount of information available to the Monte Carlo algorithm. It is therefore interesting to investigate the sensitivity of the method to the number of sensors, and possibly to establish a minimum threshold. The unknown variables optimized by the Monte Carlo simulation are the $x_0$, $y_0$ and $z_0$ coordinates of the source. The additional parameters $Q$, $\sigma_{y_0}$, $\sigma_{z_0}$ and $U$, needed to run the model defined in (1) are assumed to be known.

A total of 8960 Monte Carlo simulations were performed to study the effect of domain size, sensor density and noise. For each combination of parameters, 20 simulations were performed to analyze the robustness of the method, only varying the initial random guess. By repeating the experiment multiple times with different initial guess, it is possible to determine the overall convergence of the algorithm, and its sensitivity to the initial guess. The convergence is defined as the rate at which the calculated solution approaches the real solution. The experiments have been kept as ideal as possible in order to test the effects of different error measures without the introduction of other factors.

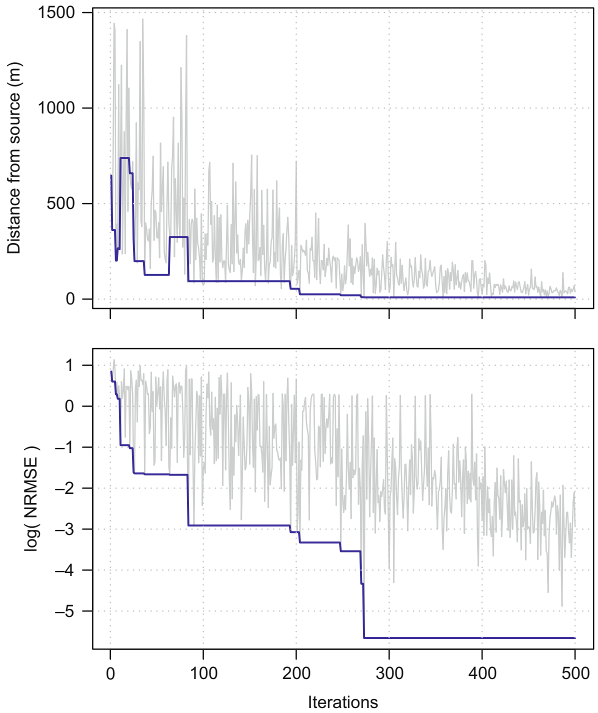

**Fig. 2.** Distance from source (in meters), and NRMSE error function (in log scale) for Monte Carlo simulation as functions of number of iterations.

Fig. 2 shows the execution of a single Monte Carlo simulation. The bottom figure shows the error at each Monte Carlo iteration according to the NRMSE function, which is the error function selected to drive the optimization. The dark line represents the minimum error obtained over all the previous iterations. The top figure shows the evolution of the calculated distance from the source as a function of the Monte Carlo iterations. The dark line represents the evolution of the distance corresponding to the minimum error found. Note that the distance does not decrease monotonically, because smaller errors do not always correspond to better solutions. This phenomenon is due to the three-dimensional nature of the problem, where the $x_0$, $y_0$ and $z_0$ are simultaneously optimized, leading to a multi-modal solution space. This implies that some solutions can have a smaller error compared to solutions closer to the source. The curve for the best solution found is consistent with the overall convergence trend of the algorithm. We have found that the initial random guess plays a major role in the convergence rate of the algorithm. If the initial random guess is located in proximity of the boundaries of the domain, the rate of convergence is much slower. Additionally, if the random initial solution is generated in proximity of local optima, the algorithm might fail to converge over the best solution possible. This is one of the main disadvantages of the Monte Carlo algorithm used in the experiments.

Fig. 3 shows an example of a trajectory converging toward the best solution determined by the Monte Carlo algorithm using the NRMSE error function. The figure illustrates the typical behavior including an initial fast convergence in the $x$–$y$ plane, followed by slower and steady improvements in the $z$ dimension. The contour lines of the error surface for the NRMSE function, and the projection of the trajectory on the $x$–$y$ plane are also shown.

Experiments were performed to study the rate of convergence and accuracy of the solution associated with:

- Different error functions.
- Domain size. (The vertical dimension has been kept constant.)

