Addressing Air Quality Problems with Genetic Algorithms: A Detailed Analysis of Source Characterization

14

Sue Ellen Haupt, Christopher T. Allen, and George S. Young

14.1 Introduction

14.1.1 Fitting the Model to the Purpose

The purposes for modeling air contaminants have evolved, and the models themselves have co-evolved to meet the changing needs of society. The original need for air contaminant models was to track the path of pollutants emitted from known sources. Therefore, the initial purpose of the models was to track and estimate the downwind transport and dispersion (T&D). Since dispersion results from turbulent diffusion, which is best modeled as a stochastic process, most models for the dispersion portion are based on a Gaussian spread.

Because many environmental problems have their sources in a region that is far from the impact, there came a need to identify remote sources of pollution. For instance, the acid rain problem that was highly studied in the 1980s was widely thought to be caused by upwind polluters. Power plant emissions in the Ohio Valley were blamed for acid rain in New York and New England. To test this conjecture, receptor models were developed. This type of model begins with monitored pollution concentrations and back calculates the sources. Some models of this type were based on a backward trajectory analysis while others separated out the mix of chemical species present in the sample and computed likely sources given knowledge of the species composition of the potential sources. These models pointed to the Ohio Valley for the source of the acid rain precursors. Receptor models are still popular for attributing pollutants to their sources.

A more recent application analyzes the impact that a toxic release of chemical, biological, radiological, or nuclear (CBRN) material might have on a nearby population. Such a release could be due to an accident at a nearby plant or in transit, intentional release by a terrorist, or enemy action in a military situation. In such cases, there is often a need for a full spectrum of modeling, beginning with characterizing the likely source, then estimating contaminant transport and dispersion as well as their uncertainty, and finally estimating the impact on nearby populations and facilities for the purpose of deciding how best to respond to the situation. Such scenarios require rapid models for source characterization, T&D, and human effects (mortality rates, casualty rates, etc.); so including the full physics and dynamics is computationally prohibitive. Therefore, faster artificial intelligence techniques may become competitive. But such techniques are only as good as the dynamics-based models on which they are built. We show here how a genetic algorithm has proven useful for coupling backward looking receptor models

Sue Ellen Haupt (✉)
Applied Research Laboratory and Department of Meteorology, The Pennsylvania State University, P.O. Box 30, State College, PA 16802, USA
Phone: 814/863-7135; fax: 814/865-3287;
email: haupts2@asme.org

Christopher T. Allen
Computer Sciences Corporation, 79 T.W. Alexander Drive, Building 4201 Suite 260,
Research Triangle Park, NC 27709, USA
email: callen24@yahoo.com

George S. Young
Department of Meteorology, 620 Walker Building,
University Park, PA 16802, USA
Phone: (814) 863-4228; fax: 814/865-3663;
email: young@meteo.psu.edu

S. E. Haupt et al. (eds.), *Artificial Intelligence Methods in the Environmental Sciences*

with the forward T&D models to leverage the strengths of each in addressing the source characterization problem. We demonstrate here how to characterize the strength, location, height, time, and meteorological conditions of a release given field data. We begin by demonstrating the GA-based technique using synthetic data and a very basic T&D model and progress toward incorporating a realistic advanced applications T&D model and validating the techniques with field experiment data.

14.1.2 The Problem of Turbulent Dispersion and Real Data

Pollutant released into a turbulent atmospheric boundary layer is subject to chaotic motions on a variety of scales in both time and space. Thus, we cannot definitively predict an exact concentration for a specific location at an instant in time. As a result, predictive T&D models typically compute an ensemble average by solving a diffusion equation to yield a Gaussian spread. We must remember what such a model can and cannot do. It can predict an expected ensemble mean concentration and its standard deviation. It cannot, however, predict the expected concentration for a specific realization (Wyngaard 1992).

In contrast, concentration measurements represent a specific realization of turbulent dispersion. Currently, there is not a good evaluation method for comparing the single realization of a field experiment with the ensemble average statistics from model output (NRC 2003).

In addition to this stochastic variability of time averages, the pollutant emission rates are often poorly characterized; therefore, the dispersion problem appears intractable. Here we detail a method that uses artificial intelligence to directly treat the problem of inherent uncertainty through coupling a dispersion model to a receptor model. The goal is to blend the predictions of the T&D models with the monitored data, which are grounded in reality. Since blending these two disparate models becomes a complex optimization problem, the genetic algorithm (GA) is an appropriate tool to couple the field measurements to the dynamically based T&D model.

The GA-coupled model described here has evolved in parallel with the focus of the application. The initial formulation was for characterizing sources of air pollution by using the GA to link a forward T&D model with a backward looking receptor model. That model is described in detail in Section 14.2. Two applications with synthetic data are presented in Section 14.3: the first is in an artificial simple geometry and a second is in a realistic geometry. Although the model is shown to perform well, we immediately notice some cases for which the model is ill-posed. A statistical analysis of model performance appears in Section 14.4, which also describes model performance in the presence of random noise. In these initial sections, our goal is to apportion the fraction of monitored pollutant to each of a list of pre-identified sources. In Section 14.5, we begin to address the issues that are relevant for homeland security: what if we don't have a list of candidate sources and what if the local meteorological conditions aren't known? In this case, we apply the GA directly to identify the location, strength, and time of the release as well as to determine the direction of the wind that is transporting the contaminant. To accomplish this feat requires multiple receptors, each monitoring concentrations as a function of time. Section 14.6 is devoted to making the GA coupled model more realistic by incorporating a highly refined T&D model, SCIPUFF. This refinement requires reformulating the model to minimize calls to SCIPUFF and to optimize GA performance. With this refinement, we are able to examine model performance on actual field-monitored data, also presented in Section 14.6.

This problem of source characterization and characterizing the meteorological conditions is a very practical one that several government agencies are addressing. The application of the genetic algorithm to this problem demonstrates the real world applicability of artificial intelligence to such problems.

14.2 Coupled Model Formulation

The purpose of the coupled model is to assimilate field monitored data and back calculate the source characteristics of the emission. Several previous investigators used information on dispersion or chemical transport in computing source apportionment. Qin and Oduyemi (2003) apportioned particulate matter to its sources by using a receptor model and incorporating

dispersion model predictions from vehicle emission sources. Cartwright and Harris (1993) used a GA to apportion sources to pollutant data monitored at receptors. Loughlin et al. (2000) also used a GA to couple an air quality model with a receptor model. They minimized the total cost of controlling emission rates at over 1,000 sources in order to design cost effective control strategies to meet attainment of the ozone standard. Kumer et al. (2004) estimated apportionment factors that match monitored data by combining factor analysis-multiple regression with dispersion modeling. We describe here how we have built on these prior studies to couple a Gaussian plume model with a receptor model via a genetic algorithm to compute the source calibration factors necessary to best match the measured pollutant (Haupt and Haupt 2004; Haupt 2005; Haupt et al. 2006; Allen et al. 2006, 2007). Camelli and Lohner (2004) computed the location of a source that would cause the maximum amount of damage using a GA and a computational fluid dynamics model. Note that all the works mentioned here use Artificial Intelligence (AI) techniques to solve a difficult problem of blending two types of models.

14.2.1 Model Formulation

One method to apportion monitored concentrations to the expected sources is with a chemical mass balance (CMB) receptor model. Such a model starts with receptor data consisting of different monitored chemical species and a list of the emission fractions for each of those species for the potential sources in the locale. Mathematically:

$$\mathbf{C}_{mnr} \cdot \mathbf{S}_n = \mathbf{R}_{mr} \tag{14.1}$$

where C_{mnr} is the source concentration profile matrix denoting the fractional emission from source n; R_{mr} is the concentration of each species measured at a receptor r, and S_n is the apportionment vector, also called calibration factors, to be computed. Subscript n denotes the source number, m the species index, and r the receptor number. The monitored data provides the R_{mr} matrix denoting the amount of each chemical species present at receptor r. If the chemical composition of the emissions from each source is known, a fit to the data produces the fractional contribution from each source, S_n. Although our coupled model is inspired by the CMB model, we do not assume mass fractions of different species, but rather substitute varying meteorological periods. That is, m denotes the meteorological period for our reconfigured model. In the coupled model framework, the emission fractions in C_{mnr} are replaced with pollutant concentrations predicted by a T&D model at each receptor for each meteorological period. The receptor data R_{mr} matches the same meteorological periods. The vector, S_n, apportions or calibrates the expected transport model dispersed emissions to match actual concentrations measured at the receptor.

14.2.2 The Solution Method – A Continuous Genetic Algorithm

While one might begin to solve (14.1) with standard matrix inversion methods, one would quickly discover that the matrix is usually poorly conditioned and not easily inverted. This poor conditioning results because the meteorological periods are seldom independent. Therefore, we pose it as an optimization problem. Traditional optimization methods such as least squares and conjugate gradient perform poorly (Haupt et al. 2006). We did find, however, that other iterative methods such as the Moore-Penrose pseudoinverse (Penrose 1955) can produce an accurate solution for some of the simpler problems solved here. When we tried that method with more complex configurations, it did not produce a viable solution (Haupt 2005). In addition, we aim toward optimizing more than just the source calibration factors (see Section 14.5), so we expect the optimization problem to progress well beyond a matrix solution. Thus, we require a very robust optimization method that can solve this difficult matrix problem while simultaneously estimating other unknown parameters. We achieve this by using a GA as the coupling mechanism that minimizes the difference between the monitored concentrations and the predicted concentrations. The GA was introduced in Chapter 5 of Part I. The continuous GA is appropriate for application to this problem since all parameters are real continuous numbers.

The cost function used here measures the root mean square difference between the left-hand side of (14.1) and the right-hand side, summed over the total number

of meteorological periods considered and normalized. This normalized residual is:

$$\text{Cost} = \frac{\sqrt{\sum_{m=1}^{M}\sum_{r=1}^{R}(C_{mnr}\cdot S_n - R_{mr})^2}}{\sqrt{\sum_{m=1}^{M}\sum_{r=1}^{R}R_{mr}^2}} \tag{14.2}$$

where M is the total number of meteorological periods and R is the total number of receptors. We assume that R_{mr} are monitored data. Thurs, the crux of the model is now to use an appropriate transport and dispersion model to estimate C_{mnr}.

14.3 A First Validation

There are many ways to estimate the dispersed emissions that form C_{mnr}. The first validation problem uses Gaussian plume dispersion:

$$C_{mn} = \frac{Q_{mn}}{u\sigma_z\sigma_y 2\pi}\exp\left(\frac{-y_{mn}^2}{2\sigma_y^2}\right)\left\{\exp\left[\frac{-(z_r-H_e)^2}{2\sigma_z^2}\right] + \exp\left[\frac{-(z_r+H_e)^2}{2\sigma_z^2}\right]\right\} \tag{14.3}$$

where: C_{mn} = concentration of emissions from source n over time period m at a receptor location

(x, y, z_r) = Cartesian coordinates of the receptor in the downwind direction from the source

Q_{mn} = emission rate from source n over time period m

u = wind speed for meteorological period m

H_e = effective height of the plume centerline above ground

σ_y, σ_z = dispersion coefficients in the y and z directions, respectively

The dispersion coefficients are computed from Beychok (1994).

$$\sigma = \exp\left\{I + J\left[\ln(x) + K(\ln(x)\right]^2\right\} \tag{14.4}$$

where x is the downwind distance (in km) and I, J, and K are empirical coefficients dependent on the Pasquill Stability Class, in turn dependent on wind speed, direction, and solar radiation. The coefficients can then be looked up in tables (Beychok 1994).

Initially, we consider data from a single receptor but allow for multiple potential sources of the pollutant to be apportioned. Thus, the receptor index, r, in (14.3) collapses to 1.0 and no longer needs included for this first problem. The pollutant predicted by the forward model form C_{mn} of (14.3) and the monitored data become the right hand side, R_m, that is, the monitored data for the same meteorological periods. The meteorological periods are common to those metrics. The remaining vector, S_n, is thus the source calibration factor, which is tuned to optimize agreement between the model predicted concentrations and the receptor observations. If we had perfect world knowledge (of source characteristics, dispersion processes, meteorological conditions, turbulence, and monitored concentrations), S_n would be composed of all 1.0s. Therefore, the difference of this factor from 1.0 can be interpreted as an error or uncertainty in the modeling process in comparison to the monitored data.

Figure 14.1 summarizes the coupled model process. Given assumed geometry, meteorology, and emissions concentrations, we compute each source's dispersion plume. Then we estimate the contribution of each plume to the total concentration at the monitor with the forward model, which fill the concentration matrix. The monitor has recorded actual concentrations, R_m, for the same meteorological periods. The computed calibration factors then assign the portion to each source. Total emissions from a source can then be computed by multiplying the originally assumed emission rate by the source calibration factor.

