1.2 VALIDATION OF RECEPTOR/DISPERSION MODEL COUPLED WITH A GENETIC ALGORITHM

Sue Ellen Haupt *
George S. Young
The Pennsylvania State University, State College, PA

1. INTRODUCTION

Air pollution models can be divided into two primary categories: receptor models and dispersion models. Receptor models are formulated to begin with pollutant information monitored at a receptor and to look backward, using data on several species and information about relative concentrations of those species from possible sources, to apportion the pollutant to the sources. (Miller, et al 2002) In contrast, chemical transport, or dispersion models start with the source characteristics and use physics, mathematical, and chemical calculations to predict pollutant concentration at some distance from the source. Important input for those dispersion models includes information about the emissions from the source, the local atmospheric conditions, and some geographical characterization. Both types of models have been highly developed and forms of them are widely used for prediction and diagnosis of events. (EPA 2003)

Combining the current technology of the forward-looking transport and dispersion models with backward-looking receptor models would enable the apportionment of monitored data to their sources and estimate the uncertainty involved. Such a coupled model would combine the physical basis of the dispersion calculations with actual monitored pollutant concentrations. In previous work, we coupled a simple dispersion model with a chemical mass balance (CMB) receptor model using a genetic algorithm to optimize source apportionment factors in order to determine the sources of monitored pollutant (Haupt and Haupt 2004, Haupt 2004a,b).

Corresponding author address: Sue Ellen Haupt, Applied Research Laboratory, P.O. Box 30, Pennsylvania State University, State College, PA 16804;
e-mail: haupts2@asme.org

A few investigators have previously used information on dispersion or chemical transport in computing source apportionment. Qin and Oduyemi (2003) began apportioning sources of PM_{10} with a receptor model, then augmented it with dispersion model predictions from vehicle emission sources. Cartwright and Harris (1993) used a genetic algorithm (GA) to apportion sources to pollutant data at receptors. The work of Loughlin, et al (2000) coupled an air quality model with receptor principles using a GA to design better control strategies to meet attainment of the ozone standard while minimizing total cost of controls at over 1000 sources. This current work goes beyond these prior efforts by using dispersion equations to predict potential pollutant concentration, then optimizing the apportionment factors to best match monitored data. Using artificial intelligence techniques, in this case a genetic algorithm, is the key to successfully computing the apportionment factors.

In the prior work (Haupt and Haupt 2004, Haupt 2004a,b) we saw that for circularly symmetric source/receptor configurations, a coupled model can correctly identify a single source. It is also proficient in identifying some combination of sources contributing to the total pollutant monitored at a receptor. In those studies the synthetic data was created using the same dispersion model and meteorological data as used in the coupled model for source apportionment. In addition, for an actual source/receptor configuration run with synthetic emissions and meteorological data, the coupled model did well at identifying a single source. There is more difficulty, however, in correctly identifying multiple sources. It is easy to misidentify sources at a significant distance from the receptor or when several sources are at the same angle upwind of the receptor. Here we seek to further

validate this coupled approach through careful analysis with synthetic data and a bootstrap technique to define error bars and confidence intervals. The synthetic data is constructed to examine the issue of accuracy in the presence of noise and the variation of accuracy with distance between the source and receptor.

2. PROBLEM FORMULATION

The chemical mass balance (CMB) receptor model is often used to apportion monitored concentrations received at receptors to the potential sources. It requires receptor data of different monitored species and known emission fractions for each of those species from a number of sources. The CMB model can be written as:

$$C \bullet S = R \tag{1}$$

where C is the source concentration matrix, which denotes the fractional emission of each species from a given source; R is the concentration of each species measured at a given receptor, and S is the unknown apportionment vector. A fit to the data produces the fraction contribution from each source, S. Our coupled approach replaces the emission fractions in C with concentrations predicted by a transport model for multiple time periods. Similarly, R now represents the monitored concentrations for those same time periods. Note that these time periods are often averages of multiple shorter time periods. Thus S becomes a calibration factor linking the dispersed emissions as predicted by the transport model for an ensemble of time periods with the actual concentrations monitored at the receptor. The modeled concentrations are time dependent, on a timescale that matches the meteorological variations. The receptor data is also time dependent, but on a timescale that matches the monitoring sample length. The source calibration vector (S) must be optimized to account for the time varying weather and emission rates. In addition, a large number of sources of error creep into both the predicted and monitored data. All of these errors will become part of the calibration factor.

For pollutant at *N* sources dispersed over *M* time periods, matrix equation (1) can be shown in expanded form:

$$\begin{bmatrix} C_{11} & C_{12} & \cdots & C_{1n} & \cdots & C_{1N} \\ C_{21} & & & & & \\ \vdots & & & & & \\ C_{m1} & & & \ddots & & \\ \vdots & & & & & \\ C_{m1} & & & & & C_{MN} \end{bmatrix} \begin{bmatrix} S_1 \\ S_2 \\ \vdots \\ S_n \\ \vdots \\ S_N \end{bmatrix} = \begin{bmatrix} R_1 \\ R_2 \\ \vdots \\ R_m \\ \vdots \\ R_M \end{bmatrix} \tag{2}$$

where: C_{mn} = the modeled contribution of source *n* at time period *m*

S_n = the unknown calibration factor for source *n*

R_m = the monitored particulate concentration at the receptor for time period *m*

As long as $M \geq N$, S can be computed by standard techniques (matrix inversion if $M = N$ or optimization otherwise). However, as demonstrated in the following section, it is not unusual for the matrix equation to become poorly conditioned, thus requiring more complex solution techniques.