- Number of sensors (density of sensors). For practicality, the sensors have been assumed to be uniformly distributed throughout the domain.
- Noise, by introducing a random error to the sensor measurements.

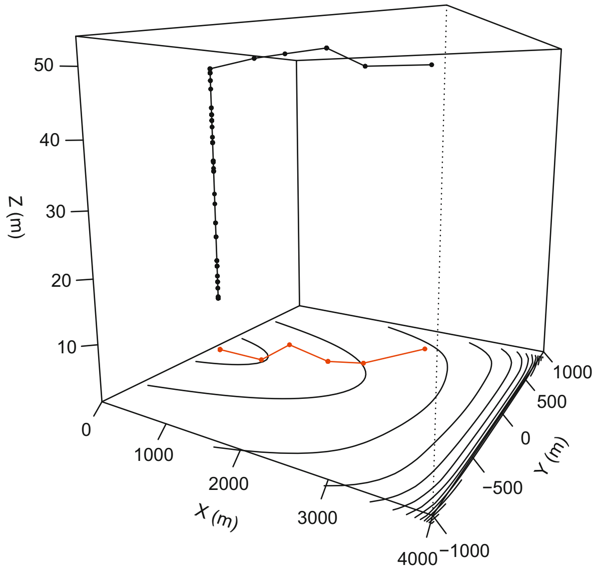

**Fig. 3.** Sample trajectory converging toward best solution ($x_0$=0, $y_0$=0) determined by Monte Carlo algorithm using NRMSE error function. Contour lines of NRMSE error surface, and projection of 3D trajectory onto $x$–$y$ plane are also shown.

Each experiment has been repeated with and without random uniform error added to the synthetic ground measurements.

Fig. 4 shows the results for all sensor configurations tested for a 0.5 km$^2$ domain (a and b) and a 200 km$^2$ domain (c and d). The figure shows the statistical results for all the experiments performed, reported using boxplots (Mosteller and Tukey, 1977). Each individual boxplot is the summary for the series of experiments (20 runs). In each boxplot the horizontal black line shows the median error achieved in the series, and the top and bottom of the box are the 25th and 75th percentile (the lower and upper quartiles) respectively. The top and bottom error bars (a.k.a. whiskers) are the minimum and maximum errors achieved in the series of experiments. Such plots are particularly useful to compare distributions between sets of experiments. The figures on the left column (a and c) show results for the synthetic concentration field without added noise, the right column (b and d) shows results with added noise. The results indicate that the presence of noise changes drastically the convergence rate of the algorithm.

In particular, we found that for the case of *small domain without noise*, all error functions determine comparable results. All functions result in large errors (in terms of source location) when four or nine sensors are used; the error is significantly smaller for the cases of 16 or more sensors. Furthermore, little difference in the convergence rate is observed with more than 16 sensors. In other words, using more than 16 sensors improves neither the accuracy of the predicted source location, nor the computational efficiency of the iterative process. Experiments for *large domain without noise* are similar, with the exception of the cases of very few sensors for the CORR, AHY2 and AHY2MOD methods. The FB method is outperformed by all other methods. For the case of *small domain with noise*, the number of sensors has a major influence on the convergence rate. In all cases, increasing