14.3.1 Synthetic Data on a Circle

The coupled receptor/dispersion model technique was first validated in a simple geometry. A receptor was sited at the origin of a circle and 16 sources were spaced every 22.5° at a distance of 500 m. Receptor data were generated using the same dispersion model to be used for the coupled model optimization (equations (14.3) and (14.4)). This approach is sometimes called an identical twin experiment (Daley 1991). To estimate the calibration factors for 16 sources requires at least 16 independent meteorological periods. This independence was achieved by using wind directions from 16 points of the wind rose and representative

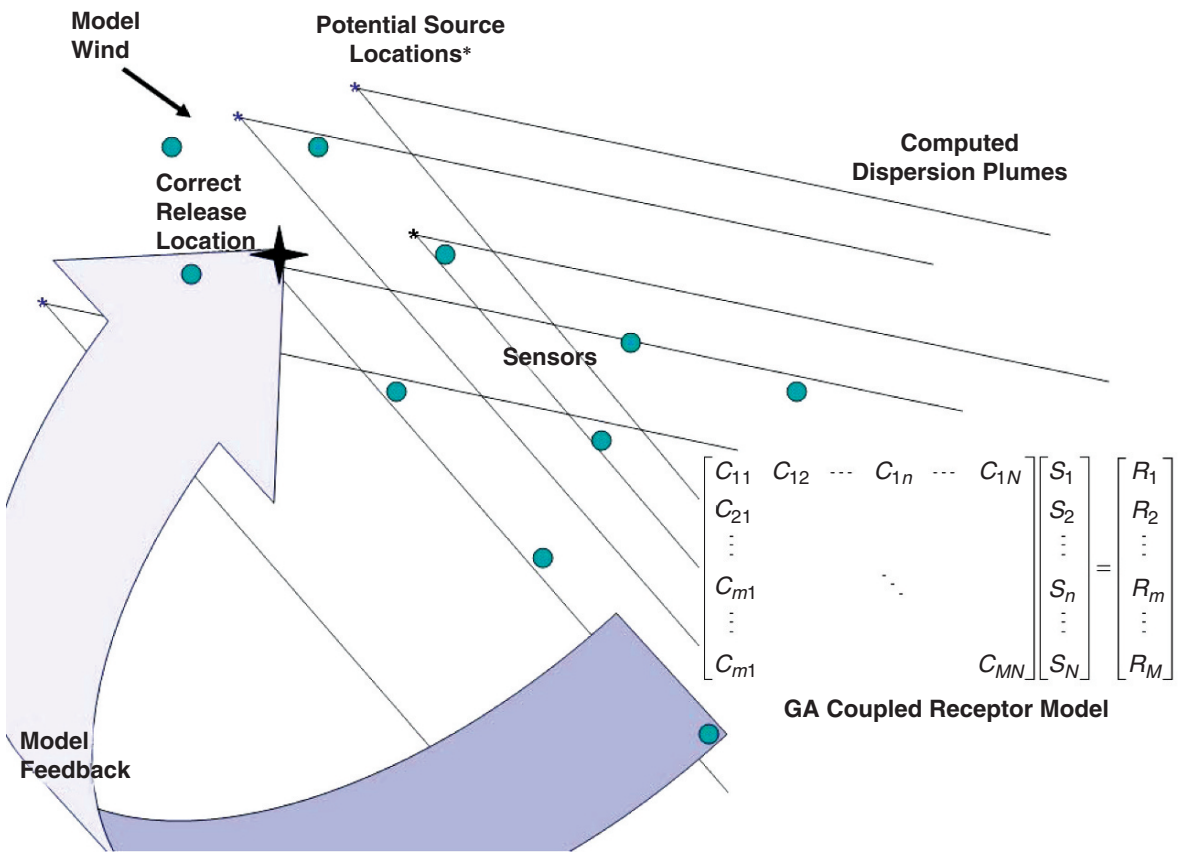

Fig. 14.1 Schematic of the GA coupled model. The monitored receptor data appears in matrix R, the concentration estimates in matrix C, and the GA computes the apportionment vector S to identify sources

wind speeds. Neutral stability was assumed for ease of comparison. The dispersion model was run using 1 h averaging over the meteorological data and specifying calibration factors, S_n, that we hoped to match with the coupled model.

The coupled receptor/dispersion model was then tested with this synthetically generated data. The first tests set calibration factors to 0.0 except for a single source that was set to a 1.0 to simulate identifying which single source might cause a contaminant event. The genetic algorithm, when run with a sufficient number of iterations, successfully evolved the correct solution. For this problem, the number of iterations determines the smallness of the cost function. Figure 14.2 shows the GA convergence over 200,000 iterations, much more than would be used in a typical run. The decrease in the residual is monotonic. So how many iterations are actually necessary or useful to get an acceptably small residual with a reasonable amount of computer time? Table 14.1 shows the results of a sensitivity study of residuals versus the number of iterations. Since the GA randomly generates the initial set of solutions, a different residual is expected for each run. Thus, for this table, each configuration was run five times and the mean and standard deviation of the residuals is listed. The results of Fig. 14.2 are confirmed: more iterations result in a smaller mean residual. The standard deviation also decreases with the number of iterations. Thus we expect this method to produce reliable results with a moderate number of iterations. If we are able to average multiple runs or to use a large number of iterations, the GA is even more likely to converge to a reliable solution. We now have confidence in our approach. Similar results hold for other configurations with two or more sources contributing.

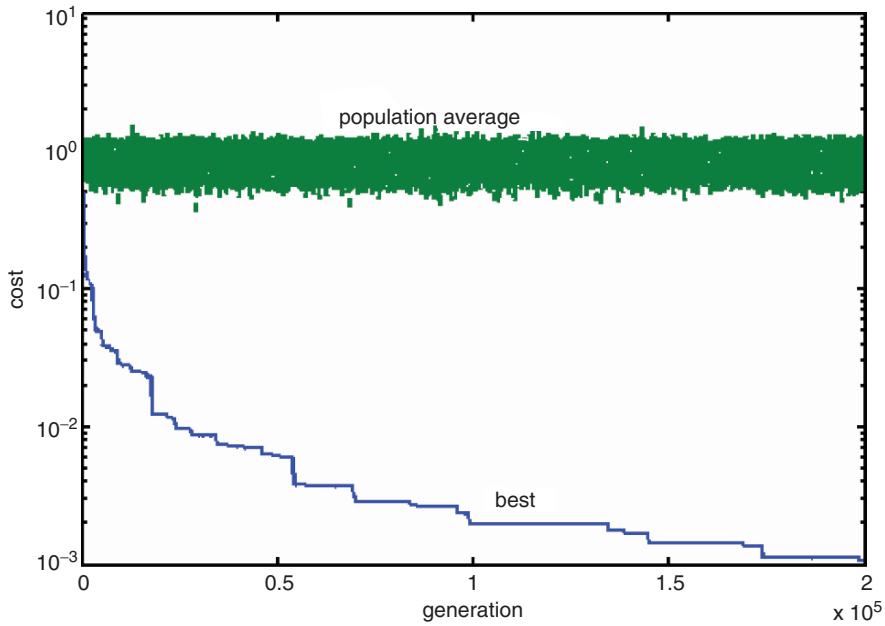

Fig. 14.2 Convergence for GA solution to the circular geometry problem

14.3.2 Actual Emission Configuration with Synthetic Meteorological Data

A second identical twin experiment used an actual emission configuration for Cache Valley, Utah. The source locations were obtained from the state of Utah emission inventory and source heights were estimated. Each source was assigned the same artificial emission rate. The receptor location is the actual monitor located on Main Street in Logan, Utah. Table 14.2 details the source data relative to the monitor. For verification purposes, the meteorological data were produced synthetically to systematically sample the range of possible winds. Source apportionment factors were assumed, once again assigning a factor of 0.0 to all except those chosen for a synthetic emission.

Table 14.1 Residual size as a function of the number of GA iterations for the circular source configuration. Statistics for all but the last row are based on five separate runs

Iterations	Best residual	Mean residual	Standard deviation
500	0.179	0.269	0.096
1,000	0.155	0.191	0.030
2,000	0.052	0.077	0.034
5,000	0.034	0.052	0.020
50,000[1]	5.36×10^{-4}		

[1]Based on a single run.

Using a real source configuration is a much more difficult problem than placing sources in a concentric circle. For instance, consider the case where two sources lie at the same angle from the receptor but at different distances. If the wind speed was not variable, it would be impossible to distinguish between the contributions from those two sources and so the problem would be ill-posed. Thus, we use a variety of meteorological conditions to produce a correct allocation of source apportionment factors.

Table 14.2 Source configuration for Cache Valley, UT

Source number	Distance from monitor (m)	Angle from monitor (° from north)
1	1,492	26.4
2	25,031	8
3	8,550	176
4	25,096	8
5	25,700	9
6	4,789	350
7	13,854	5
8	6,030	178
9	2,227	171
10	9,998	4
11	11,540	245
12	23,285	5
13	2,328	55
14	13,802	4
15	17,994	152
16	569	71

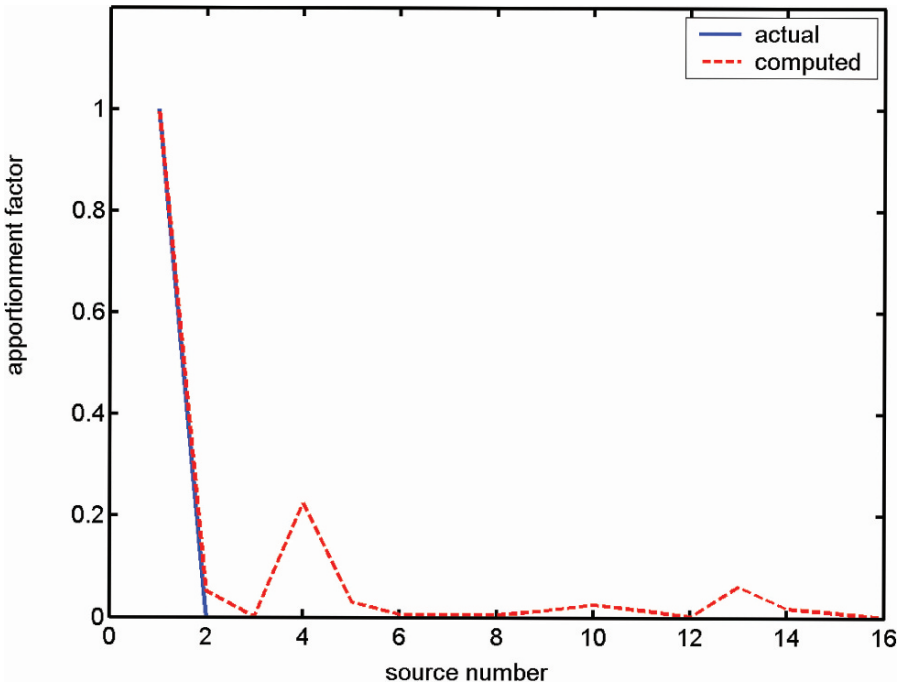

Fig. 14.3 Source apportionment for a single source (source 1) for the Cache Valley, UT configuration

The first validation example had only one source factor of 1.0 so only a single source contributed to the observed concentration. The source chosen was 1.5 km west of the receptor, a direction with no other sources. As seen in Fig. 14.3, the algorithm identified the correct source in less than 10,000 iterations. Some spurious contribution was attributed to source 4; but since that source is 25 km away, its contribution would be well dispersed by the time it reached the receptor. The normalized residual for this run was quite small: 0.0047604.

A second example was intentionally made more difficult to solve by setting an apportionment factor of 1.0 for three sources while the rest were assigned 0.0. The apportionment factors were optimized by the coupled model using 64 meteorological periods and 10,000 iterations. The results appear in Fig. 14.4. The three sources that were given 1.0s were well captured. An additional four sources were spuriously assigned large apportionment factors, in spite of the relatively small residual of 0.070144. Three of those, sources 2, 4, and 5, are located 23–25 km away from the receptor. Thus, their contribution was likely to disperse to a nearly zero concentration by the time it reached the receptor using the Gaussian plume model. Their apportionment factors, when multiplied by near zero, have little impact on the residual and are meaningless. Source 12 is 8.5 km away and therefore more likely to contribute, but it is in the same direction as the three sources that are making a real contribution. The lack of directional distinction makes it difficult to correctly identify only those sources that contribute to receptor pollutant concentration with the current configuration of the coupled model. The problem is depicted in Fig. 14.5. If a source that is 2 km from the receptor is much stronger than one that is only 1 km from the receptor, either could produce an equivalent concentration.

14.3.3 Model Sensitivity to Cost Function Formulation

Would a different formulation of the cost function produce different results? A cost function with a higher power on the difference than the root mean square (RMS) value in (14.2) would weight the outliers more heavily. To evaluate how this might impact the results, we look at alternate formulations for the cost function.