For a simple demonstration of the coupling technique, the dispersion model used to compute the elements in the modeled concentration matrix C will be Gaussian plume dispersion:

$$C_{mn} = \frac{Q_{mn}}{u\sigma_z\sigma_y 2\pi} \exp\left(\frac{-y_{mn}{}^2}{2\sigma_y{}^2}\right) \left[\begin{array}{l} \exp\left(\frac{-(z_r - H_e)^2}{2\sigma_z{}^2}\right) \\ + \exp\left(\frac{-(z_r + H_e)^2}{2\sigma_z{}^2}\right) \end{array} \right] \tag{3}$$

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

(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

H_e = effective height of the plume centerline above ground

σ_y, σ_z = standard deviations of the concentration distribution in the y and z directions, respectively.

For our simple example, we'll compute the standard deviations following Beychok (1994).

$$\sigma = \exp\left[I + J(\ln(x) + K(\ln(x))^2 \right] \qquad (4)$$

where x is the downwind distance (in km) and $I, J, \text{and } K$ are empirical coefficients dependent on the Pasquill Stability Class, which depends on wind speed, direction, and insolation and can be looked up in tables (Beychok 1994). The concentrations computed in this manner from each source form the C_{mn} in equation (2).

The source calibration factor serves to maximize agreement between the transport model and the receptor observations. When multiplied by the source contribution predicted by the transport model, it serves to attribute a certain percentage of the monitored concentration to that source. That factor can also be interpreted as an error or uncertainty in the modeling process in comparison to the monitored data. This uncertainty comes from the input data and from the modeling process itself. The primary sources of error could be characterized as: 1. the source emission rate; 2. the accuracy and representativeness of the meteorological input, both in terms of directly measured variables such as wind speed and direction, as well as in derived quantities such as mixing height (representing the boundary layer depth) and atmospheric stability characterization; 3. the model's characterization of the atmospheric dispersion and chemical transformations; 4. not correctly modeling the stochastic fluctuations due to turbulence; and 5. errors in the monitoring data.

3. SOLUTION METHOD

3.1 Justification

The remaining issue is how to best optimize the fit between the modeled dispersion and the monitored receptor data. This fit involves computing the best calibration factor, S. For square matrices, we should be able to solve the matrix problem directly. Our experience, however, indicates that the matrices are often poorly conditioned. Because there are often more meteorological conditions available, it makes sense to use M>N where possible to aid the optimization. In such cases, least square optimization solutions are possible. We find that such solutions are not always accurate due to the poor conditioning of the matrices. For example, let's consider a scenario using the geometry for Cache Valley, Utah (see Haupt 2004a,b) combined with synthetic equal emission rates for each of 16 sources. Synthetic receptor data was created assuming 64 independent meteorological conditions representing 16 different wind directions (every 22.5 degrees) and four different wind speeds. The synthetic receptor data was created assuming a source apportionment matrix composed of a 1. for the first source and 0. for all remaining sources. The resulting dispersion matrix, C, is poorly conditioned with a condition number of 2.561×10^{-26}. Given that the data is synthetic, we know the actual solution, and can compute the root mean square (RMS) difference from the known solution. For this scenario, the least squares results are rather disappointing, showing an RMS error of 8.26. Thus, a more robust technique is required. Here we choose to use a genetic algorithm to optimize the linear system. For this synthetic scenario, a GA was run 100 times to observe the typical range of solutions expected. The minimum RMS error was 0.39, with the average RMS error of the 100 runs being 0.60. Thus, for such difficult scenarios the GA is more robust at finding the correct solution.

Another issue in deciding the optimization method used here is the consideration of future uses of this methodology. One use is expected to be identifying sources of uncertainty. To do that will require a technique that can handle IF,THEN constructs in the cost (or objective) function. The AI methods are

formulated to be oblivious to the intricacies of the cost function and tend to be quite robust at finding an optimal solution. The remaining applications presented here all use genetic algorithms (GAs) to perform the optimization.

GAs are well suited to many optimization problems where more traditional methods fail. Some of the advantages of the GA for this problem include that they

- Don't require derivative information,
- Simultaneously search from a wide sampling of the objective function surface,
- Deal with a large number of parameters,
- Are well suited for parallel computers,
- Optimize parameters with extremely complex objective function surfaces,

Such advantages outweigh the GAs' lack of rigorous convergence proofs and speed. In addition, since the GA is based on generating random numbers, each solution will be slightly different. Thus, the GA is run repeatedly to gather statistics on the solution. The best solution of repeated runs is often chosen. Other practitioners prefer to report the mean solution of an ensemble of GA runs.

3.2 The Continuous Genetic Algorithm

The many breeds of GA are discussed in detail in Haupt and Haupt (2004). Here we apply a continuous parameter GA, that is, one in which the parameters are real numbers. The flow chart in Figure 1 provides a "big picture" overview of a continuous GA. The parameters are the genes which are strung together in a one-dimensional array known as a chromosome. The GA begins with a population of chromosomes which are fed to the cost function for evaluation. The fittest chromosomes survive while the highest cost ones die off. This process mimics natural selection in the natural world. The lowest cost survivors mate. The mating process combines information from the two parents to produce two offspring. Some of the population experiences mutations.

As seen in the figure, the first step of a continuous parameter genetic algorithm is creating the population of chromosomes. First, the real parameters are concatenated together into a chromosome as:

$$chromosome = \lfloor p_1 p_2 \cdots p_\alpha \cdots p_{N_{par}} \rfloor \quad (5)$$

where the p_i are the parameters and there are a total of N_{par} parameters. The parameters are simply floating point numbers. The encoding function needs only keep track of which digits represent which parameters and to make sure they are within given bounds. A population of such chromosomes is created using a random number generator so that the chromosome arrays are gathered together in a two dimensional matrix. Once the chromosomes have been created, their cost or fitness must be evaluated. This is done by the cost or objective function, which is very problem specific. The lowest cost chromosomes (N_{keep}) remain in the population while the higher cost ones are deemed less fit and die off. The reduced population is then the portion of the population available for mating.

