A Genetic Algorithm Method to Assimilate Sensor Data for a Toxic Contaminant Release

Sue Ellen Haupt [1,2] , George S. Young [2] , Christopher T. Allen [3]
[1] The Pennsylvania State University/Applied Research Laboratory, State College, PA, USA
[2] The Pennsylvania State University/Meteorology Department, University Park, PA, USA
[3] Computer Sciences Corporation, Research Triangle Park, NC, USA.
Email: haupts2@asme.org; young@meteo.psu.edu; callen24@yahoo.com

Abstract— **Following a toxic contaminant release, either accidental or intentional, predicting the transport and dispersion of the contaminant becomes a critical problem for Homeland Defense and DoD agencies. To produce accurate predictions requires characterizing both the source of hazardous material and the local meteorological conditions. Decision makers use information on contaminant source location and transport prediction to decide on the best methods to mitigate and prevent effects. The problem has both observational and computational aspects. Field monitors are likely to be used to detect the release and measure concentrations of the toxic material. Algorithms are then required to invert the problem in order to infer the characteristics of the source and the local meteorology. Here, a genetic algorithm is coupled with transport and dispersion models to assimilate sensor data in order to characterize emission sources and the wind vector. The parameters computed include two dimensional source location, amount of the release, and wind direction. This methodology is demonstrated for a basic Gaussian plume dispersion model and verified in the context of an identical twin numerical experiment, in which synthetic dispersion data is created with the same dispersion model. Error bounds are set using Monte Carlo techniques and robustness assessed by adding white noise. Algorithm speed is tuned by optimizing the parameters of the genetic algorithm. Specific GA configurations and cost function formulations are discussed.**

Index Terms—**source inversion, genetic algorithm, data assimilation, sensor data fusion**

I. INTRODUCTION

It is often important to characterize sources of an atmospheric contaminant. A modern application in homeland security is locating and estimating the emission conditions of a hazardous release. Such a release could range from an accidental spill of a toxic contaminant through an intentional release by terrorists. It is expected that Homeland Defense and DoD agencies will have monitors in the field to detect such toxic emissions. These data could prove critical for 1) determining the extent of the emergency, 2) characterizing the source of the emissions, and 3) initializing subsequent predictive modeling. This work examines the use of these data to determine the source and wind information required by a transport and dispersion model for predicting the transport and dispersion of the contaminant. This inversion method provides a means of assimilating observations into the modeling problem.

Some difficulties in the source characterization problem that must be addressed include errors in the monitored data, inadequate area coverage of the monitors, poor first guesses of the location and strength of the contaminant emission, meteorological data that are inadequate to characterize the atmospheric conditions, imperfect models of atmospheric transport and dispersion, and most importantly, the inherently chaotic nature of atmospheric turbulence. This last difficulty means that although we can statistically characterize pollutant concentrations, we cannot definitively predict an exact concentration at an instant in time, but instead compute ensemble average concentrations. Measurements, in contrast, represent a specific realization. There is not currently a good evaluation method for comparing the single realization of a field experiment with the ensemble average statistics from model output [1]. Thus, the physical specification of the source characterization problem is difficult. Our current approach attacks this problem using methods from computational intelligence, in this case, the genetic algorithm.

A. Model Concept

Air pollution models can be divided into two primary categories: receptor and dispersion models. Receptor models are formulated to begin with contaminant concentration data from one or more receptors and project that information backward to characterize the source. In contrast, forward transport and dispersion models start with the source characteristics and meteorological conditions, then use physical, mathematical, and chemical calculations to predict contaminant concentration at some distance from the source. Important input for these dispersion models

Based on "A Genetic Algorithm Method to Assimilate Sensor Data for Homeland Defense Applications," by S.E. Haupt, C.T. Allen, and G.S. Young which appeared the Proceedings of SMCals06: 2006 IEEE Mountain Workshop on Adaptive and Learning Systems, Logan, UT, July 24-26., 2006. © 2006 IEEE.

includes information about the emissions from the source, the local atmospheric conditions, and the geographical characterization. Both types of models are highly developed and forms of them are widely used for diagnosis and prediction of atmospheric contaminant transport events [2].

We formulate a coupled model that uses principals of receptor modeling to compare the monitored data with the predictions of the forward-looking dispersion model. The amount of the observed concentration attributable to each source is controlled by a source-specific tuning parameter, the value of which is determined by the data inversion method. This method has been described in the literature [3-5]. This current work goes a step further, using the GA to tune the wind direction as well as to directly evolve the location and time of the release. Thus, the current work reports on using the GA coupled model methodology to compute the release location (two dimensions), source strength, and the wind direction.