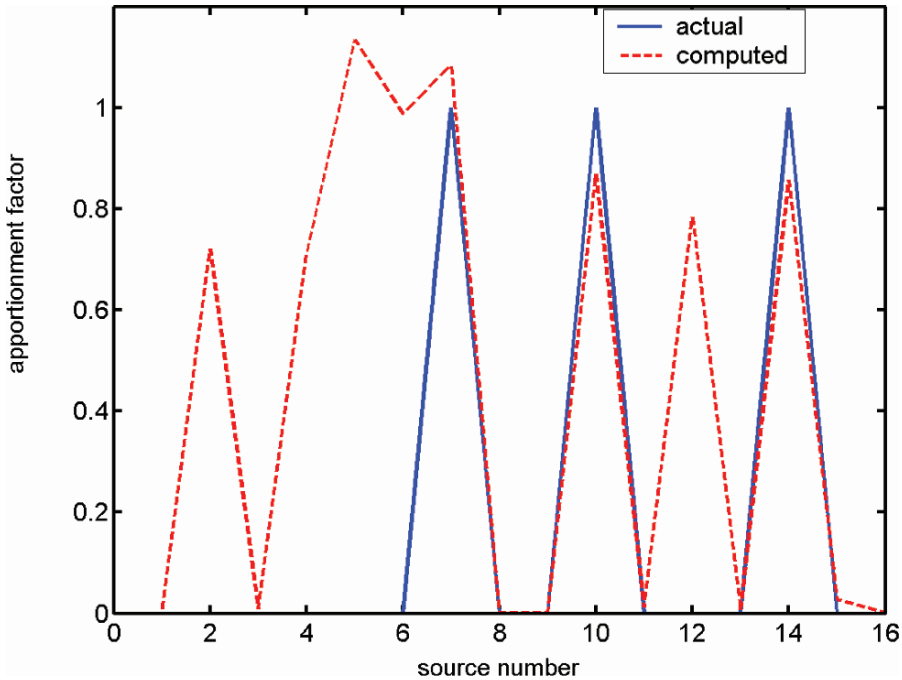

Fig. 14.4 Source apportionment for three sources (sources 7, 10, and 14) for the Logan, UT configuration

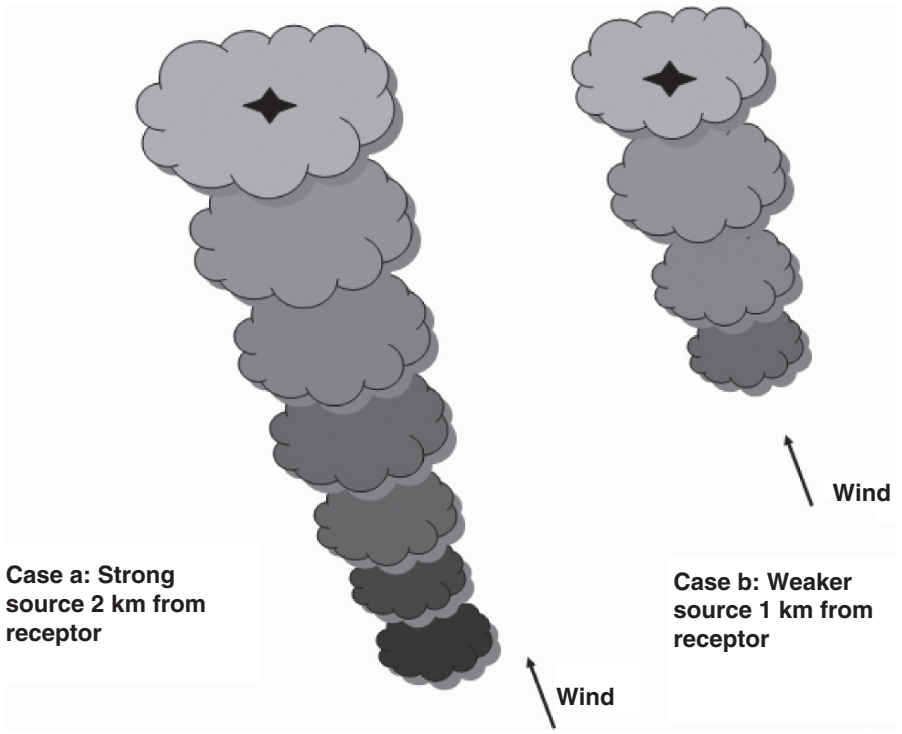

Fig. 14.5 Schematic of plume from two different sources at the same wind angle

Table 14.3 Evaluation of different cost function formulations for a circular geometry

Case & Metric	RMS	SqRoot	AbsVal	FourthRoot	EighthRoot	RMSAbs
RMS	0.050919	0.048137	0.044658	0.056269	0.063798	0.049764
Max	1.02305	1.02045	1.02457	1.02503	1.0352	1.02443
Min	0.97063	0.9727	0.97757	0.97215	0.97195	0.97546
In 0.01	10.5	11.2	11.3	8.0	9.8	11.2

The normalization method makes no difference since the GA mating function used here is based on ranking rather than absolute difference. The formulation of the cost function's numerator, however, could make a difference in the results or in the convergence properties of the model. We showed in Fig. 14.2 that, for this problem, the more GA iterations performed, the lower the cost function. We choose to lump accuracy and convergence properties into a single issue by holding the number of iterations in each GA coupled model run to 20,000.

Five additional cost function formulations are considered:

$$\text{SqRoot} = \frac{\left(\sum_{m=1}^{M} \sqrt{|C \cdot S - R|}\right)^2}{\left(\sum_{m=1}^{M} \sqrt{|R|}\right)^2} \tag{14.5}$$

$$\text{AbsVal} = \frac{\sum_{m=1}^{M} |C \cdot S - R|}{\sum_{m=1}^{M} |R|} \tag{14.6}$$

$$\text{FourthRoot} = \frac{\sqrt[4]{\sum_{m=1}^{M} (C \cdot S - R)^4}}{\sqrt[4]{\sum_{m=1}^{M} (R)^4}} \tag{14.7}$$

$$\text{EighthRoot} = \frac{\sqrt[8]{\sum_{m=1}^{M} (C \cdot S - R)^8}}{\sqrt[8]{\sum_{m=1}^{M} (R)^8}} \tag{14.8}$$

$$\text{RMSAbs} = \text{RMS} + \text{AbsVal} \tag{14.9}$$

Table 14.3 summarizes the results for the circular geometry with all sources assigned a calibration factor of 1.0. The results reported there are for the average of six coupled model runs of 20,000 iterations each. The four different metrics used are:

1. RMS: The RMS difference from the calibration factor that was used to create the synthetic data. We hope to see this minimized.
2. Max: The maximum calibration factor for each run, averaged over the six runs. We hope to see this as close to the actual as possible (1.0 for the circle).
3. Min: The minimum calibration factor each run, averaged over the six runs. We again hope to see this as close to the actual (1.0) as possible.
4. In 0.01: The number of sources calibrated within 1% of actual. A higher value for this metric implies a better result. For the circle case that includes 0.0 apportionment factors, this means within 1% of 1.0.

As seen in the table, there is no clear winner among the cost functions, although the higher power cost functions perform somewhat worse than the SqRoot, AbsVal, RMS, and RMSAbs. For the circular configuration, the AbsVal function works best, closely followed by the SqRoot. For a different geometry the results were somewhat different, but performance differences between the cost functions are relatively small.

A few runs of the GA coupled model with 200,000 iterations for the RMS and AbsVal cost functions confirmed the results of Table 14.3. Thus, although genetic algorithm results can be sensitive to formulation of the cost function, for this problem, any of the cost functions described above will give similar results. We conclude that our original choice of an RMS cost function was reasonable and easy to compare with other methods that are based on RMS differences (Haupt et al. 2006).

14.3.4 Tuning the GA to the Problem

We saw that for both a simple geometry and for a more realistic geometry, the coupled model is able to correctly apportion concentrations to sources in spite of a few spurious apportionments for the most difficult

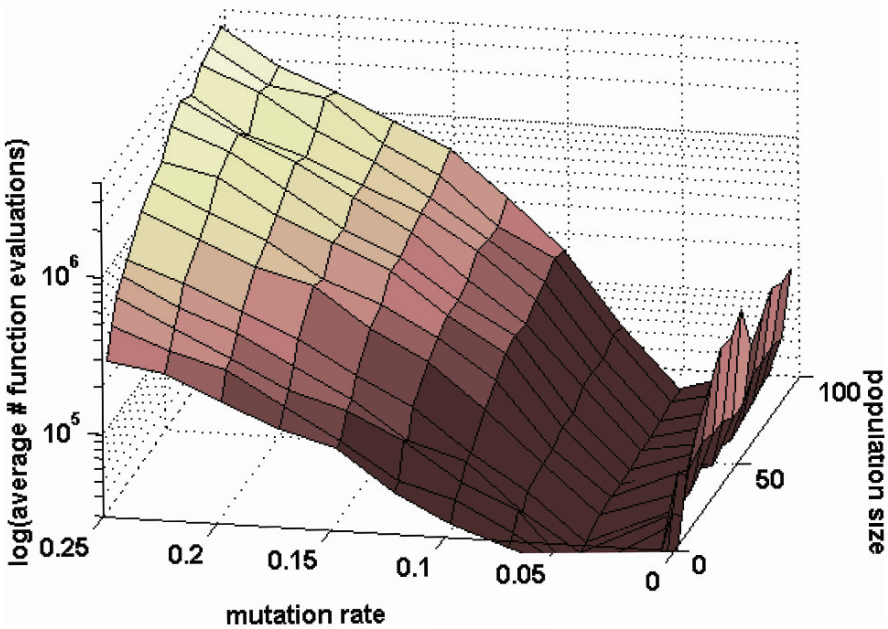

Fig. 14.6 The number of cost function evaluations required to reach a tolerance of 0.01 as a function of population size and mutation rate, averaged over 10 runs

situations. Now we wish to analyze which combinations of GA parameters optimize model performance. When we move to more refined dispersion models (Section 14.6 below), we expect the computation of the dispersion matrix to be computationally expensive, so we wish to determine the best combinations of population size and mutation rate to minimize the number of calls to the cost function, similar to the analysis given in Chapter 5. We noted that for this problem, the GA convergence depends on the number of iterations. Since the solution is known for these identical twin experiments, we can stop the GA when the error has reached a pre-specified tolerance level, in this case 0.01. We wish to explore a wide range of parameter combinations. The goal is to minimize the number of cost function evaluations required to reach this level in an effort to minimize the CPU time. Mutation rates examined are 0.001, 0.005, 0.01, 0.05, 0.075, 0.1, 0.125, 0.15, 0.175, 0.2, 0., and 0.25. Population sizes are 4, 8, 12, 16, 20, 32, 40, 48, 56, 64, 72, 80, 88, and 96. We run the GA for each combination of population size and mutation rate and count the total number of calls to the cost function to achieve convergence (population size times the number of generations, reduced by the number of members that have not changed from one generation to the next). Since the convergence of the GA progresses differently with each random initialization, we average ten separate runs for each mutation rate/population size combination to produce the results in Fig. 14.6. It shows that for this problem, there are various ways to combine population size with mutation rate to produce fast convergence. One way is to use relatively small mutation rates (order of 0.01). The other is to use moderately small population sizes (8–20), even with larger mutation rates (0.15 to 0.2) such as we did in the previous runs. The lowest average number of function evaluations occurred when the mutation rate was 0.05 and the population size was 12. Such a configuration for running the GA is sometimes referred to as a micro-GA due the small population size. These results are similar to those of Chapter 5. Parameter ranges such as these tend to emphasize the impact of mutation and are preferred when there are multiple closely spaced local minima. When such parameter combinations are used, the emphasis is on finding the single best solution rather than evolving the entire population. This is why the mean residual in Fig. 14.2 remained relatively constant in spite of the rapid decrease for the best solution. Note that using elitism, which maintains the best individual

in the population unchanged, is essential for such applications.

The analysis presented here has assumed a serial computer. The analysis would be quite different if a large number of parallel processors were available and the GA was coded to take advantage of them as discussed in Chapter 5.

14.4 Statistical Analysis of Model Performance

14.4.1 The Monte Carlo Approach

This statistical analysis revisits the circular geometry consisting of a single receptor surrounded by 16 potential sources at a radius of 500 m. As discussed above, a single run of a GA coupled model is typically sufficient to estimate the actual calibration factor to within two significant digits for this case.

To analyze confidence in the ability of the GA-optimized coupled model methodology to match a known solution, a Monte Carlo technique is used. The GA is run on the same problem 100 times with different initial random seeds. From the resulting sample of solutions we are able to estimate the mean, median, and error bars. Figure 14.7 depicts the mean calibration factor at each source as found by the GA along with the corresponding error bars. The inner error bars represent one standard deviation. The outer bars denote the 90% confidence interval; that is, 5% of the solutions are above the highest bar and 5% are below the lowest. We see that we are 90% confident that solutions range between 0.97 and 1.03 for each source, closely bracketing the true solution of 1.0. The mean of the 100 cases ranges between 0.9976 for source number 1 through 1.003 for source 11. Thus, the mean value computed from 100 runs is even more reliable than the already good solutions from a single GA coupled model run. Therefore, a single GA-optimized coupled model run is accurate to within 3% and the mean of 100 runs accurate to 0.3%.

Our prior work confirmed that these results are not unique to this prescribed configuration – either the specific calibration factors or the geometry (Haupt et al. 2006). When a spiral geometry, ranging in source-receptor distance of 250–1,750 m, was used instead, the results showed that the GA coupled model can correctly apportion the sources and that using the mean of the 100 Monte Carlo runs reduced the error.