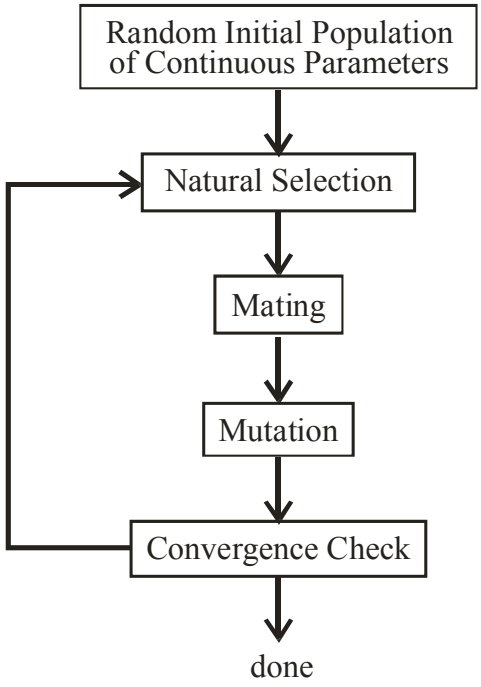

Figure 1. Flowchart of continuous parameter genetic algorithm.

There are a variety of methods to pair the chromosomes for mating. Some popular methods are reviewed by Haupt and Haupt (2004). Here, we choose to pair the chromosomes according to numerical rank. After the cost function evaluation, the chromosomes are sorted in order from lowest

cost to highest. That is, the nth chromosome will have a probability of mating of:

$$P_n = \frac{N_{keep} - n + 1}{\sum_{k=1}^{N_{keep}} k} \quad (6)$$

The cumulative probabilities are used for selecting which chromosomes mate.

Once two parents are chosen, some method must be devised to produce offspring that are some combination of these parents. Many different approaches have been tried for crossing over in continuous parameter genetic algorithms (Adwuya 1996, Haupt and Haupt 2004).

The method used here is a combination of an extrapolation method with a crossover method. It begins by randomly selecting a parameter in the first pair of parents to be the crossover point.

$$\alpha = roundup\{random \times N_{par}\} \quad (7)$$

Wel let

$$\begin{aligned} parent_1 &= \lfloor p_{m1} p_{m2} \cdots p_{m\alpha} \cdots p_{mN_{par}} \rfloor \\ parent_2 &= \lceil p_{d1} p_{d2} \cdots p_{d\alpha} \cdots p_{dN_{par}} \rceil \end{aligned} \quad (8)$$

where the *m* and *d* subscripts discriminate between the *mom* and the *dad* parent. Then the selected parameters are combined to form new parameters that will appear in the children:

$$\begin{aligned} p_{new1} &= p_{m\alpha} - \beta[p_{m\alpha} - p_{d\alpha}] \\ p_{new2} &= p_{d\alpha} - \beta[p_{m\alpha} - p_{d\alpha}] \end{aligned} \quad (9)$$

where β is also a random value between 0 and 1. The final step is to complete the crossover with the rest of the chromosome as before:

$$\begin{aligned} offspring_1 &= \lfloor p_{m1} p_{m2} \cdots p_{new1} \cdots p_{dN_{par}} \rfloor \\ offspring_2 &= \lceil p_{d1} p_{d2} \cdots p_{new2} \cdots p_{mN_{par}} \rceil \end{aligned} \quad (10)$$

If the first parameter of the chromosomes is selected, then only the parameters to right of the selected parameter are swapped. If the last parameter of the chromosomes is selected, then only the parameters to the left of the selected parameter are swapped. This method does not allow offspring parameters outside the bounds set by the parent unless β is greater than one. In this way, information from the two parent chromosomes is combined a way that mimics the crossover process during meiosis.

If care is not taken, the genetic algorithm converges too quickly into one region of the cost surface. If this area is in the region of the global minimum, that is good. Some functions, however, have many local minima and the algorithm could get stuck in a local well. If we do nothing to solve this tendency to converge quickly, we could end up in a local rather than a global minimum. To avoid this problem of overly fast convergence, we force the routine to explore other areas of the cost surface by randomly introducing changes, or mutations, in some of the parameters. A mutated parameter is replaced by a new random parameter.

3.3 Application to Coupled Model

The cost function for coupling a receptor model with a dispersion model was formulated to minimize the difference between the two sides of (1), summed over the total number of meteorological periods considered. This normalized residual is:

$$\text{resid} = \frac{\sqrt{\sum_{m=1}^{M} (C \bullet S - R)^2}}{\sqrt{\sum_{m=1}^{M} (R)^2}} \quad (11)$$

where M is the total number of meteorological periods. Note that each meteorological period might be composed of averages over many shorter periods. Thus, the GA evaluates the summation cost function for each random chromosome of parameters for each iteration. In spite of the large number of cost function evaluations, CPU time remains modest. Moreover, for the poorly conditioned problems we often encounter in real data, we have found that the GA works better at minimizing the matrix equation than competing techniques. This cost is also the metric used

below to compare the performance for different cases.

To run the GA in our 100-run Monte Carlo set most efficiently, we must carefully choose the best GA parameters. The most commonly tuned parameters are the crossover rate, population size, and mutation rate. The population size and mutation rate, in particular, can make a huge difference in the number of cost function evaluations required, and thus the CPU time. Prior experience indicates relatively low population sizes combined with a high mutation rate tend to minimize required CPU time to convergence (Haupt and Haupt 2000, 2004). The runs reported here use population sizes of 8, mutation rate of 0.20 and a crossover rate of 0.5. The calibration factors are assumed to fall in the range of 0 through 5. Note that these values are easy to vary in actual situations and have been chosen to allow reasonable exploration of the solution space applicable for the synthetically generated data of this study. The number of iterations used and convergence are discussed below.