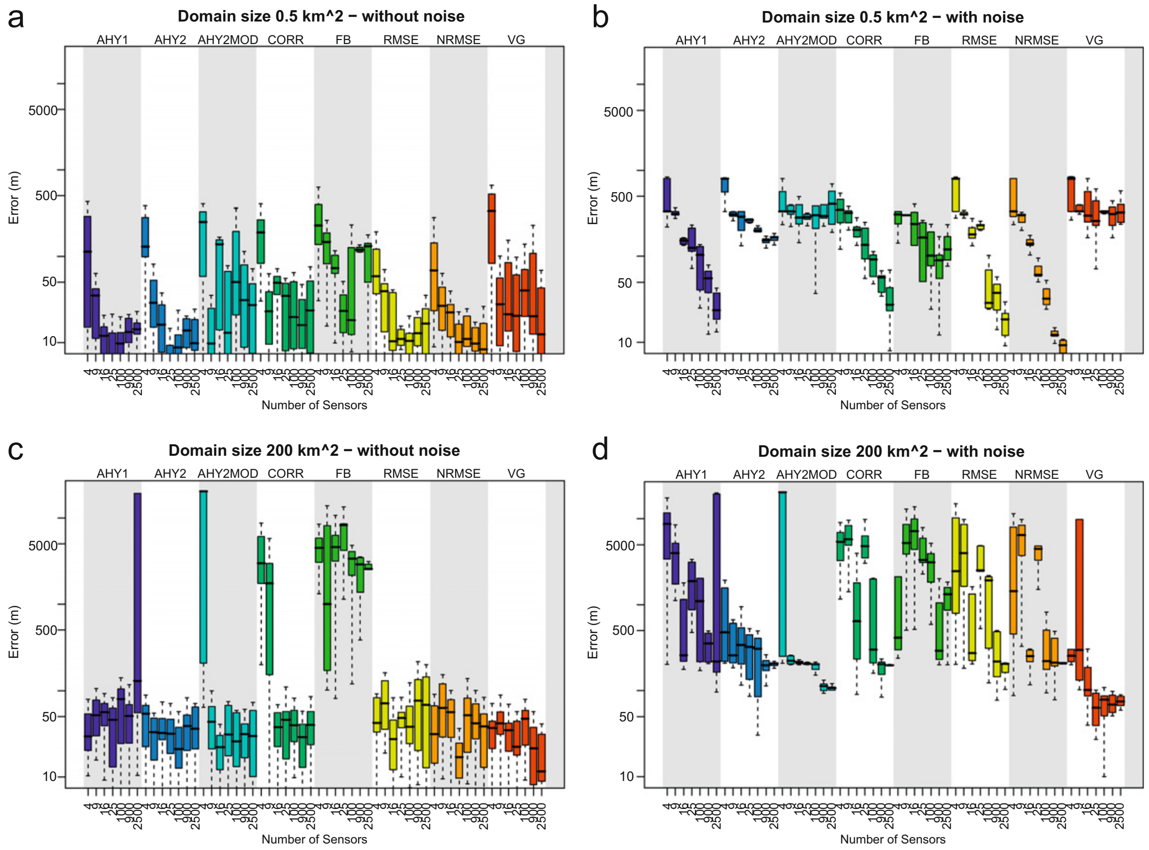

**Fig. 4.** Error as a function of domain size: (a) domain size of 0.5 km$^2$ without noise; (b) domain size of 0.5 km$^2$ with noise; (c) domain size of 200 km$^2$ without noise; and (d) domain size of 200 km$^2$ with noise.

the number of sensors used in the experiments up to a maximum of 2500 leads to significant improvement of the solution. VG and AHY2MOD are consistently outperformed by all other error functions, and are both quite insensitive to an increase in the number of sensors. Note that the large errors for AHY1 show a median which is consistent with the overall pattern, but a long bar due to only a few experiments not having converged. In all cases the source location is predicted with an accuracy of about 100 m.

For the case of *large domain with noise*, the number of sensors does not have a major influence on the converge rate, with the exception of the VG function. The experiments performed using the VG function are also the only ones able to find a solution within 50 m of the source.

Fig. 5 shows the results for all domain sizes tested for 25 sensors (a and b) and 2500 sensors (c and d). As in Fig. 4, the graphs on the left column (a and c) show results for the synthetic concentration field without added noise, the right column (b and d) shows results with added noise.

We determined that increasing the size of the domain does not play a major role in the convergence rate of the algorithm. In particular, we found that for 25 *sensors without noise* all functions performed similarly regardless of the domain size, with the exception of FB, whose results are significantly worse as the domain size increases. For the case of 2500 *sensors without noise* an almost identical behavior is observed, with FB performing consistently worse than any other methods. For 25 *sensors with noise* an overall worse performance is found, with very few cases converging within 100 m of the source. All functions except VG are sensitive to the domain size, with errors up to 5 km for the largest domain. For 2500 *sensors with noise* the overall error increases with the domain size.

Although most functions exhibit the same trends, some outperformed others under specific conditions. NRMSE and AHY2 are good overall performers, achieving good results both under very large and small domains, regardless of the presence of measurement errors. VG outperforms all other functions for large domains, especially with noisy observations. However, it is not a good performer for small domains. The FB function is consistently one of the worst performers throughout the experiments. This is because it takes into account only the distribution of concentrations, rather than their relative locations.