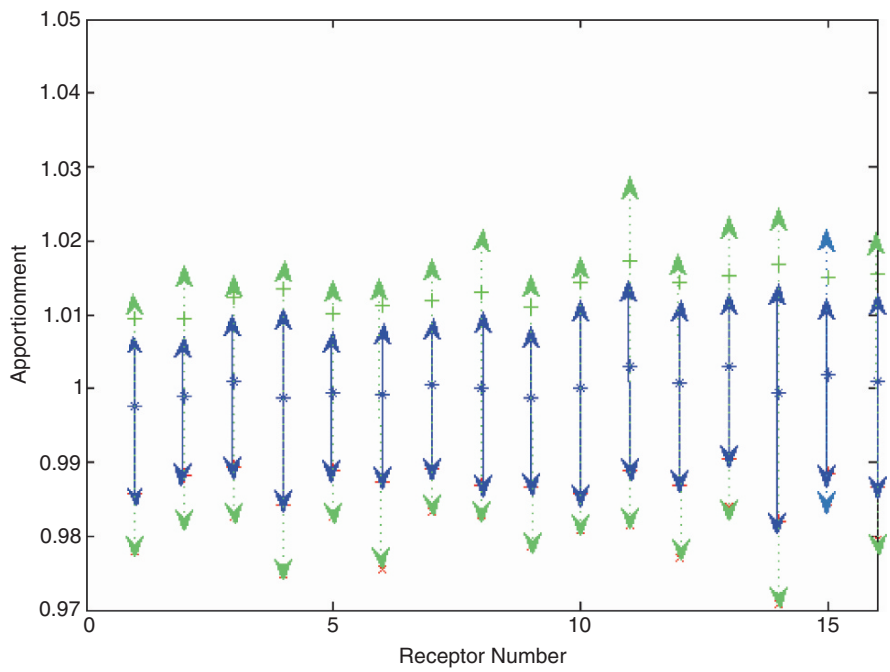

Fig. 14.7 Error bars from 100 Monte Carlo runs of the GA coupled model for the circular source configuration with correct apportionment factors all set to 1. The inner (solid) error bars are the one standard deviation level and the outer (dashed) error bars denote the 90% confidence level

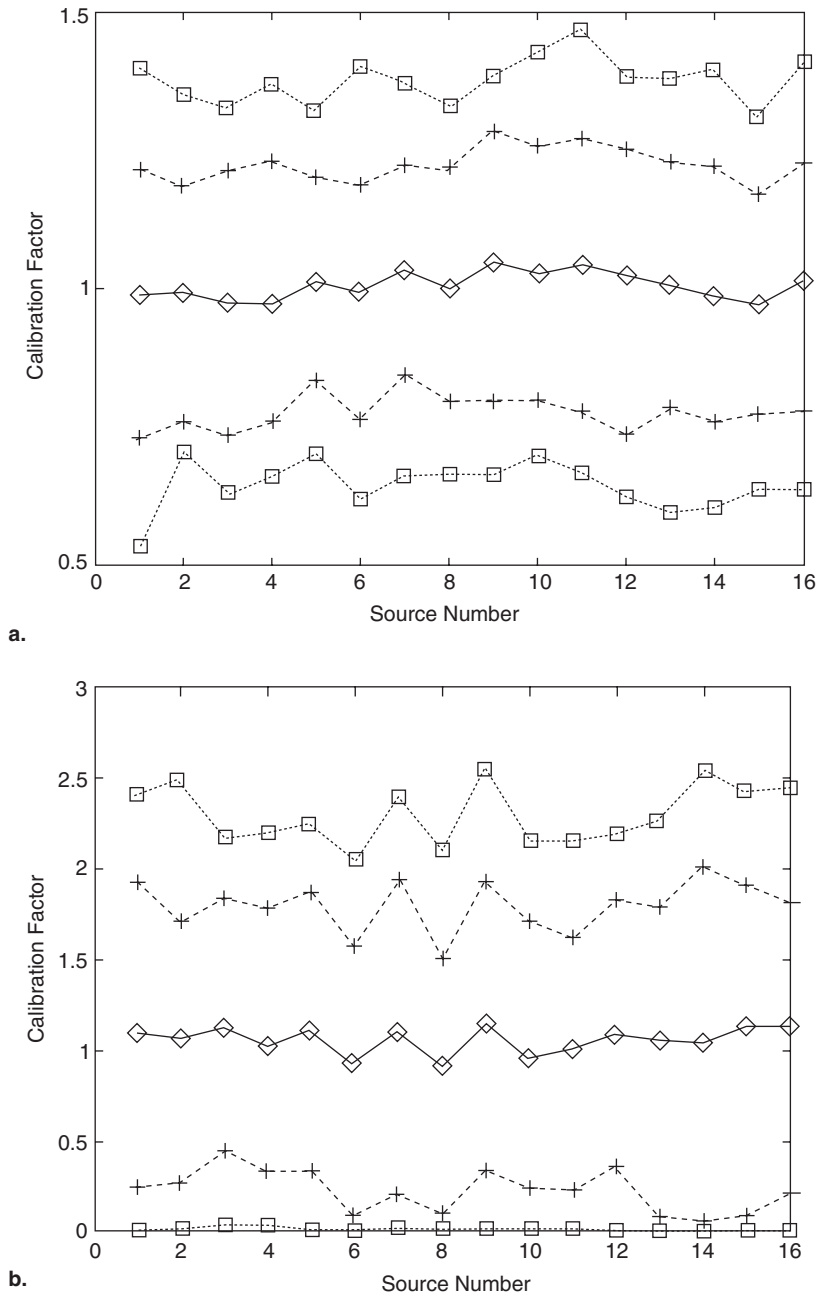

Fig. 14.8 Error bars from 100 Monte Carlo runs of the GA coupled model for the circular source configuration with correct apportionment factors all set to 1. White noise is added with amplitude equal to that of the signal: (a) Additive noise and (b) multiplicative noise. The inner solid black line marked by diamonds is the mean solution. Error lines denote one standard deviation (long dash marked by crosses) and 90% confidence level (short dashes marked by squares)

14.4.2 Analysis Including Noise

Real-world data do not have the pure signal available in our synthetically constructed data. Typical situations involve errors and uncertainties in both the emission and receptor data as well as the meteorological data. In addition, there is an inherent mismatch between the ensemble average nature of the model predictions and the single realizations yielded by the monitor measurements. In our analysis here, we simulate the aggregate uncertainty by incorporating white noise into the data, and then using the GA coupled model to optimize the calibration factor. No assumption is made of the source of this noise. It represents errors in both the monitored data and in the modeling process (ranging from uncertainty in source strength,

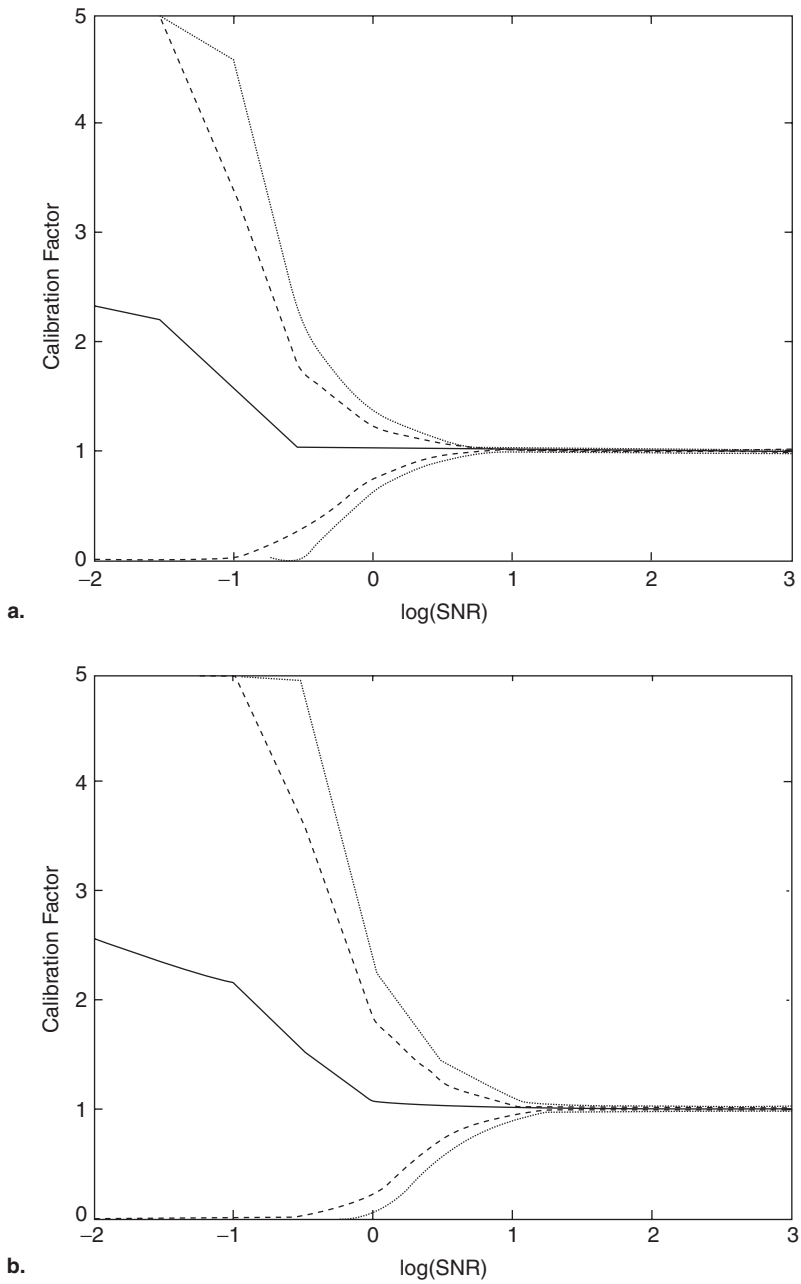

Fig. 14.9 Error bars from 100 Monte Carlo runs of the GA coupled model for the circular source configuration with correct apportionment factors all set to 1. White noise is added for various amplitudes as indicated on the abscissa: (a) Additive noise and (b) multiplicative noise. The inner solid black line is the mean solution. Error lines denote one standard deviation (long dash) and 90% confidence level (short dashes)

meteorological data, numerical error, and modeling simplifications, including the ensemble averaging assumption).

We again consider the circular geometry (see Section 14.3.1) with meteorological data representing 16 points of the wind rose. The Monte Carlo analysis uses assumed source calibration factors of all ones to create the receptor data. In this case, however, two separate methods of including white noise with an amplitude equal to that of the signal are used to simulate errors and uncertainties in the modeling process. First, white noise with mean amplitude of 1.0 is added to the dispersion model when creating the synthetic receptor data. The second analysis uses white noise to multiply the signal. Thus, the receptor data that goes into R_{mr} in equation (14.1) includes as much noise as signal. The GA coupled model is then used to compute the optimal calibration factors.

Figure 14.8 shows the mean, standard deviation, and 90% confidence interval for 100 runs of the GA coupled model. Figure 14.8a depicts additive noise while Fig. 14.8b depicts the multiplicative noise. In both cases, the curves depicting the error span a much wider range than for the case with no noise shown in Fig. 14.7. Aggregating the full 1,600 source cases (16 sources all with actual apportionment value of 1.0 over 100 runs) is equivalent to having 1,600 runs for a single source. Such an aggregation produces a mean value of 1.0066 for additive noise, quite close to the actual value of 1.0. The mean standard deviation of the aggregated 16 sources is 0.02595, an order of magnitude larger than for the case without noise (0.0015109) but still sufficiently small to provide confidence in model performance with imperfect information. Figure 14.8b indicates that the spread of the standard deviation and 90% confidence interval curves is greater for the multiplicative noise than for additive noise. In this case, the standard deviation for the multiplicative noise is 0.07449, which is greater because variability is proportional to the data itself.

Figure 14.9 depicts the performance of the coupled model over a range of signal to noise ratios (SNRs) for the additive and multiplicative noise cases. This plot aggregates the data over all 16 sources. We see that as long as $\log(\text{SNR}) > 1$, the solutions are quite close to the actual solution of 1.0 and the scatter is quite small. As noise becomes greater than the signal ($\log(\text{SNR}) < 0$), however, the computed solution diverges from the actual and the scatter becomes wider. Note that the mean of the solutions is still 1.0. At $\log(\text{SNR}) = 0$ the noise equals the signal and we have the case presented in Fig. 14.8 above. As expected, when the noise becomes much larger than the signal, as on the left side of the plot, the coupled model no longer reconstructs the solution reliably. In fact, the mean solution tends to 2.5, which is the center of the range allowed in the optimization routine. The standard deviation and 90% confidence lines approach the limits of the range. For multiplicative noise, the variability increases with decrease in SNR more rapidly than for the additive noise.

Haupt et al. (2006) report results for SNR analysis of other source configurations. The results described above generally hold and can be summarized as: (1) when multiple runs are averaged, confidence in the results is higher and (2) the GA coupled model run in Monte Carlo mode can apportion the sources correctly in the presence of noise of the same order of magnitude as the signal.

14.5 Tuning Meteorological Data

Accurate transport and dispersion modeling of pollutant releases requires accurate meteorological data – in particular, an accurate wind field. For most dispersion modeling applications, we don't have the meteorological fields at the preferred resolution, making precise computation of atmospheric dispersion quite difficult. Moreover, available wind data are not always accurate or representative. Thus, accurate source characterization can be difficult.

Here we present a new GA-based method that addresses the uncertainty associated with meteorological data by using a GA to tune the surface wind direction in addition to pollutant source characteristics. This method is an extension of the GA-coupled model described in Section 14.2. This extended method has an advantage over the original GA-coupled model in that it is far less sensitive to the uncertainty in meteorological data.

14.5.1 Architecture

Because this problem is different than the simpler source characterization problem presented earlier, changes must be made in the model architecture. As discussed in Section 14.2, the coupled model considers an array of candidate source locations, and the GA optimizes the strength of each candidate source by comparing dispersion model predictions with monitored receptor data. The source(s) with non-zero strengths are then assumed to be the actual emitters of the pollutant. In the method presented in this section, however, potential source locations are not known *a priori*. Neither is the wind direction. Instead, the source location comprises two of the GA-tunable parameters (x and y location) so that the model is free to choose any location within the domain. The wind direction can be any number between 0° and 360°. Thus, the performance of the method is not dependent on the appropriateness of a pre-defined candidate

source array or the presumed wind direction. Those are now free parameters, as is the source strength.