4. MODEL VALIDATION

4.1 Synthetic Data on a Circle

The coupled receptor/dispersion model technique is validated here by testing it on carefully constructed synthetic data. The initial test cases place a receptor at the origin and 16 sources in a circle of radius 500 m spaced every 22.5 degrees (see Figure 2). The source numbering system assigns number 1 to the source 22.5° east of north and proceeds clockwise to source 16 directly north of the receptor. Synthetic receptor data is created using the same dispersion model (3) used for the coupled model optimization. To fit data for 16 sources, we need at least 16 independent meteorological periods. Meteorological data were created to represent wind directions from 16 points of the wind rose and representative wind speeds. All results shown here use stability D for ease of comparison. The dispersion model was run using one hour averaging over the meteorological data and using assumed calibration factors, *S*, that we hope to match with the coupled model.

The coupled receptor/dispersion model was then run with the synthetically generated data. The genetic algorithm, when run with a sufficient number of iterations, will gravitate toward the correct solution. For this problem, the number of iterations determine the smallness of the residual. An initial GA population is based on random numbers, therefore the solution is slightly different for every run. The number of iterations allowed in a run controls how closely the final solution matches the actual solution. Figure 3 demonstrates the GA convergence over 200,000 iterations. We see that the GA continues to minimize the residual throughout the run. Because our routines with a very simple Gaussian dispersion model take little CPU time, the remainder of the paper reports GA runs with 20,000 iterations, roughly where the cost function reaches 10^{-2} in Figure 3. To obtain another order of magnitude in accuracy would require roughly an order of magnitude additional CPU time.

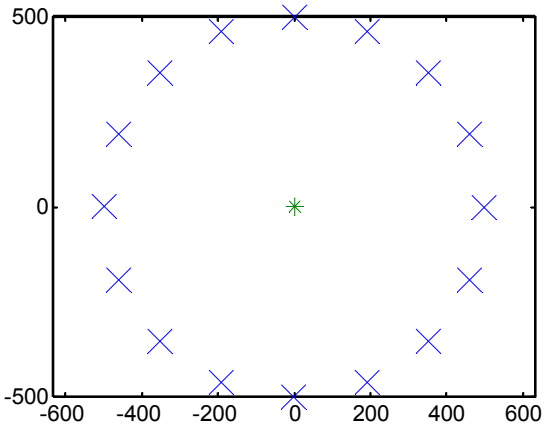

Figure 2. Configuration for circular synthetic data. Sources are denoted by "X" and the receptor as "*".

An initial set of numerical experiments sets the calibration factor, S to a vector of 16 ones. This known vector is used to compute synthetic receptor data. The GA is then run to determine how close it gets to this known solution. Note that the cost function (11) does not presuppose the known solution, rather it acts to minimize the residual of the difference of the two sides of equation (1). A single run of a GA is typically sufficient to estimate the

actual calibration factor to within 2 significant digits for this case.

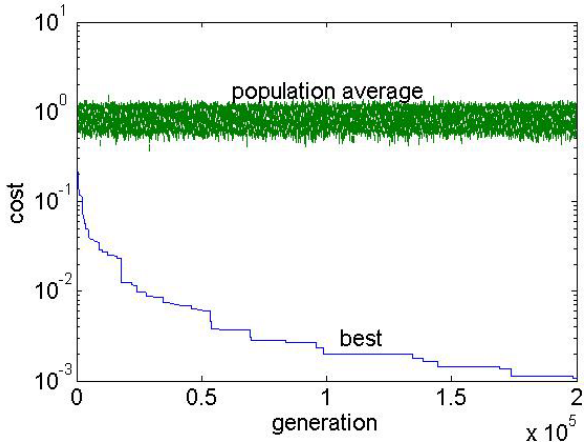

Figure 3. Genetic algorithm convergence for 200,000 iterations. The upper green line denotes the mean error (normalized residual) while the blue lower line is the minimum error denoting the best solution. The minimum error continues to improve throughout the run.

To further analyze the uncertainty of the GA solution methodology, a Monte Carlo technique is used. The GA is run on the same problem 100 times with different initial random seeds. Then we are able to draw error bars and evaluate the accuracy by statistical methods. Figure 4 shows the mean calibration factor at each source found by the GA plus 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.

The mean of 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 run.

Because the source positions and strengths, the meteorological data, and the model are all horizontally homogeneous, we can further aggregate the data from the 16 sources over 100 runs to a pool of 1600 values that approximate an exact value of one. The mean of the 1600 values is 1.0002 and the standard deviation becomes 0.0015109.

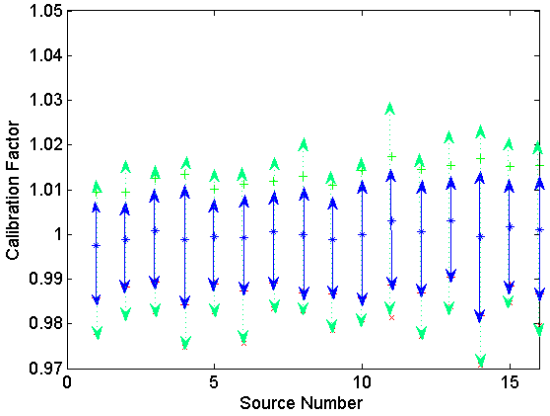

Figure 4. Apportionment of the Calibration Factors to each of the 16 sources for the case of synthetic circular cylinder data.