In a few instances, the general trend was disrupted by at least one of the experiments not converging, or converging towards a local optimum. This is a major weakness of the proposed stochastic approach, and the reason why each experiment was performed multiple times. Overall the method performed well, and the statistical analysis of the convergence shows a predictable behavior.

A better understanding of the behavior of the error functions can be gained by performing an analysis of their geometric properties. We generated error function surfaces by successively placing a tentative source at each point of the domain, and calculating the corresponding value of the error function. The set of values of the error function at all points forms a surface, whose contour lines at $z=0$ are plotted in Fig. 6. All functions were normalized between 0 and 1. The real source is located at $(x_0, y_0, z_0)=(0, 0, 10)$.

- In general, at large distance from the source, the along-wind gradient is stronger than the cross-wind gradient. Close to the

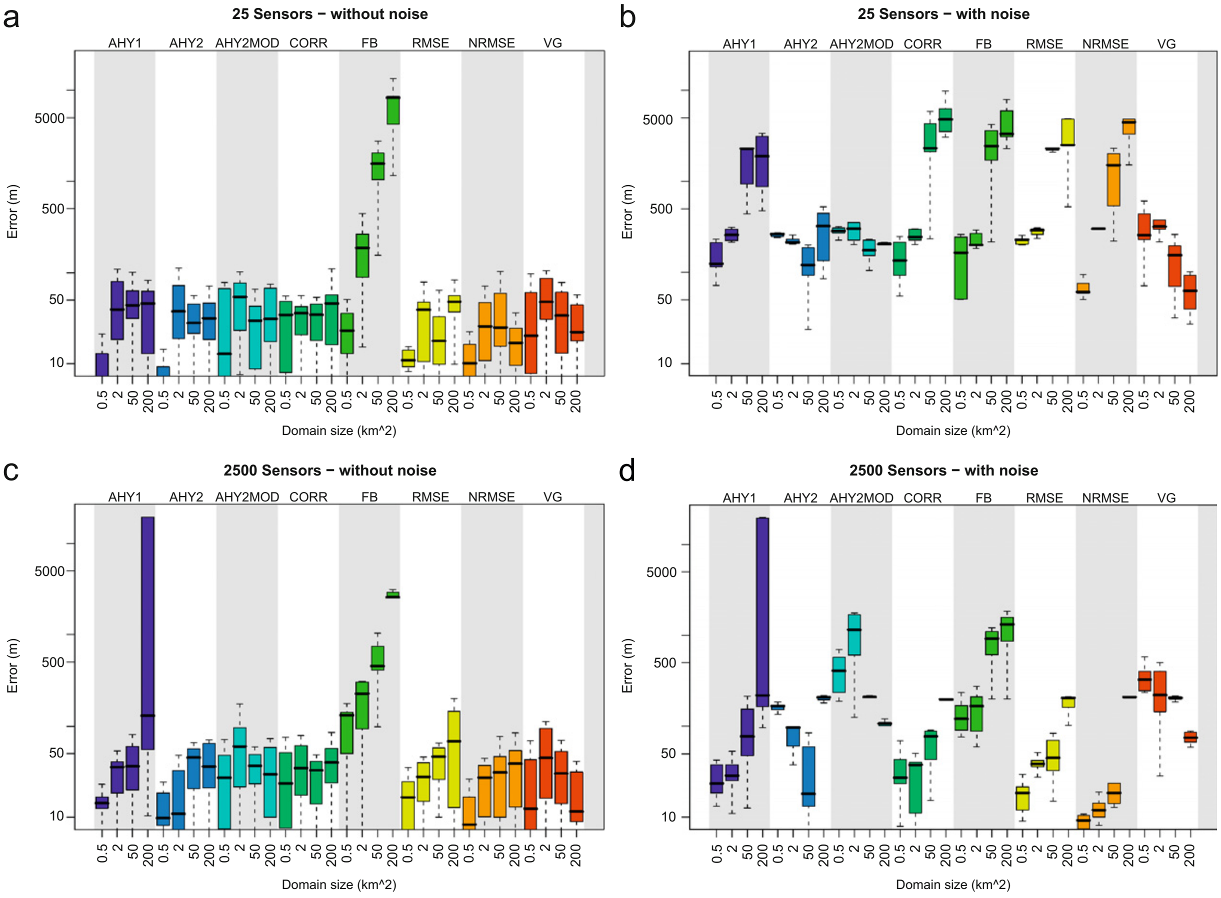

**Fig. 5.** Error as a function of sensor density: (a) 25 sensors without noise; (b) 25 sensors with noise; (c) 2500 sensors without noise; and (d) 2500 sensors with noise.