14.5.1.1 Forward Model

The disadvantage of this new architecture is that because the source locations and wind directions change as the GA evolves the population, the model must recalculate the pollutant dispersion from all candidate solutions at each iteration, thereby increasing the required CPU time. We use the Gaussian plume model (14.3) to test the method. This method uses concentration forecasts for each trial solution created using equation (14.3), receptor data for an arbitrary number of sites, and the GA to find the combination of source location, strength, and surface wind direction that provides the best match between the monitored receptor data and the expected concentrations.

14.5.1.2 Cost Function

The cost function used by the GA to evaluate each candidate solution is the root mean square difference between concentrations predicted by (14.2) and receptor data values, summed over all receptors. The cost function is similar to (14.2), except for changes in notation associated with the context of the current problem. Specifically, the cost function is defined as:

$$\text{Cost} = \frac{\sqrt{\sum_{r=1}^{TR} \left(\log_{10}(aC_r + 1) - \log_{10}(aR_r + 1)\right)^2}}{\sqrt{\sum_{r=1}^{TR} \left(\log_{10}(aR_r + 1)\right)^2}} \tag{14.10}$$

where C_r is downwind concentration at receptor r as calculated by (14.3), R_r is the receptor data value at receptor r, TR is the total number of receptors, and a is a constant. It is necessary to add 1.0 to the concentrations because the logarithm of zero is undefined. Doing this has the beneficial side effect of minimizing the contribution from the weakest concentrations values whose magnitudes are many orders of magnitude less than 1.0. The data must therefore be scaled since many of the C_r and R_r values are several orders of magnitude less than 1.0. The scaling factor, a, depends on the sum of all data values over all receptors:

$$a = \max\left(\frac{1}{\sum_{r=1}^{TR} R_r}, 1\right) \tag{14.11}$$

If the receptor data sums to a value greater than 1.0, then a is 1.0. Otherwise, a is greater than 1.0, so that at least some values are comparable in magnitude to 1.0. Scaling the concentration values allows the cost function to retain sensitivity to signal while still reducing sensitivity to noise via the logarithm.

14.5.1.3 Mating Scheme

In prior sections, the GA used a continuous version of single point crossover discussed in Chapter 5. For this reformulated problem, we obtain better GA performance by using a uniform crossover mating scheme that blends all parameters rather than just a single parameter. The uniform crossover method improves the average skill score (see Appendix for the definition of skill scores – lower skill scores are better) across six runs from 0.613 to 0.061, a remarkable improvement.

The superiority of this uniform crossover scheme to single-point crossover used in the models is most likely due to correlations between the effects of different parameters, specifically the dependence of plume structure on both wind direction and source location. Each wind direction has a unique optimal source location resulting in the best match to the receptor data. If the GA finds this location for a particular wind direction, and the wind direction is modified, the location is no longer optimal. Single-point crossover tends to converge to one of these "optimal" locations while failing to progressively improve the wind direction. Blending all parameters ensures that both the wind direction and source location are modified simultaneously, allowing both parameters to be progressively improved through the GA and decreasing the likelihood of premature convergence. Because of the correlations described above, the changes resulting from simultaneous modification of the wind direction and source location must complement each other. In a general sense, effects of parameters that are highly correlated in other applications are expected to exhibit similar behavior here.

14.5.1.4 GA Parameters

With the all-blending mating scheme, we have chosen to alter the GA parameters toward larger population sizes and smaller mutation rates. With a population size of 1,200, the GA can find the solution in a single run with 100 iterations or less. Larger population sizes than 1,200 and longer runs than 100 iterations result in slightly better performance, but the improvement is not significant when compared to the extra computing time. Smaller population sizes often converged too quickly to an incorrect solution, even when using a high mutation rate. Here, we use a mutation rate of 0.01 and a crossover rate of 0.5 for this problem of finding source location and wind direction in addition to source strength.

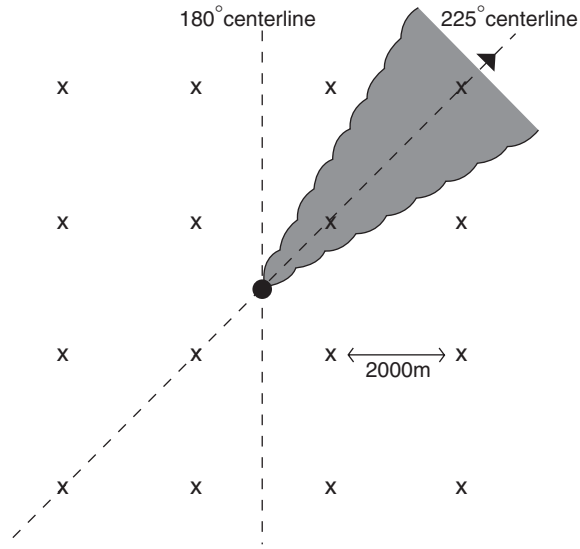

Fig. 14.10 The synthetic setup for a 4-by-4 grid of receptors. The black dot in the center represents the source and the X's are the receptors, each separated by 2,000 m. The dashed lines represent the plume centerline for the two wind directions considered in the synthetic data sets, and the shaded area represents a sample plume for the 225° wind direction

14.5.2 Demonstration

To demonstrate and validate the method of tuning meteorological data and source characteristics, we use synthetic data produced by (14.3) as receptor data. We place the receptors on a grid surrounding a single source with 2,000 m separating each receptor, and the source located in the center of the receptor domain at the point defined as the origin (0,0). To determine the dependence of model performance on the quantity of receptor data available, model runs are performed using 2-by-2, 4-by-4, 8-by-8, 16-by-16, and 32-by-32 grids of receptors. Synthetic data is produced for each receptor configuration for two different wind directions, 180° and 225°. These two wind directions represent opposite scenarios: a wind direction of 180° places the plume centerline directly between receptors, and a wind direction of 225° places the plume centerline directly over the receptors located along the $x = y$ diagonal in the northeast quadrant of the domain. Figure 14.10 shows the source and receptor setup for a 4-by-4 grid of receptors, where the black dot in the center is the source, the Xs are the receptors, and the dashed lines are the plume centerlines for the 180° and 225° wind directions. Model runs are performed for these five receptor configurations and two wind directions.

Table 14.4 shows the results for six different setups (receptor grids of 8-by-8, 16-by-16, and 32-by-32, for each of two wind directions) using a population size of 1,200 and 100 iterations. All GA runs produced a solution close to the actual, and some a tolerance of 0.01° in wind direction, 1% of source strength, and 1.0 m in source location. It may be puzzling at first that one of the 32-by-32 runs returned a worse result than

Table 14.4 GA-produced wind directions, source strengths, source locations, and skill scores for six synthetic configurations using a population size of 1,200, mutation rate of 0.01, after 100 iterations, for a single GA run. The correct solution is $\theta = (180°$ or $225°)$, strength = 1.00, and (x, y) = (0, 0). Appendix describe skill scores

Configuration	θ	Strength	(x, y) (in m)	Skill score
8-by-8, $\theta = 180°$	184.12°	2.96	−417, 1,346	1.4581
8-by-8, $\theta = 225°$	223.95°	1.06	−26, − 56	0.1952
16-by-16, $\theta = 180°$	180.01°	1.00	−1, 0	0.0029
16-by-16, $\theta = 225°$	225.01°	1.00	−1, 1	0.0019
32-by-32, $\theta = 180°$	180.00°	1.00	0, 0	0.0000
32-by-32, $\theta = 225°$	220.27°	1.12	−123, 519	0.6870

either of the 16-by-16 runs, but this occurred because each GA run begins with a random initialization, and the results in Table 14.4 reflect a single "test" run, not an average over many runs. The 32-by-32, 225° test run was just not as fortunate in its initialization. In general, runs with a 32-by-32 receptor grid perform at least as well as runs with fewer receptors.

Table 14.5 Number of runs (out of six) that produced a solution within tolerance for the given combination of population size and number of iterations. The rows are different population sizes, and the columns are different numbers of iterations

	Iter = 50	Iter = 100	Iter = 150	Iter = 200
Pop = 400	3	4	4	5
Pop = 800	4	4	4	5
Pop = 1,200	5	6	6	6
Pop = 1,600	5	6	6	6

14.5.2.1 Refinement

The solution after the 100th GA iteration is often close to, but not exactly at the global minimum of the cost function. Increasing the number of iterations above 100 does not greatly improve the solution for this reformulated problem. Therefore, we investigate whether a hybrid GA incorporating a traditional gradient descent method such as the Nelder-Mead Downhill Simplex NMDS method (Nelder and Mead 1965) could further improve the solution more efficiently than a GA does after the 100th iteration. The NMDS starts from a previously chosen point on a multidimensional surface (i.e. the cost function) and finds a local minimum in the vicinity of the starting point. For our application, we use the best GA-produced solution after 100 iterations as the starting point for the NMDS method. Gradient descent methods such as the NMDS are ineffective alone, however, as the it can only find the global minimum if the first guess is in the correct valley.

The NMDS method was run using each solution from Table 14.4. Each time, the NMDS returned a solution within our close tolerance limits, even for GA-generated starting points that were not "close enough". This improvement suggests that even though some of the specific values in the solutions from Table 14.4 are not within the tolerances, they are within the same cost function basin as the true solution. Under these circumstances, the NMDS can be used effectively to further improve the accuracy of the solution after the termination of the GA. The procedure as a whole is often called a hybrid GA, where a GA first is used to locate the basin of the global minimum of the cost function, and then the more traditional NMDS method is used to fine-tune the minimum. This hybrid GA produces a consistently good solution, better than either the GA or the NMDS method alone, in less computation time than the GA alone.

Running the GA beyond the 100th iteration does continue to improve the solution, but not as efficiently as the NMDS algorithm. Thus, we wish to run the GA just long enough to get to a solution that is an in-basin starting point for the simplex method. To determine where we should stop the GA, we ran the hybrid GA using 16 combinations of population size and number of iterations (each of which is proportional to computing time) to determine how much computing time is necessary to obtain an in-basin starting point.

Table 14.5 shows how many of six runs returned a solution within the tolerance after application of the NMDS method for each combination of population size and number of GA iterations. The combination of population size of 1,200 and 100 iterations was the most efficient to achieve this level of accuracy for all six runs made with the least computing time, the reason we use these values here.

Since the NMDS method is fairly efficient, could we just randomly generate initial guesses and still converge to the solution? Has the GA added any value? To answer these questions, we make multiple NMDS runs originating with random starting points within the solution domain. Table 14.6 indicates the number of function calls (a uniform unit of computing time) required by the GA and the random initialization NMDS method to find a solution within tolerance. The results were averaged over two runs for each receptor and wind direction configuration, for a total of 12 runs. The number of function calls required in any individual run using the NMDS varies greatly because the success of the NMDS method depends on the starting point, and the total number of function calls required is simply a function of how long it takes to produce an in-basin first guess. Because these first guesses are random in this experiment, it is not surprising that in some instances, the NMDS method found the solution faster than the GA. The performance of the GA, however, is far more consistent than NMDS over the

Table 14.6 Number of cost function evaluations required to find the solution for the GA and the Nelder-Mead downhill simplex, averaged over two runs for each configuration. Each configuration consists of an n-by-n receptor grid and a wind direction of either 180° or 225°

Configuration	GA function calls	Nelder-Mead function calls
8-by-8, $\theta = 180°$	19,200	17,180
8-by-8, $\theta = 225°$	1,200	123,225
16-by-16, $\theta = 180°$	13,800	60,874
16-by-16, $\theta = 225°$	3,600	121,035
32-by-32, $\theta = 180°$	10,200	16,996
32-by-32, $\theta = 225°$	23,400	133,034

12 runs performed, because it is able to overcome an unfortunate starting population and find the basin of the global minimum. Averaged across all six configurations tested, the GA took an average of 11,900 function calls to find a solution within tolerance, while the NMDS method took an average of 78,725 function calls. Thus, running the simplex from random starting points until the solution is found is inefficient compared to the GA, and particularly to the GA-NMDS hybrid. Moreover, if we did not know the correct solution *a priori*, the hybrid GA would assure us of convergence, particularly with multiple runs while the NMDS method alone would not.

14.5.2.2 Receptor Grid

How much receptor data is necessary to determine both wind direction and the source characteristics? The reason Tables 14.4 and 14.6 do not give results for the 2-by-2 and 4-by-4 receptor grids is that correct solutions could only be found consistently when using at least an 8-by-8 grid of receptors. For a 2-by-2 receptor grid, solutions were nearly random. For a 4-by-4 grid, solutions were somewhat better, but not nearly as good as the 8-by-8 grid solutions. This result suggests that a 4-by-4 grid of receptors does not provide enough receptor data to distinguish the effects of wind direction from those of source location and source strength. It does not imply that more than 16 total receptors are needed, as only two or three of the receptors in a 4-by-4 grid provide useful data (the others are outside the plume or nearly so). Because there are four parameters to be tuned (wind direction, source strength, and two for source location), having fewer than four data values does not provide enough information to resolve all the unknowns. In contrast, for an 8-by-8 grid, the number of receptors inside the plume exceeds the number of unknowns, so the hybrid GA is successful.

14.5.2.3 Noisy Observations

The success of the synthetic data runs is partly due to the exact match between the synthetic receptor data and the expected concentrations calculated by (14.3). This was also the case with synthetic data in Section 14.3 with the GA-coupled model. Therefore, we again contaminate our synthetic data with white noise to simulate the variability and errors present in monitored receptor data in order to gauge our new hybrid GA's performance when faced with inexact receptor data.