Are the results above dependent on the fact that the solution for each of the sources is identically one? To assess this issue, we studied this same geometry/meteorological configuration for a second calibration factor, $S = [0,1,2,3,0,1,2,3,0,1,2,3,0,1,2,3]^T$. Table 1 gives the results of 100 GA Monte Carlo runs for this configuration. The width of the error bars is consistent with those found for the prior case with all ones as a solution. Note that the solution is constrained to never go below 0; therefore the standard deviation and confidence limits for the sources with an actual apportionment of 0. are constrained.

Table 1. Mean and standard deviations for each group of sources with actual calibration values of 0, 1, 2, and 3 respectively in a circular geometry as computed for 100 runs.

Source	0	1	2	3
Mean	0.0204	1.0007	1.9989	3.0001
Std dev	0.0014	0.0008	0.0020	0.0011

4.2 Synthetic Data in a Spiral

The circular geometry presented above assumed that all sources are the same distance from the receptor. To investigate relaxing that assumption, we constructed a

second geometry of a spiral configuration, with the sources ranging from 250m through 1750m from the receptor as shown in Figure 5. The assumed calibration factor in this case is $S = [0,1,2,3,0,1,2,3,0,1,2,3,0,1,2,3]^T$. This calibration factor was used with the same meteorological data and constant emission rates to construct synthetic receptor data.

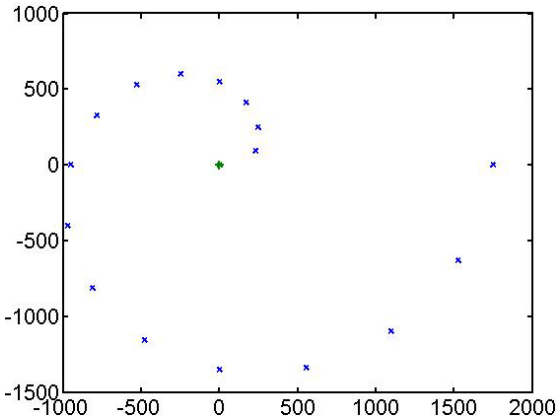

Figure 5. Configuration for spiral synthetic data. Sources are denoted by "X" and the receptor as "*".

The GA was used to optimize the calibration factors for 100 runs. Table 2 summarizes the results. Although the standard deviations are generally larger than for the circular source geometry, they are still quite small.

Table 2. Mean and standard deviations for each group of sources with actual calibration values of 0, 1, 2, and 3 respectively in a spiral geometry as computed for 100 runs.

Source	0	1	2	3
Mean	0.0186	0.9996	2.0014	3.0007
Std dev	0.0054	0.0013	0.0015	0.0022

5. VALIDATION INCLUDING NOISE

Although the results of the previous section are quite heartening, typical data does not have the clear signal available in our synthetically constructed data. Typical situations involve errors in the meteorological data, emissions data, receptor data, as well as the differences between the model and the real atmosphere. Therefore, we simulate the expected error by adding white noise to the data, then using the GA coupled model to optimize the calibration factor.

5.1 Circular Configuration

The first case is again a very simple geometry, the circular geometry with meteorological data representing 16 points of the wind rose. The first series of bootstrapped runs again use assumed source calibration factors of all ones to create the receptor data. In this case, however, white noise with a mean amplitude of 1 is also added to the dispersion model when creating the synthetic receptor data. Thus, the receptor data that goes into R in equation (1) includes as much noise as signal. The GA coupled model is then used to compute the optimal calibration factors.

Figure 6 shows the mean, standard deviation, and 90% confidence interval for 100 runs of the model. The lines depicting the error are much wider spaced than for the case with no noise in Figure 4. Aggregating the full 1600 source cases (16 sources all with actual apportionment value of 1 over 100 runs) produces a mean value of 1.0066, amazingly close to actual value. The mean standard deviation of the 16 sources is 0.02595, larger than for the case without noise but still small enough to give us confidence in model performance with imperfect information.

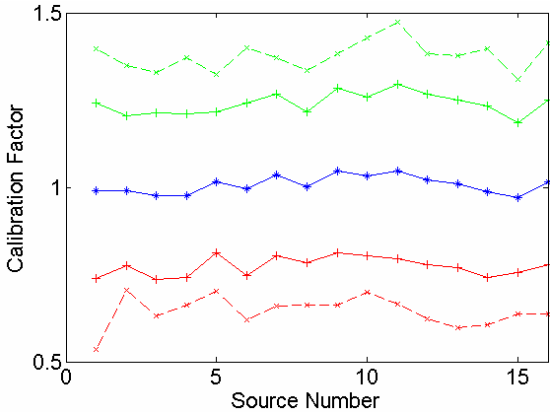

Figure 6. Apportionment of the calibration factors to each of the 16 sources for the case of synthetic circular data and a signal to noise ratio of 1.

Figure 7 depicts the performance of the coupled model over a range of signal to noise ratios (SNRs). This plot aggregates the data over all 16 sources. The center black line is the mean of the 16 sources times the 100 runs, the solid lines on either side depict the standard deviation, and the dashed lines indicate the 90% confidence interval. We see that as long as the log(SNR)>1, the solutions are quite close to the actual solution of 1 and the scatter is quite small. However, as noise becomes greater than the signal (log(SNR)<0) then the computed solution diverges from the actual and the scatter becomes wider. Note that the mean of the solutions is still 1. For log(SNR) = 0.5, the standard deviation and 90% confidence interval has grown. At log(SNR) = 0 the noise equals the signal and we have the case presented in Figure 6 above. As expected, when the noise becomes much larger than the signal on the left side of the plot, the coupled model no longer optimizes the solution reliably. In fact, the mean solution tends to 2.5, which is the mean of the range allowed in the optimization routine. The standard deviation and 90% confidence lines approach the limits of the range.