B. Prior work

In our prior work, we demonstrated that coupling receptor models with dispersion models using a GA is an effective tool for attributing concentration contribution at a receptor to each of a specified number of sources [3]. This methodology was tested using a basic Gaussian plume dispersion model on synthetic data for circular source configurations plus an actual source configuration for Logan, Utah. The methodology was then validated by using Monte Carlo techniques to determine the confidence intervals [4]. We also studied the robustness of the methodology by considering both additive and multiplicative white noise [4]. We found that even when the noise was the same magnitude as the signal, the GA coupled model could correctly apportion the pollutant to the correct source. The next step was to replace the Gaussian plume dispersion model with an operational puff model, SCIPUFF [5]. The GA coupled model performed as well with SCIPUFF computing the dispersion as with the Gaussian plume model. That enhanced coupled model was then tested on field test data [5]. Within the limitations of the data, the coupled model still performed admirably. The cases where performance was disappointing proved to be difficult situations during the field test that would be expected to impact data quality. For those cases, prior comparisons of model results to the measured concentrations were also quite poor [6]. The initial reformulation of the problem for tuning the wind appears in [7]. This current paper describes how to best apply this model.

II. MODEL FRAMEWORK

A. Model Formulation

Combining the technology of the forward-looking dispersion models and backward-looking receptor models enables using monitored concentrations to characterize sources, to estimate uncertainty, and to characterize the mean wind conditions during the time of transport. This coupled model integrates the physical basis of the dispersion calculations with the ground truth of the actual monitored pollutant concentrations. We choose to formulate the coupling problem as one in optimization and solve it using a genetic algorithm (GA). In particular, we wish to minimize the cost function formulated as

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

where C_r is the emissions predicted by the forward dispersion model, R_r is the monitored data value at receptor r, TR is the total number of receptors, and a is a constant:

$$a = \max\left(\frac{1}{\sum_{r=1}^{TR} R_r}, 1\right) \tag{2}$$

Our prior work showed that taking a difference of logarithms works better than a linear difference [5]. This issue is revisited in section V.

The concentrations are computed with a Gaussian plume model:

$$C_r = \frac{Q}{u\sigma_z\sigma_y 2\pi}\exp\left(\frac{-y_r^2}{2\sigma_y^2}\right)\left[\exp\left(\frac{-\left(z_r-H_e\right)^2}{2\sigma_z^2}\right)+\exp\left(\frac{-\left(z_r+H_e\right)^2}{2\sigma_z^2}\right)\right] \tag{3}$$

where: C_r = concentration of emission from source at receptor r

(x_r, y_r, z_r) = Cartesian coordinates of the receptor in the downwind direction of the source

Q = emission rate from the source

u = wind speed

H_e = effective height of the plume centerline above ground

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

This is the same Gaussian plume equation used in the original coupled model from our early experiments [3,4]. The dispersion coefficients are calculated following [8].

$$\sigma = \exp\left[I + J\left(\ln\left(x_r\right) + K\left(\ln\left(x_r\right)\right)^2\right)\right] \tag{4}$$

where x is the downwind distance (in km) and I, J, and K are empirical coefficients dependent on the atmospheric stability [8].

Concentration forecasts are created for each trial solution with (3). These are then compared with receptor data for an arbitrary number of sites. The GA optimizes the combination of source location, strength, and surface wind direction that provides the best match between the monitored receptor data and the expected concentrations as compared by (1).

B. The Continuous Genetic Algorithm

For this problem we chose a continuous parameter GA, that is, one in which the parameters are real numbers. Fig. 1 flowcharts the GA solution process. The genetic algorithm starts with a population of random vectors (i.e. chromosomes) that are evaluated using the forward model and cost function (3). The GA then mates the best chromosomes, producing two new chromosomes from two existing chromosomes. Haupt and Haupt (2004) describe several mating schemes. The mating scheme used in [4] and [5] is single-point crossover, which chooses a random crossover point, swaps all parameters after the crossover point, and blends at the crossover point, thus producing two new chromosomes. Here we instead use a uniform crossover scheme that blends all parameters, not just a single parameter at the crossover point. This blending scheme has the advantage of simultaneously changing all parameters, which can improve performance when the response of the cost function to some of the parameters is correlated. In this case, the response to source location and wind vector are highly correlated. The number of new chromosomes produced by mating is determined by the selection rate, which is the fraction of the population retained in each generation.