Twelve model runs are performed for each combination of receptor grid size and signal-to-noise ratio (SNR). Six SNRs are tested: infinity (no noise), 100, 10, 1, 0.1, and 0.01. Analyses are made for wind directions of both 180° and 225°. It is expected that runs with more receptors are less sensitive to noise than runs with fewer receptors. While an 8-by-8 receptor grid provides sufficient information to produce the solution with no noise, it may not provide enough information when degraded by noise.

Figure 14.11 shows median skill scores across twelve runs for each combination of SNR and n-by-n receptor grid. Recall that lower skill scores denote better solutions as described in Appendix. The median is used instead of the mean, because the median is less sensitive to outliers and is more indicative of what to expect in a single run. Figure 14.11 shows the results for multiplicative noise; results for additive noise are similar. The figure shows that the ability of the model to compute the correct solution is not significantly affected as long as the magnitude of the signal is greater than the magnitude of the noise (i.e. SNR > 1). For SNR $= 1$ where the signal and noise are of equal magnitude, the model performs slightly better with more data beyond an 8-by-8 grid. Performance at this point has deteriorated, however, as indicated by the sharp skill score gradient between SNR $= 10$ and SNR $= 1$. For runs with more noise than signal (SNR < 1), the GA is unable to compute the solution to any acceptable degree of accuracy.

Recall that in the synthetic data runs with no noise, the NMDS algorithm can further improve the

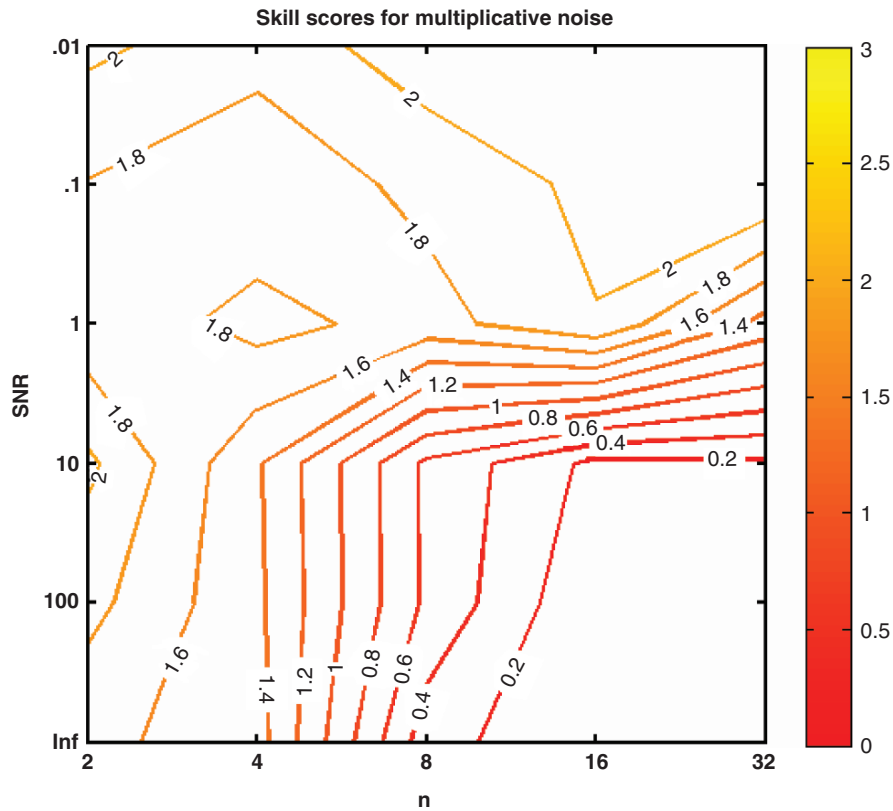

Fig. 14.11 Contour plot of median skill score for various n-by-n receptor grids and signal-to-noise ratios (SNRs) for multiplicative noise. The median skill scores are taken over 12 runs. Lower skill scores denote better solutions

solution found after the 100th GA iteration. In the runs with noise, however, application of the NMDS algorithm after the 100th GA iteration did not appreciably improve the solution. The average skill score of the GA-produced solutions across all SNRs and receptor grids was 1.578, while the average skill score after the application of the Nelder-Mead downhill simplex was 1.582, which is slightly worse. This result is not surprising, because after the receptor data is contaminated with noise, the solution corresponding to the lowest cost function value is usually not the correct solution. While the NMDS method may find a lower cost function value than the GA, the objective skill score does not consider the cost function value, only the specific values of each parameter.

14.5.3 A Parting Look

To help cope with the uncertainty in meteorological data, we have described a method that tunes wind direction and contaminant source characterization simultaneously by using a GA. The model works extremely well for synthetic data given a grid of at least 8-by-8 receptors. A smaller set of receptors, such as a 4-by-4 grid, does not provide enough data to distinguish wind direction from source characteristics. Using synthetic data contaminated with white noise, as long as the magnitude of the noise does not exceed the magnitude of the signal, the GA can still find the wind direction, source location, and source strength fairly well. Increasing the quantity of available data increases the amount of noise the GA can cope with in determining the approximate solution.

For demonstration purposes, the meteorological tuning experiments presented here use a Gaussian plume equation to calculate expected downwind concentrations in order to reduce the computational complexity. For a real data application, a more sophisticated dispersion model can provide a closer match to the receptor data than the Gaussian plume equation. The current model configuration, however, requires a new set of dispersion calculations in each GA iteration, so the direct use of a more complex model would impose substantial computational cost.

14.6 Incorporating Realism: SCIPUFF and Field Test Data

We have shown that the GA-coupled model can determine the source characteristics for pollutant emissions using synthetic data produced by the Gaussian plume equation. We now wish to use the coupled model as a source characterization tool in the context of an operational dispersion model and real data. Thus, we replace the Gaussian plume equation (14.3) with a more sophisticated dispersion model, SCIPUFF. This new coupled model can then be tested with real contaminant data. If successful, such a coupled model could be useful in determining the source characteristics for those hazardous release events where monitored contaminant concentrations are available. For this section, we also assume that the meteorological data are known. The general architecture of the GA coupled model of this section returns to that described in Section 14.2.

14.6.1 Adding SCIPUFF as the Dispersion Model

The primary upgrade to the GA coupled model of Section 14.2 is the replacement of the Gaussian plume equation (14.3) with the much more sophisticated SCIPUFF dispersion model (Sykes et al. 1998). As the forward component of the coupled model, SCIPUFF calculates the contributions from each potential source. These contributions are represented by matrix **C** in (14.1).

SCIPUFF, the *S*econd-order *C*losure *I*ntegrated *PUFF* model, is an ensemble average transport and dispersion model that computes the field of expected concentrations resulting from one or more sources at multiple times. The model solves the transport equations using a second-order closure scheme, and treats releases as a collection of Gaussian puffs (Sykes et al. 1986; Sykes and Gabruk 1997). SCIPUFF can be used for dispersion applications requiring expected concentrations of source material. SCIPUFF is a suitable choice for insertion in our GA coupled model because of its ability to compute expected concentrations over predefined time periods for any number of sources, and the ease in integrating its output into matrix **C** of (14.1).

SCIPUFF is run once for each potential source considered by the coupled model. The output from each SCIPUFF run corresponds to a particular column in the **C** matrix. This use of SCIPUFF does not impose substantial computational cost, because the SCIPUFF runs only need to be executed once prior to the GA initialization and not in every GA iteration.

The parameters of the GA return to those of Section 14.2, with a population size of 8, mutation rate of 0.2, crossover rate of 0.5, and the same cost function (14.2).

14.6.1.1 Validation with SCIPUFF

To gauge the impact of upgrading the GA coupled model's forward component, the validation technique performed in Section 14.4 using the Gaussian plume equation is repeated for the coupled model incorporating SCIPUFF. The validation consists of model runs using synthetic data produced by SCIPUFF. Subsequent tests, including validation with real data, can then be performed with confidence that any issues encountered are not related to incorporating SCIPUFF into the coupled model.

For the validation phase, SCIPUFF was used to create synthetic receptor data, representing matrix **R** in (14.1). The synthetic receptor data are instantaneous contaminant concentrations at a previously defined receptor location 5 m above the surface, with each time-dependent observation corresponding to one value of **R**. Sets of synthetic data corresponding to particular source configurations were created using a synthetic two-dimensional wind field using the same circular geometry used in Section 14.3.1, with 32 independent meteorological periods (here, hours) and 16 potential sources. The validation runs also use the same synthetic meteorological data as in Section 14.3.1. The validation uses a 100-run Monte Carlo simulation for each set of synthetic data as done in Section 14.4. Three source configurations were analyzed with similar results; this section focuses on a spiral configuration with varying source strength ($\mathbf{S} = [0, 1, 2, 3, \ldots, 0, 1, 2, 3]^T$). Allen et al. (2006) provides results for the other source configurations along with more detailed analysis.

Table 14.7 shows the means and standard deviations for 4 of the 16 sources, each corresponding to a different **S** value (0, 1, 2, or 3). All of the means are very

Table 14.7 Means and standard deviations for four sources in the spiral configuration setup across 100 Monte Carlo runs

	Source 1	Source 6	Source 11	Source 16
Calibration factor	0	1	2	3
Mean	0.0142	0.9996	2.0004	2.9998
Std Dev	0.0145	0.0142	0.0193	0.0167

close to the known solutions (with the slight exception of source 1), and all of the standard deviations are less than 0.02. The mean for source 1 is further from the solution than for the other sources because the GA imposes a lower bound of 0 on the solutions. Overall, the GA does an exceptional job of approaching the solution, not just in terms of the mean across all 100 Monte Carlo runs, but also for single runs, as shown by the small values of the standard deviations.

As in Section 14.4, additional Monte Carlo simulations were run using synthetic data contaminated with noise. The noise simulates the impact of imprecise monitoring data, errors in the meteorological data, and the disparity between the ensemble average nature of the model as compared to data from a specific realization. Figure 14.12 summarizes the results for the spiral configuration using multiplicative noise. The figure shows the GA-computed **S** for 4 of the 16 sources as a function of the logarithm of the signal-to-noise ratio. These four sources are representative of each of the four different solutions. Dashed error bars signify plus and minus one standard deviation from the mean, and the dotted error bars represent the 90% confidence interval. A detailed discussion of the results can be found in Allen et al. (2007).

The graphs and results from other source configurations (not shown) are quite similar. This suggests that the choice of dispersion model used within the coupled model does not affect the performance of the GA in obtaining the optimal solution, allowing models of increasing complexity to be used in the coupled model with no performance-related side effects. Computing time depends more on the GA than the dispersion model, because the dispersion model is only run once for each source, further supporting the use of a dispersion model of any level of complexity within the coupled model. Of course, this does not mean

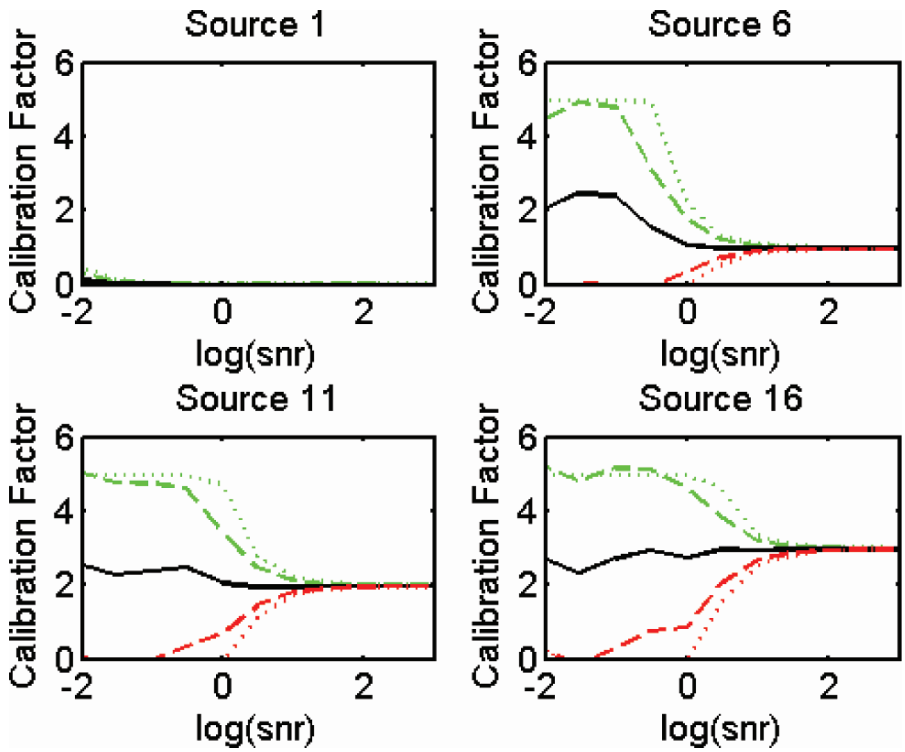

Fig. 14.12 Calibration factor as a function of the signal-to-noise ratio (SNR) for the spiral source configuration using multiplicative noise. Four sources with different S values are shown: Source 1 (S = 0), Source 6 (S = 1), Source 11 (S = 2), and Source 16 (S = 3). Mean (solid), standard deviation (dashed), and 90% confidence interval (dotted) are shown

incorporating SCIPUFF into the coupled model does not upgrade the performance in a general sense, but only that it does not increase the programming complexity or hinder the performance of the GA.