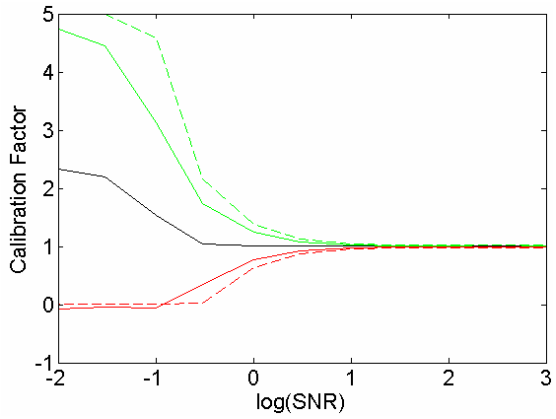

Figure 7. Calibration factors for all sources versus the log of the signal to noise ratio for a circular source configuration.

A second case for analysis is the one with circular geometry and an actual source calibration vector of $S = [0,1,2,3,0,1,2,3,0,1,2,3,0,1,2,3]^T$.

Table 3 lists the mean and standard deviation for each set of sources and an SNR of 1. It is clear that the mean differs from the actual by more than for the case with no noise (Table 2) and the standard deviations are higher by at least an order of magnitude. It is interesting to note, however, that the means of the cases with calibration factors of 2 and 3 still converge quite well to the actual. The cases with actual factors of 0. are quite far from the actual, with the cases for 1. being in between the other two. Recall that 0. is the lowest apportionment factor that can be found (a hard limit to the range in our algorithm), thus forcing all values to be above that. When there is more freedom in the higher numbers, the standard deviation is greater but the mean is closer to the actual. Figure 8 gives a graphical depiction of the scatter about the mean values for each source for the SNR of 1. Although the scatter about each source is unique, there is no systematic difference between sources.

Table 3. Mean and standard deviations for each group of sources with actual calibration values of 0, 1, 2, and 3 and for a circular configuration and a signal to noise ratio of 1.

Source	0	1	2	3
Mean	0.3470	1.1123	1.9907	2.9944
Std dev	0.0261	0.0332	0.1460	0.0867

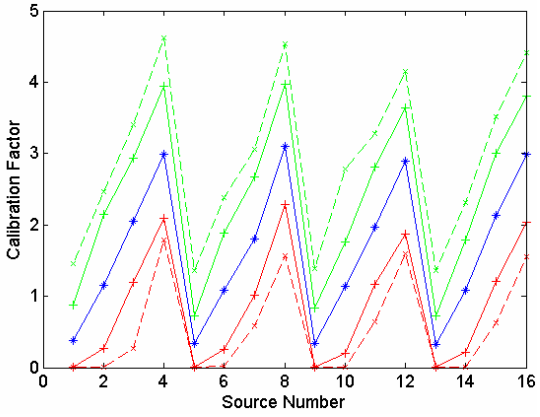

Figure 8. Apportionment of the calibration factors to each of the 16 sources for the case of synthetic circular data and a signal to noise ratio of 1.

For this case with four differing values for the apportionment factor, a single plot showing the scatter in the apportionment as a function

of SNR does not suffice. Instead, Figure 9 (shown at the end of the paper) gives plots for 4 different sources, numbers 1, 6, 11 and 16 representing actual calibration factors of 0, 1, 2, and 3 respectively. As for the prior case, all four source calibration factors are very close to the exact when log(SNR)>0.5 and becomes rather poor as the noise overshadows the signal. The mean value is representative for equal signal and noise and worsens as noise becomes greater.

5.2 Spiral Configuration

As for the pure signal cases, the cases including noise were repeated for a spiral configuration of the sources as depicted in Figure 6 and calibration factors of $S = [0,1,2,3,0,1,2,3,0,1,2,3,0,1,2,3]^T$. Table 4 summarizes the results for a signal to noise ratio of 1. The mean differs from the exact by a larger amount than for the circular case of Table 3. In addition, the standard deviations are larger. Because there is no longer a consistent difference from the source, the scatter about the solution can become larger. This point is further demonstrated in Figure 10. That figure shows the mean, standard deviation and 90% confidence interval for each source and can be compared to Figure 8. The calibration factors for the sources nearest to the receptor are computed rather accurately despite the noise level. However, as the source/receptor distance increases toward source number 16, the mean gets further from the actual value and the scatter about the mean becomes larger. Note that the values close to 0 are much worse than those at 3, which is closer to the midpoint of the range of the search.

Table 4. Mean and standard deviations for each group of sources with actual calibration values of 0, 1, 2, and 3 and for a spiral configuration and a signal to noise ratio of 1.

Source	0	1	2	3
Mean	0.5904	1.2176	2.1683	2.8523
Std dev	0.4437	0.3039	0.1830	0.1359

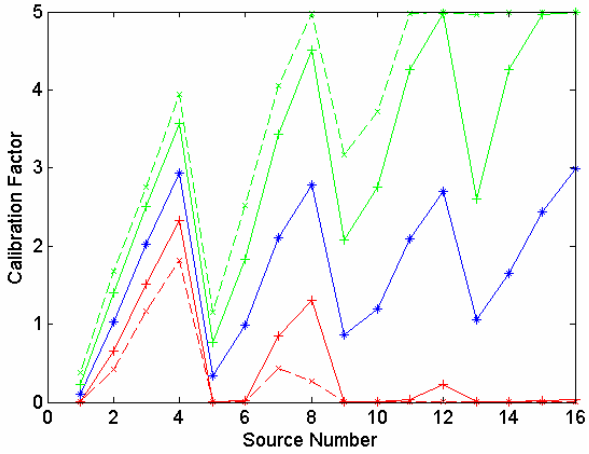

Figure 10. Apportionment of the calibration factors to each of the 16 sources for the case of synthetic spiral data and a signal to noise ratio of 1.