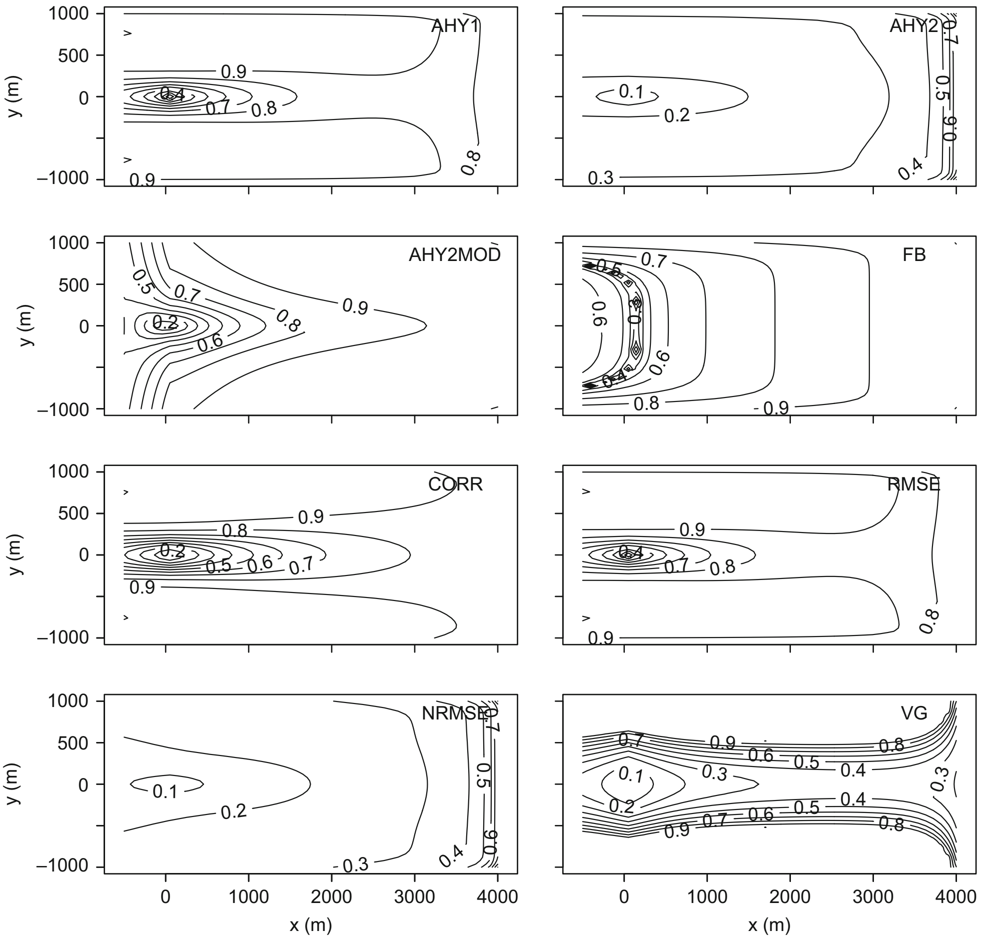

**Fig. 6.** Contour lines of error functions AHY1, AHY2, AHY2MOD, FB, CORR, RMSE, NMQSE, and VG.

source the along-wind gradient is stronger only for AHY2MOD and FB.

- AHY1 and RMSE are virtually identical. This is due to the two functions differing only in the normalization coefficient. The absolute values for the functions are indeed different.
- AHY1, RMSE, CORR and FB display local maxima and minima. This can cause the algorithm to converge to an incorrect solution. More sophisticated search techniques, such as standard evolutionary algorithms (e.g., Goldberg, 1989) or non-Darwinian evolutionary algorithms (Cervone et al., 2000, 2010) which perform simultaneous parallel searches, are less likely to converge to local optima.
- NRMSE and AHY2 show a very distinct area of minima around the source, and a sharp gradient in every direction, especially in the cross-wind direction. Further analysis showed that the Monte Carlo algorithm quickly converges towards the right $y$ location, while the optimization along $x$ requires several additional iterations.