14.6.2 Verification with Monitored Data

Now that we have validated the GA coupled model incorporating SCIPUFF, we can conduct coupled model runs using real data – specifically, neutrally buoyant tracer concentration data from the Dipole Pride 26 (DP26) field tests. These runs are used to demonstrate the model's ability to characterize pollutant sources correctly despite the stochastic scatter of realizations around the forecast ensemble mean.

The DP26 field experiments took place in November 1996 at the Nevada Test Site (Biltoft 1998). The tests released sulfur hexafluoride (SF6), a passive tracer, at locations nearby a domain of 90 receptors. Seventeen different field tests were carried out during the DP26 experiments. Our study only used data from 14 of these tests due to missing data in the other three tests. Figure 14.13 shows the test domain and the orientation of sources and receptors. N2, N3, S2, and S3 are the source locations, and the thick black lines show the approximate receptor locations. Further details on these field experiments can be found in Biltoft (1998) and Watson et al. (1998).

Chang et al. (2003) used the DP26 data to validate various dispersion models, including SCIPUFF. While SCIPUFF performed as well as the other dispersion models they tested, about 50–60% of SCIPUFF-predicted concentrations came within a factor of two of the observations. Most large errors occurred when the modeled puff missed the receptors altogether due to errors in the wind field. To alleviate the effects of these large errors and other issues associated with real data, several changes need to be made to the coupled model architecture.

In order to use data from all 90 receptors, the **C** and **R** matrices in (14.1) are expanded so that r > 1. Because the purpose of the GA-coupled model is to find a single source apportionment vector, **S** providing the best fit across all receptors, the calibration vector **S** remains a one dimensional vector. If the model matches the data perfectly, a single **S** vector would be all 1.0s.

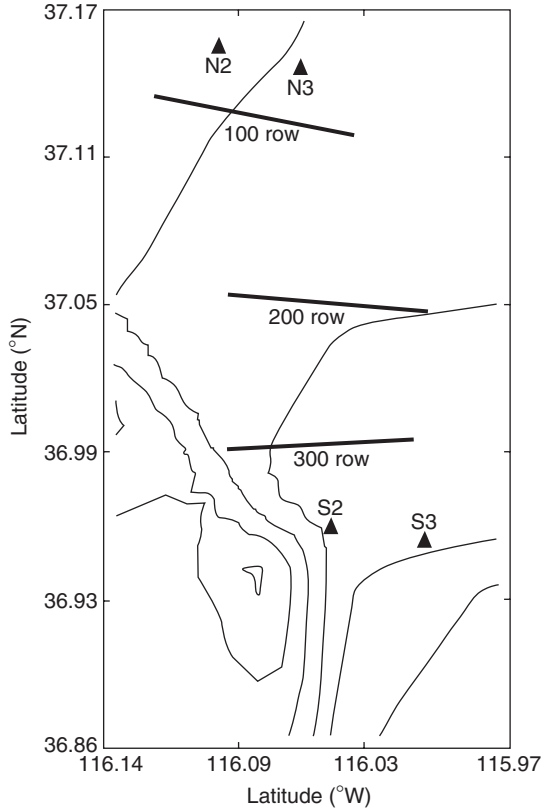

Fig. 14.13 Dipole Pride 26 test domain as represented in coupled model. N2, N3, S2, and S3 are the emission source locations. The thick black lines represent the approximate locations of the receptors (30 along each line). The thin black lines represent terrain contours, corresponding to heights of 1,000–1,519 m. Source: Modeled after similar figures in Biltoft (1998) and Chang et al. (2003).

Two modifications also must be made to the cost function (14.2) – due to the large errors in contaminant magnitude often found in real data applications. First, the cost function incorporates the natural logarithm of the squares of differences as in Section 14.5.

$$\mathrm{RMS} = \frac{\sum_{r=1}^{90}\sqrt{\sum_{m=1}^{M}\left(\log_{10}\left(C_{mnr}\cdot S_n+1\right)-\log_{10}\left(R_{mr}+1\right)\right)^2}}{\sum_{r=1}^{90}\sqrt{\sum_{m=1}^{M}\left(\log_{10}\left(R_{mr}+1\right)\right)^2}} \tag{14.12}$$

Second, it is necessary to add 1.0 to the concentration values before taking the logarithm because

the logarithm of zero is undefined. For this application, the concentration values are typically several orders of magnitude greater than 1.0, so the values are not dwarfed. Section 14.5 discusses a scheme for applications where most concentration values are less than 1.0.

Allen et al. (2007) shows that the logarithmic cost function (14.12) is more effective in determining the source characteristics than the linear cost function (14.2), particularly for finding the source location and emission time of an instantaneous release. This logarithmic variable transformation acts to ameliorate the order of magnitude differences that often arise in concentration data. This behavior is desired because the primary issue in source identification is not strictly the magnitude, but rather the non-zero nature of each source's contribution (i.e. whether or not a source's puff passes over a particular receptor at all). For example, if the receptor data value is 200 parts per trillion (ppt), but the model's predicted concentration is 2,000 ppt, a logarithmic cost function rates the value of 2,000 ppt more highly than a value near 0.0 ppt.

Haupt et al. (2006) and Section 14.3.3 above show that the RMS cost function produced the most efficient convergence. Therefore, all cost functions considered here involve some form of a squared difference. The two normalization schemes discussed here affect model performance, but the specific normalization values used are arbitrary, because the GA mating mechanism is based on ranking rather than absolute difference.

One more issue with real data applications such as DP2 is that it is difficult to characterize non-emitting sources whose potential plumes disperse completely outside the receptor domain. For instance, some potential source may be downwind of the receptors. We deal with this issue by introducing a scale factor that adjusts each source's maximum allowed magnitude. This scale factor sums each column in the **C** matrix representing the pollutant contribution of each source n, and normalizes that sum by the maximum contribution from any source to produce a number ranging from 0.0 to 1.0.

$$\text{scale}(n) = \frac{\sum_{r=1}^{90}\sum_{m=1}^{M} C_{mnr}}{\max\left(\sum_{r=1}^{90}\sum_{m=1}^{M} C_{mnr}\right)} \tag{14.13}$$

The scale factor (14.13) is then multiplied by a predefined upper limit to give the maximum source strength allowed by the GA for each source. Sources that cannot emit into the domain have scale factors of 0.0, forcing the GA to limit these sources' $\mathbf{S}_n$ values to 0.0. This method does not assume any prior knowledge regarding which sources are potential emitters, but does provide objective estimates of each source's potential contribution to the domain. This process eliminates the 50% of the candidate sources that are downwind of the receptor for the Dipole Pride data set.

A possible side effect of using the scale factor is limiting the maximum allowable strength for the correct sources below their actual strengths. To account for this side effect, the range of strengths allowed by the GA should be set beyond the expected range of possible strengths. The range should not be made too large, however, since the run-to-run variability in solutions is proportional to this range. Thus, **S** values are set to range from 0 to 10, increased beyond the original range of 0 to 5.

Several initial runs were made with the GA coupled model using the DP26 data and these coupled model modifications. The goal of these runs was to characterize the emission locations and times (strength characterization is the focus of subsequent sections). These runs used the four emission locations (N2, N3, S2, S3) at two times each, for a total of eight sources. S_n should be equal to 1.0 at the emitting sources (one or two sources in each field test), and 0.0 for all non-emitters, if all else is perfect. In other words, it should detect which source was the actual emitter for each field. In the initial runs, the correct source and time of emission were identified 64% of the time.

14.6.3 Performance Optimization

Now we seek to optimize the performance of the coupled model with the DP26 data set by performing various tests, each designed to determine the impact of different parameters. While the optimization is specific to DP26, many of the results can be applied to the coupled model in general for other data sets.

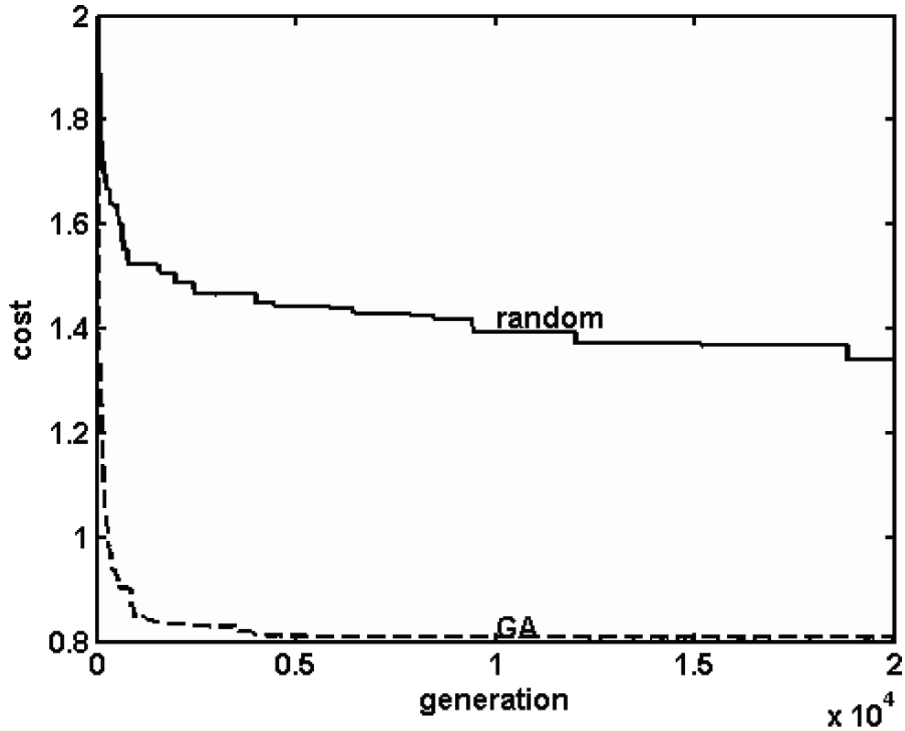

Fig. 14.14 Minimum cost function value as a function of iteration number for the GA (dashed) versus a random search method (solid), carried out to 20,000 iterations

14.6.3.1 GA vs. Random Search

The first test determines if solving the matrix problem requires the GA at all. The GA's performance is compared to the performance of the random search method in Fig. 14.14, which shows the minimum cost for one of the DP26 tests, as found by the GA (dashed) and the random search (solid), averaged over five runs, each with 20,000 iterations. While the "number of iterations" is specific to the GA, the corresponding computing time for the random search method is normalized to be equivalent to the number of GA iterations, so that the graph provides a fair comparison. The random search clearly took much longer to find a solution with a sufficiently low cost function value. In fact, out to 20,000 iterations, the random search never caught up to the GA while the GA converged to the optimal solution in about 7,000 iterations. This result shows that a random search is inefficient, and that a more sophisticated optimization method such as a GA is required.

14.6.3.2 Population Sizes and Mutation Rates

Section 14.3.4 presented a sensitivity study on GA population sizes and mutation rates using synthetic data and found that two combinations of sizes and rates were most efficient in converging to the solution: high population sizes coupled with relatively low mutation rates, and low population sizes coupled with high mutation rates. To determine if the same conclusion applies to a real-data application, a similar sensitivity study is made using the DP26 data set using 5 of the 14 field tests. The goal is to determine which combination of population size and mutation rate minimizes the number of cost function evaluations required for convergence to a correct solution.

Figure 14.15 shows the number of cost function calls required for 80 combinations of population sizes and mutation rates, averaged across five runs for each field test. The optimal mutation rate was 0.15, and the optimal population size was in the range of 4 to 12, similar to the results in Section 14.3.4. Unlike the

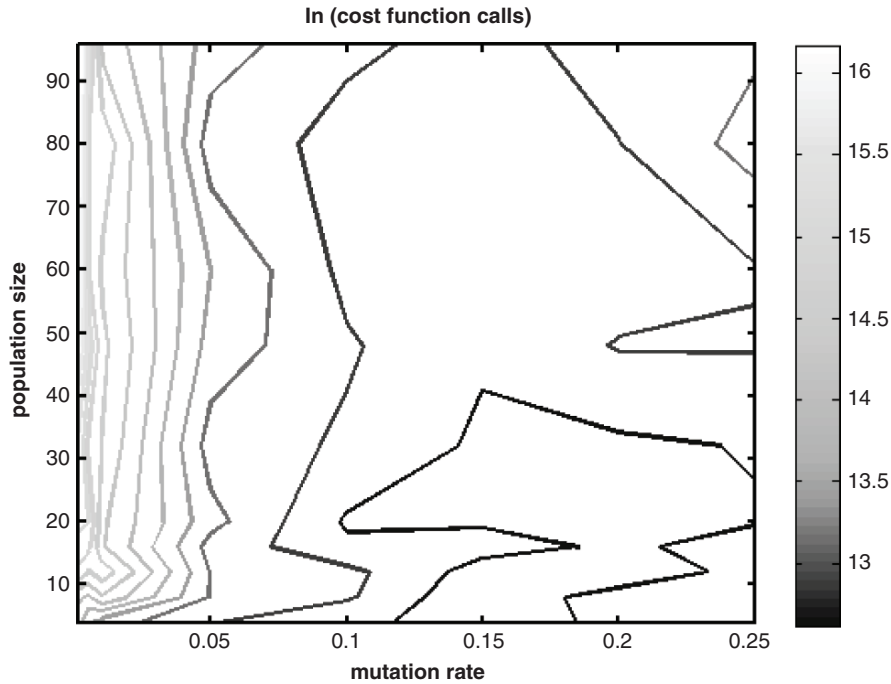

Fig. 14.15 Contour plot of average number of cost function calls versus mutation rate and population size for the coupled model. Darker contours correspond to fewer cost function calls. The number of cost function calls has been normalized by the natural log for viewing purposes

case in Section 14.3.4, however, a high population size coupled with a low mutation rate was not efficient in finding the solution. A possible reason for this difference is that these runs were conducted with a candidate source array of size 8 (N2, N3, S2, and S3 at two times each). The scale factor eliminates four of the sources, leaving four. With only four significant GA parameters in the chromosome, mating is less effective than mutation for finding progressively better solutions. Therefore, a relatively high mutation rate coupled with a rather small population size works well for this specific application.