The chromosome population is further modified through mutations. Mutations replace individual values with new random values. The mutation process enables the algorithm to continue to search the entire solution space rather than converge to a local minimum. The number of mutations in each generation is controlled by the mutation rate.

Each round of mating and mutating constitutes one GA generation. We run the GA for a pre-determined number of iterations, or until convergence has occurred. More details of the technique are found in [9]. We employ elitism, which prevents the best solution computed in each generation from being changed until it is supplanted. We discuss sensitivity to selection of the mutation rate and population size below.

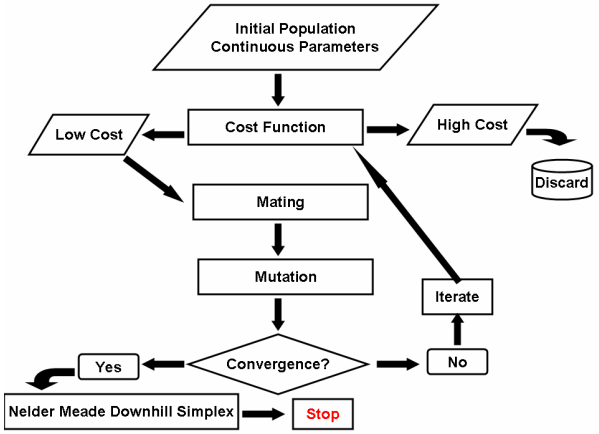

Figure 1. Flowchart of the continuous GA.

Here, we use a hybrid GA, which uses the GA to find the correct solution basin, then applies the Nelder-Meade Downhill Simplex (NMDS) method to complete finding the minimum point of that basin. The rationale for this combination is that the GA is sufficiently robust to usually find the basin of the global minima. Once that basin is identified, however, the NMDS finds the bottom of that basin more rapidly. As demonstrated in section IV, the NMDS method alone is not reliable for finding the global minimum.

III. Application

The coupled receptor/dispersion model technique was demonstrated using both synthetic and real data [3], validated using carefully constructed synthetic data [4], demonstrated to work well with a highly refined dispersion model using field test data [5], and reformulated to directly solve for meteorological variables in addition to the source parameters [7]. Here we demonstrate that we can use the genetic algorithm to additionally back-calculate the wind direction for the transport and dispersion in the context of a basic Gaussian plume dispersion model. The method is validated in identical twin experiments described below. Because the GA is a stochastic method initialized with random values for the parameters being sought, a slightly different solution and varying convergence properties are expected for each run. Therefore, all results reported here are for the average of several model runs to remove the stochasticity from our analysis.

A. Test Configuration

The first step is to demonstrate and validate the method of tuning meteorological data and source characteristics. We do this in the context of an identical twin experiment; that is, we generate synthetic data produced by (3) to compute contaminant concentration at the receptors. Using the same dispersion model to generate the synthetic data as is used in the coupled GA system to back-calculate the source parameters enables us to eliminate part of the potential source of variability for the purpose of validating the technique. The receptors are sited on a grid surrounding the source, each separated by 2000 meters. Model runs are performed using 2×2, 4×4, 8×8, 16×16, and 32×32 grids of receptors. For all five receptor configurations, the source is located in the center of the receptor domain at the point defined as the origin (0,0). Synthetic data is produced for each receptor configuration for two different wind directions, 180° and 225°. A wind direction of 180° places the plume centerline directly between receptors, and a wind direction of 225° places the plume centerline directly over some of the receptors. Fig. 2 shows the 8×8 receptor grid for the 225°wind direction.

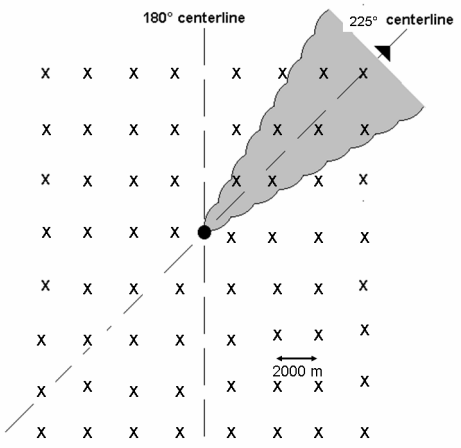

Figure 2. Schematic of receptor grid, pictured with plume from 225°.