Figure 11 (at the end of the paper) is equivalent to Figure 9, showing the convergence of 100 runs for various different SNR values at each of 4 different sources. In addition to seeing the scatter about the mean become greater for smaller log(SNR) values, we can see the change with distance between the source and receptor. Figure 11a is for source 1, only 250m from the receptor. The agreement of the mean with the actual is quite good and the scatter is very small for noise less than or equal to the signal. As the source is further from the receptor, the scatter about the mean becomes greater for larger values of log(SNR). For source 16, 1750m from the receptor, the scatter is significant when the noise is only one-tenth of the signal. However, the mean of 100 runs is still quite reliable for SNRs of 1.

6. DISCUSSION

This study has demonstrated the ability of a coupled dispersion/receptor model to correctly optimize the source calibration factors for synthetic data. Although the synthetic receptor data was created using the same dispersion model used for the calibration, the coupled model performed well even in the presence of a moderate level of noise. It was demonstrated that even when the noise or geometry make the apportionment problem rather difficult, the mean of a larger number of optimization runs is still quite reliable until the noise becomes overwhelming. This conclusion

provides some hope that this technique could prove useful for apportioning pollutant to potential sources when the expected error is small to moderate.

These results assume a very simple dispersion model. This technique could be much more useful when more complex dispersion models are coupled to a receptor model. We plan to explore this option in future work.

Other useful validation exercises could involve using careful field experiments where the concurrent emission rates of each surrounding source and the background are carefully controlled. If meteorological data is also known to a high level of accuracy, a receptor model coupled with a more accurate dispersion model could give a better understanding of our ability to model atmospheric dispersion.

Note that one can also interpret the source apportionment factors as the total model error. If one had perfect knowledge of dispersion characteristics, meteorological conditions, source emissions information, and receptor data, one would expect the apportionment factors to be all ones. So the difference from one indicates the total uncertainty in the modeling and monitoring process. This implies that such a coupled model could also be used to estimate uncertainty for models.

ACKNOWLEGEMENTS – Part of this work was supported by in house funds from the director of the PSU Applied Research Laboratory. We thank Randy Haupt who is coauthor of the genetic algorithm used for this study.

REFERENCES

Beychok, M.R., 1994, *Fundamentals of Stack Gas Dispersion,* 3rd Ed, Milton Beychok, pub., Irvine, CA, 193 pp.

Cartwright, H.M, and S.P. Harris, 1993, "Analysis of the distribution of airborne pollution using GAs," *Atmos. Environ.*, **27A**, No. 12, pp. 1783-1791.

EPA, 2003a, Revision to the Guidelines on Air Quality Models: Adoption of a Preferred Long Range Transport Model and Other Revisions, Federal Register, 68, (72), 40 CFR Part 51, April 15, 2003.

Haupt, S.E., 2004: Coupled Receptor/Dispersion Modeling with a Genetic Algorithm, AMS 13th Conference on Applications of Air Pollution Meteorology, Vancouver, BC, Canada, paper 6.4.

Haupt, S.E., 2004: Coupling Receptor and Dispersion Models with AI, 8th Annual George Mason University Conference on Transport and Dispersion Modeling, Fairfax, VA, July 15.

Haupt, R.L. and S.E. Haupt, 2000: Optimum Population Size and Mutation Rate for a Simple Real Genetic Algorithm that Optimizes Array Factors, *Applied Computational Electromagnetics Society Journal***, 15**, No. 2, pp. 94-102.

Haupt, R.L. and S.E. Haupt, 2004, *Practical Genetic Algorithms, 2nd Edition with CD*, John Wiley & Sons, New York, NY, 255 pp.

Haupt, S.E., 2005, Genetic Algorithms and their Applications in Environmental Sciences, in *Progress of Artificial Intelligence in Sustainability Science*, J. Kropp, ed., Nova Science Publish, Inc. N.Y., in press.

Loughlin, D.H., S.R. Ranjithan, J.W. Baugh, Jr., and E.D. Brill, Jr., 2000, "Application of GAs for the Design of Ozone Control Strategies," *J. Air & Waste Manage. Assoc.*, **50**, 1050-1063.

Miller, S.L., M.J. Anderson, E.P. Daly, J.B. Milford, 2002, "Source apportionment of exposures to volatile organic compounds. I. Evaluation of receptor models using simulated exposure data," *Atmos. Env.,* **36**, 3629-3641.

Qin, R. and K. Oduyemi, 2003, Atmospheric aerosol source identification and estimates of source contributions to air pollution in Dundee, UK, *Atmos. Env.,* **37**, 1799-1809.