14.6.3.3 Other Studies and Multi-stage Process

Other sensitivity studies were performed, resulting in the following conclusions, which are elaborated on in Allen et al. (2007) and Allen (2006):

- The DP26 data set provides receptor data every 15 min. SCIPUFF can also output values every 15 min; however, the DP26 receptor data are not instantaneous concentrations, but rather time-integrated averages. Shortening the output interval in SCIPUFF to 5 min and then averaging back up to 15 min improves the GA's performance; shortening the output interval beyond 5 min did not further improve solution accuracy.
- It is necessary to include all 90 receptors in the analysis to have the best solution accuracy. Using fewer than 90 receptors improves computing time, but at the expense of less accurate solutions.
- DP26 includes upper-air meteorological data, but the upper-air data was found to have little effect on the source characterizations that are based on surface data only.
- The run-to-run variability in solutions is proportional to the range of values allowed by the GA. As discussed earlier, this range should be larger than the range of all possible strengths because of the scale factor. Because correct sources had scale factors as low as 0.1 in some instances, the range of values allowed by the GA should be an order of magnitude larger than the range of presupposed possible strengths.
- In a typical model run, the source strength is underestimated because the GA attributes small amounts of pollutant to non-emitting sources, thus decreasing the calculated strength at the correct source.

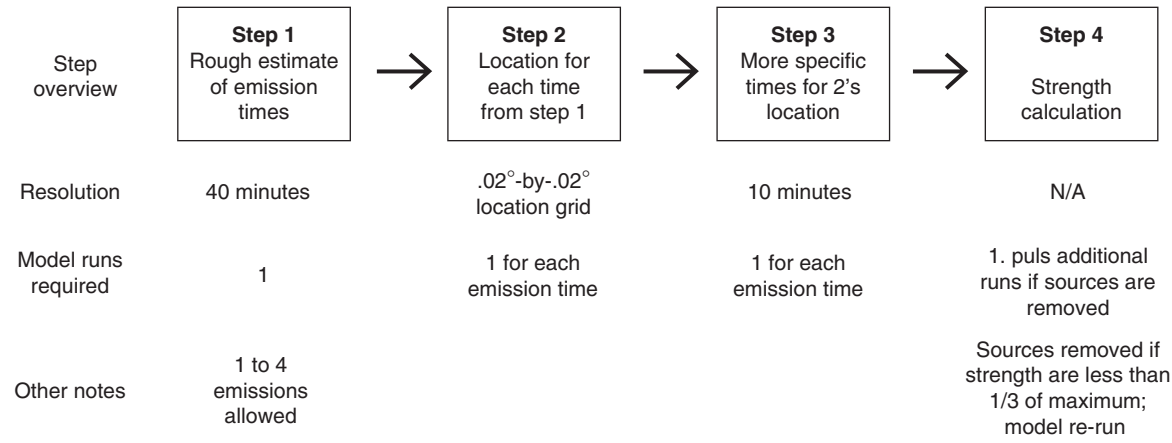

Fig. 14.16 Illustration of the steps in the multi-stage process for source characterization

A more accurate source strength is obtained in a model run by including only the correct source(s) in the candidate source array, thus forcing the GA to attribute all pollutant to only the correct source(s).

These observations helped us to produce an automated multi-stage process that best characterizes the location, time, and strength of the pollutant source(s) assuming as little *a priori* information about the sources as possible. Figure 14.16 provides an outline of the multi-stage process. The first stage uses a coarse grid designed to estimate the number of emission sources and time of each emission. The second stage performs a separate coupled model run for each emission time found in the first iteration. The third stage pinpoints the most probable emission time for each release by running the coupled model once for each emission with the locations found in the second stage in the source array. The final stage then calculates the strength of each emission. The final result is a list of emission locations, times, and strengths.

14.7 Summary and Prospects

This chapter has followed the development of an AI technique to solve a real world problem. We are attempting to identify and characterize a source of contaminant in spite of imprecise knowledge of source location, emission rate, and time of release; uncertain and changing meteorological conditions; monitoring errors; and the inherent uncertainty of turbulent transport and dispersion. In spite of these formidable problems, our GA coupled model is shown to work rather well. It was developed and tested using synthetic identical twin experiments and contrived geometries to test the limits and tune the method (Section 14.3.1). A realistic geometry (Section 14.3.2) revealed some additional limitations along with the successes. We saw that certain geometries could produce ill-posed problems for that model formulation. We also studied GA performance by varying the GA parameters of population size and mutation rate. For this complicated cost surface, the mutation operator is critical for GA convergence.

Next, a statistical analysis of model performance using Monte Carlo runs revealed that the average of multiple runs produces an even better solution. In spite of applying either additive or multiplication white noise to simulate the highly uncertain model environment, the GA coupled model is successful in the Monte Carlo framework, even when the noise level is of the same magnitude as the signal. As noise overshadows the signal, however, confidence in the solution degrades.

What about situations where either no meteorological data are available or the wind data are not representative of local conditions? Section 14.5 demonstrates an extended GA model to simultaneously search for source location, emission rate, and wind direction. A different mating scheme is required for this application, which leads to quite different choices for GA parameters of population size and mutation rate. This GA application combined with a more traditional

NMDS method speeds convergence once the GA has found the correct solution basin for the simplex starting point. Note that this traditional gradient-based method did not work well without the GA to provide that first guess.

These initial demonstrations were done in the context of a very basic Gaussian plume T&D model. To incorporate more realism into the dispersion process, Section 14.6.1 replaced the Gaussian plume with the refined SCIPUFF model. The GA coupled model can be validated much the same as the original version. The most realistic test was accomplished on data from the Dipole Pride 26 field test with all its inherent errors and uncertainties. Note that prior modeling studies had difficulty matching the T&D for these data. The GA model still showed success on some of the trials.

Subsequent work has used similar techniques to back-calculate up to seven modeling parameters: two-dimension location, emission height, source strength, time of release, wind direction, and wind speed (Long et al. 2008; Long 2007). A mixed integer genetic algorithm is able to characterize atmospheric stability (Haupt et al. 2008).

The GA coupled model is not perfect. Neither is any other model attempting to solve this difficult source characterization problem. The exercise does demonstrate, however, that an AI-based technique is competitive for solving a real world environmental problem.

Acknowledgements We thank Joseph Chang for providing the Dipole Pride 26 data, Ian Sykes for helpful information on SCIPUFF, and Kerrie Long for many helpful suggestions.

Appendix: Skill Scores

For the evaluation of results in the simultaneous tuning of surface wind direction and source characterization, an objective skill score is required to evaluate the proximity of solutions to the actual solution. The skill score used here is designed to weight the error in wind direction, source strength, and source location equally. The errors in each parameter are normalized to a [0,1] scale, with a score of 0 given to exact solution, and a score of 1 when inaccuracy exceeds a predefined upper bound. These scores are then added up to give a final score from 0 to 3, with a score of 0 for an exact solution.

The formulas for the three skill score components are:

$$S_{wind} = \ln\left(|\theta_{GA} - \theta_{act}| + 1\right)/5.199 \tag{14.14}$$

$$S_{str} = \max\left(\left(\frac{S_{GA}}{4 * S_{act}} - \frac{1}{4}\right), \left(\frac{S_{act}}{4 * S_{GA}} - \frac{1}{4}\right)\right) \tag{14.15}$$

$$S_{loc} = 1.0746 * \left(-\exp\left(-dist/1500\right) + 1\right) \tag{14.16}$$

where θ_{GA} is the wind direction found by the GA, θ_{act} is the actual wind direction, S_{GA} is the source strength found by the GA, S_{act} is the actual source strength, and *dist* is the distance from the GA-computed source location to the actual source location in meters.

The constants in these equations were computed to give the desired scores for various solutions – specifically, to give a score of 0 for an exact solution, and a score of 1 for a solution at or above a predefined threshold (180° for wind direction, a factor of 5 for the source strength, and 4,000 m for the source location). For example, in the wind direction equation (14.14), *ln*(181) is approximately equal to 5.199, so the constant 5.199 results in a score of 1 for the highest possible error of 180°. For each equation, if the computed value exceeds 1, the value is truncated to 1. The final skill score is $S_{wind} + S_{str} + S_{loc}$, where 0 is a perfect score, and 3 is the worst possible score.

References

Allen, C. T. (2006). *Source characterization and meteorological data optimization with a genetic algorithm-coupled dispersion/backward model*. M.S. thesis, Department of Meteorology, The Pennsylvania State University, p. 69.

Allen, C. T., Haupt, S. E., & Young, G. S. (2007). Source characterization with a receptor/dispersion model coupled with a genetic algorithm. *Journal of Applied Meteorology and Climatology*, *46*, 273–287.

Allen, C. T., Young, G. S., & Haupt, S. E. (2006). Improving pollutant source characterization by optimizing meteorological data with a genetic algorithm. *Atmospheric Environment*, *41*, 2283–2289.

Beychok, M. R. (1994). *Fundamentals of stack gas dispersion* (3rd ed., p. 193). Irvine, CA: Milton Beychok.

Biltoft, C. A. (1998). Dipole Pride 26: Phase II of Defense Special Weapons Agency transport and dispersion model validation. DPG Doc. DPG-FR-98-001, prepared for Defense Threat Reduction Agency by Meteorology and Obscurants

Divisions, West Desert Test Center, U.S. Army Dugway Proving Ground, Dugway, UT, 76 pp.

Camelli, F., & Lohner, R. (2004). Assessing maximum possible damage for contaminant release event. *Engineering Computations*, *21*, 748–760.

Cartwright, H. M., & Harris, S. P. (1993). Analysis of the distribution of airborne pollution using GAs. *Atmospheric Environment*, *27A*, 1783–1791.

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.

Daley, R. (1991). *Atmosphericdata analysis*. Cambridge, UK: Cambridge University Press.

Haupt, R. L., & Haupt, S. E. (2004). *Practical genetic algorithms* (2nd ed., p. 255). New York: Wiley, with CD.

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 and Climatology*, *45*, 476–490.

Haupt, S. E., Haupt, R. L., & Young, G. S. (2008). A mixed integer genetic algorithm used in Chem-Bio defense applications, submitted to *Journal of Soft Computing*.

Kumar, A. V., Patil, R. S., & Nambi, K. S. V. (2004). A composite receptor and dispersion model approach for estimation of effective emission factors for vehicles. *Atmospheric Environment*, *38*, 7065–7072.

Long, K. J. (2007). *Improving contaminant source characterization and meteorological data forcing with a genetic algorithm*. Master's thesis, The Pennsylvania State University.

Long, K. J., Haupt, S. E., & Young, G. S. (2008). Improving meteorological forcing and contaminant source characterization using a genetic algorithm, to be submitted to *Optimization and Engineering*.

Loughlin, D. H., Ranjithan, S. R., Baugh, J. W., Jr., & Brill, E. D., Jr. (2000). Application of GAs for the design of ozone control strategies. *Journal of the Air & Waste Management Association*, *50*, 1050–1063.

National Research Council (2003). *Tracking and predicting the atmospheric dispersion of hazardous material releases. implications for homeland security*. Washington, DC: The National Academies Press.

Nelder, J. A., & Mead, R. (1965). A simplex method for function minimization. *Computer Journal*, *7*, 308–314.

Penrose, R. (1955). A generalized inverse for matrices. *Proceedings of the Cambridge Philosophical Society*, *51*, 406–413.

Qin, R., & Oduyemi, K. (2003). Atmospheric aerosol source identification and estimates of source contributions to air pollution in Dundee, UK. *Atmospheric Environment*, *37*, 1799–1809.

Sykes, R. I., & Gabruk, R. S. (1997). A second-order closure model for the effect of averaging time on turbulent plume dispersion. *Journal of Applied Meteorology*, *36*, 1038–1045.

Sykes, R. I., Lewellen, W. S., & Parker, S. F. (1986). A gaussian plume model of atmospheric dispersion based on second-order closure. *Journal of Applied Meteorology*, *25*, 322–331.

Sykes, R. I., Parker, S. F., Henn, D. S., Cerasoli, C. P., & Santos, L. P. (1998). PC-SCIPUFF version 1.2PD technical documentation (ARAP Rep. 718). Princeton, NJ: Titan Research and Technology Division, Titan Corporation, 172 pp.

Watson, T. B., Keislar, R. E., Reese, B., George, D. H., & Biltoft, C. A. (1998). The Defense Special Weapons Agency Dipole Pride 26 field experiment (NOAA Air Resources Laboratory Tech. Memo. ERL ARL-225), 90 pp.

Wyngaard, J. C. (1992). Atmospheric turbulence. *Annual Review of Fluid Mechanics*, *24*, 205–233.