### 3.1. Comparison with a field experiment

The method was applied to find the source location of a gas released in controlled conditions during the 1956 Prairie Grass field experiment (Barad and Haugen, 1958). Concentrations of $SO_2$ trace gas released at ground level for 10 min were measured by samplers radially located at distances of 50, 100, 200, 400 and 800 m. The Monte Carlo method was used to find the source location for release 43, which occurred in slightly unstable atmosphere (i.e., 'C' stability class). The error functions defined in (4)–(11) were used. Table 2 shows the results in terms of minimum distance from the source achieved, number of iterations required to achieve the minimum distance, and their respective standard deviations. Each series of experiments was performed 20 times varying only the initial guess. The minimum distance was found using the AHY1 function, although with a higher standard deviation. The smallest standard deviation for the distance was observed using VG. As already observed for the synthetic case, the FB function was the overall worst performer. In terms of number of iterations, all functions performed comparably, with slightly better results achieved using the AHY2MOD function, and slightly worse using the RMSE function.

**Table 2**
Mean and standard deviation of distance from source, along with number and standard deviation of iterations obtained for Prairie Grass experiment 43.

| | RMSE | NRMSE | FB | CORR |
|---|---|---|---|---|
| Distance from source (m) | 6.8 ± 1.8 | 6.5 ± 2.3 | 8.0 ± 2.3 | 6.9 ± 2.5 |
| Iterations | 281 ± 70 | 254 ± 96 | 256 ± 130 | 255 ± 116 |
| | VG | AHY1 | AHY2 | AHY2MOD |
| Distance from source (m) | 6.9 ± 1.3 | 5.5 ± 2.6 | 6.8 ± 2.8 | 7.4 ± 2.2 |
| Iterations | 275 ± 88 | 275 ± 130 | 257 ± 79 | 235 ± 120 |

## 4. Conclusions

We implemented a specialized Monte Carlo algorithm for the detection of the source of a localized continuous emission of an atmospheric pollutant. The concentration field is simulated using a Gaussian dispersion model. The paper focus on the performance analysis of the algorithm when different functions are used to quantify the error between the actual concentration field and the simulated concentration field from a candidate source. The Monte Carlo algorithm selects candidate sources based on the minimization of the error. In addition to the standard case, which can attain a potential perfect match between the sensor (synthetic) observations and the candidate source simulations, the performance analysis was also conducted in the more realistic cases where random noise was added to the sensor (synthetic) observations to simulate uncertainty. The analysis investigates the effects of the number of sensors, domain size and measurement uncertainties. The sensitivity of the error functions to the variations of these parameters was also analyzed by representing the contour lines of the corresponding error surfaces. The study shows the existence of a threshold (in our configuration 16 sensors), beyond which a larger number of sensors has little effect on the convergence rate. In the presence of noise, both number and density of sensors play a major role on the rate of convergence. The algorithm was tested with a real release from the Prairie Grass field experiment.

In several occasions the Monte Carlo algorithm converged towards a local optimum rather than the correct solution. This is due to the multi-modal characteristics of the search space, which includes several local minima. The convergence towards local optima occurred exclusively in cases where the initial guess was generated in the proximity of a minimum. Although a varying step size mutation rate was adopted to try to ‘escape’ from local optima, different methods can be used to avoid premature convergence. For example, possible alternative methods are evolutionary algorithms (e.g., Haupt, 2005), or more relaxed survival mechanisms, where the admitted solutions may not be the one with the highest evaluation score (e.g., Delle Monache et al., 2008). Both methods try to avoid local solutions by slowing down the rate of convergence to perform a more diverse sampling of the solution space.

## Acknowledgment

This material is partly based upon work supported by the National Science Foundation under Grant no. AGS 0849191.

## References

Allen, C.T., Young, G.S., Haupt, S.E., 2007. Improving pollutant source characterization by better estimating wind direction with a genetic algorithm. Atmospheric Environment 41, 2283–2289.

Arya, P.S., 1999. Air Pollution Meteorology and Dispersion. Oxford University Press, Oxford, UK 310pp.

Barad, M., Haugen, D., 1958. Project Prairie Grass. A Field Program in Diffusion. Geophysical Research Paper, No. 59, vol. I, Report AFCRC-TR-58-235(I), Air Force Cambridge Research Center, 299pp.