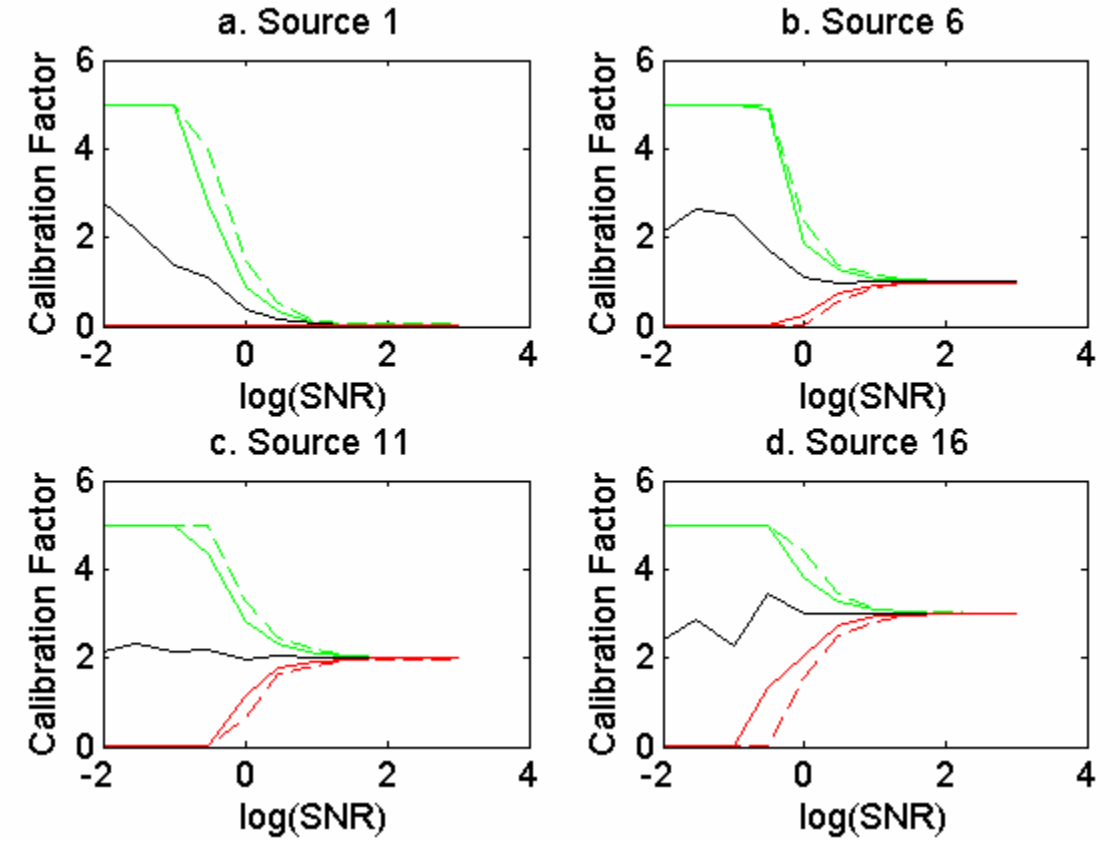

Figure 9. Calibration factor as a function of SNR for a circular source configuration. Mean (black), standard deviation (solid), and 90% confidence interval (dashed) are shown.

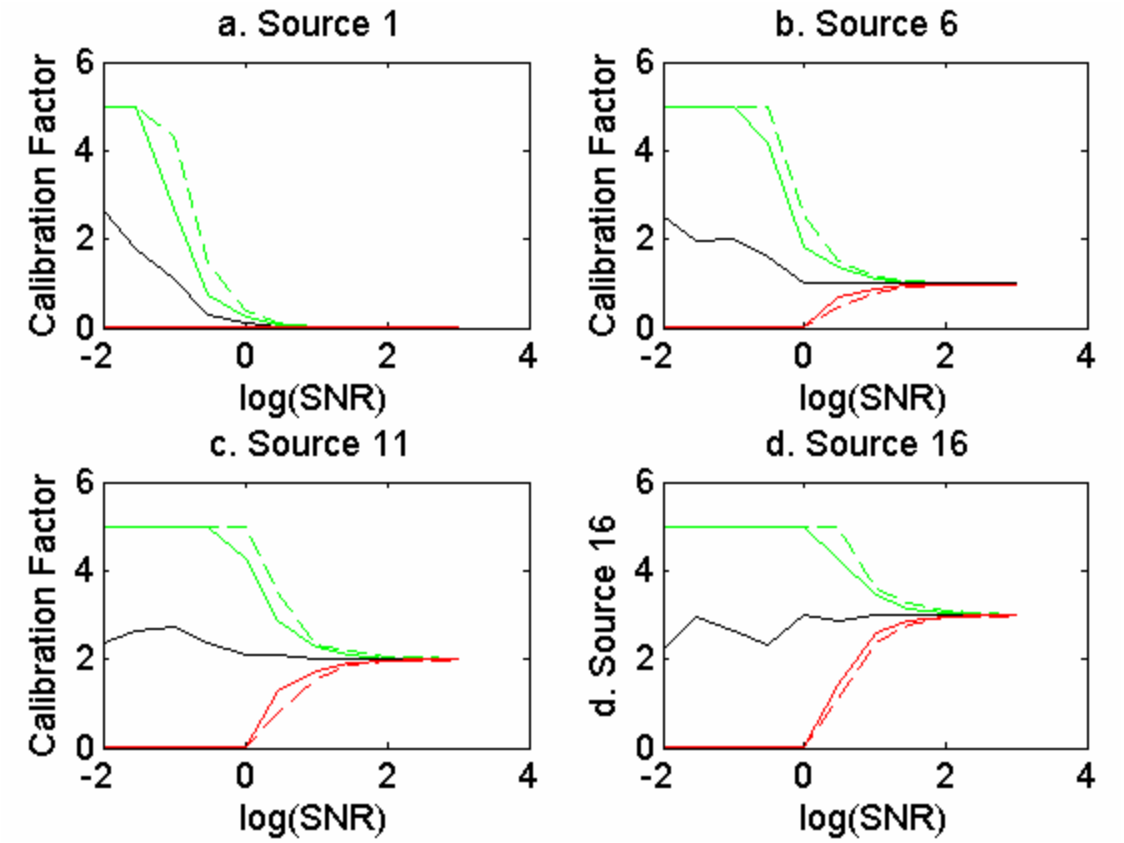

Figure 11. Calibration factor as a function of SNR for a spiral source configuration. Mean (black), standard deviation (solid), and 90% confidence interval (dashed) are shown.