Cervone, G., Franzese, P., Ezber, Y., Boybeyi, Z., 2008. Risk assessment of atmospheric emissions using machine learning. Natural Hazards and Earth System Science 8, 991–1000.

Cervone, G., Franzese, P., Keesee, A.P., 2010. Algorithm quasi-optimal (AQ) learning. WIREs: Computational Statistics 2, 218–236.

Cervone, G., Michalski, R., Kaufman, K., Panait, L., 2000. Combining machine learning with evolutionary computation: recent results on lem. In: Proceedings of the Fifth International Workshop on Multistrategy Learning (MSL-2000), Guimaraes, Portugal, pp. 41–58.

Chang, J.C., Franzese, P., Chayantrakom, K., Hanna, S.R., 2003. Evaluations of CALPUFF, HPAC, and VLSTRACK with two mesoscale field datasets. Journal of Applied Meteorology 42, 453–466.

Chang, J.C., Hanna, S.R., 2004. Air quality model performance evaluation. Meteorology and Atmospheric Physics 87, 167–196.

Chow, F., Kosović, B., Chan, T., 2006. Source inversion for contaminant plume dispersion in urban environments using building-resolving simulations. In: Proceedings of the 86th American Meteorological Society Annual Meeting, Atlanta, GA, pp. 12–22.

Delle Monache, L., Lundquistand, J., Kosović, B., Johannesson, G., Dyer, K., Aines, R., Chow, F., Belles, R., Hanley, W., Larsen, S., Loosmore, G., Nitao, J., Sugiyama, G., Vogt, P., 2008. Bayesian inference and Markov Chain Monte Carlo sampling to reconstruct a contaminant source on a continental scale. Journal of Applied Meteorology and Climatology 47, 2600–2613.

Enting, I., 2002. Inverse Problems in Atmospheric Constituent Transport. Cambridge University Press, Cambridge, UK 392pp.

Gelman, A., Carlin, J., Stern, H., Rubin, D., 2003. Bayesian Data Analysis. Chapman & Hall/CRC, London, UK 668pp.

Goldberg, D.E., 1989. Genetic Algorithms in Search, Optimization, and Machine Learning. Addison Wesley, Reading, MA 412pp.

Hammersley, J., Handscomb, D., 1964. Monte Carlo Methods. Fletcher and Son Ltd, Norwich, UK 160pp.

Hanna, S.R., Chang, J.C., Strimaitis, G.D., 1993. Hazardous gas model evaluation with field observations. Atmospheric Environment 27A, 2265–2285.

Haupt, S.E., 2005. A demonstration of coupled receptor/dispersion modeling with a genetic algorithm. Atmospheric Environment 39, 7181–7189.

Haupt, S.E., Young, G.S., Allen, C.T., 2006. Validation of a receptor/dispersion model coupled with a genetic algorithm using synthetic data. Journal of Applied Meteorology 45, 476–490.

Haupt, S.E., Young, G.S., Allen, C.T., 2007. A genetic algorithm method to assimilate sensor data for a toxic contaminant release. Journal of Computers 2, 85–93.

Hourdin, F., Issartel, J.P., 2000. Sub-surface nuclear tests monitoring through the CTBT xenon network. Geophysical Research Letters 27, 2245–2248.

Metropolis, N., Ulam, S., 1949. The Monte Carlo method. Journal of the American Statistical Association 44 (247), 335–341.

Mosteller, F., Tukey, J.W., 1977. Data Analysis and Regression: A Second Course in Statistics, Addison-Wesley Series in Behavioral Science. Addison-Wesley, Reading, MA 588pp.

Pudykiewicz, J., 1998. Application of adjoint tracer transport equations for evaluating source parameters. Atmospheric Environment 32, 3039–3050.

Robert, C., Casella, G., 2004. Monte Carlo Statistical Methods. Springer, New York 536pp.

Rubinstein, R., Kroese, D., 2007. Simulation and the Monte Carlo Method. John Wiley and Sons, Hoboken, NJ 374pp.

Senocak, I., Hengartner, N., Short, M., Daniel, W., 2008. Stochastic event reconstruction of atmospheric contaminant dispersion using Bayesian inference. Atmospheric Environment 42, 7718–7727.