B. Results

Table 1 gives results computed by the GA alone: wind directions, source strengths, source locations, and skill scores for six synthetic configurations. We used a population size of 1,200, mutation rate of 0.01, and 100 iterations for each GA run. The correct solution is θ = (180 or 225°) for our two cases, strength = 1.00, and (x, y) = 0, 0. Skill scores are designed to equally weight the error in wind direction, source strength, and source location. 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 denoting an exact solution.

The formulas for the three skill score components are:

$$S_{wind} = \ln\left(\left|\theta_{GA} - \theta_{act}\right| + 1\right) / 5.199 \tag{5}$$

$$S_{str} = \max\left(\left(\frac{S_{GA}}{4 * S_{act}} - \frac{1}{4}\right), \left(\frac{S_{act}}{4 * S_{GA}} - \frac{1}{4}\right)\right) \tag{6}$$

$$S_{loc} = 1.0746 * \left(-\exp\left(-dist/1500\right) + 1\right) \tag{7}$$

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 scale each score from 0 to 1. 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.

The GA system was able to find the exact solution given a grid of at least 8×8 receptors. For fewer receptors, such as a 4×4 grid, performance was less satisfactory, as a smaller set of receptors does not provide enough information to distinguish the effect of wind direction from that of the source characteristics.

TABLE 1. 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 ONE GA RUN. THE CORRECT SOLUTION IS θ= (180 OR 225), STRENGTH = 1.00, AND (X, Y)=(0.,0.)

Configuration	Strength	(x, y) m	θ	Skill score
8×8, θ=0°	2.96	-417,1346	184.12°	1.4581
8×8, θ=225°	1.06	-26, -56	223.95°	0.1952
16×16, θ=0°	1.00	-1, 0	180.01°	0.0029
16×16, θ=225°	1.00	-1, 1	225.01°	0.0019
32×32, θ=0°	1.00	0, 0	180.00°	0.0000
32×32, θ=225°	1.12	-123,519	220.27°	0.6870

C. Noisy Observations

The results shown to this point have assumed that the sensors provide perfectly accurate data, the remaining atmospheric variables (except for the wind direction) are exactly known, there is no unrepresented turbulence, and the Gaussian plume model is an accurate representation of the effects of atmospheric dispersion. Of course these assumptions are unreasonable. In reality, we would expect high variability in atmospheric state, the specific realization does not match the ensemble average Gaussian plume model, and the sensors have thresholds and are prone to errors. In addition, the success of the GA in matching the synthetic data runs is partially due to the nature of the identical twin experiment: the synthetic receptor data is computed with the same dispersion model as the expected concentrations. With real data, the dispersion model would not provide a perfect match to the receptor data, as there are discrepancies between ensemble-means being predicted and realization values that are measured. The match is compromised further by monitoring errors. Therefore, the next step in validating the GA model is to contaminate our synthetic data with white noise to simulate the variability and errors present in monitored receptor data.

Twelve model runs are performed spanning two wind directions and six different ratios of signal-to-noise (SNRs): infinity (no noise), 100., 10., 1., 0.1, and 0.01. An SNR above 1 indicates less noise than signal, while an SNR below 1 indicates more noise than signal. Analyses are made for wind directions of both 180° and 225°. Each of these runs is performed repeatedly, once using each the five receptor grid configurations from section IIIA, so as to resolve the role of data quantity in determining the sensitivity of the results to various quantities of noise. It is expected that runs with more receptors are less sensitive to noise than runs with fewer receptors.

Fig. 3 indicates median skill scores across twelve runs for each combination of SNR and $n \times n$ receptor grid. Recall that lower skill scores denote better solutions. Fig. 3 shows the results for additive noise; the results for multiplicative noise are quite similar. The median skill score is considered rather than the mean, because the median is less sensitive to outliers and is more indicative of what to expect in a single run. Looking along the

horizontal lines, this figure shows that the ability of the model to compute the correct solution is not appreciably affected as long as the signal is greater than the noise (SNR > 1) and the receptor grid is sufficiently large. For SNR = 1, where the signal and noise are of equal magnitude, the model performs slightly better with additional receptors beyond an 8×8 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 reasonable degree of accuracy. It is likely that with this much noise, the actual plume can no longer be detected from the receptor data. At this noise level, we expect that no optimization method can find the solution. This conclusion is supported by the graph, which shows that additional noise beyond SNR = 0.1 does not affect the GA's ability to compute the known solution, as solutions at SNR = 0.1 are already poor.

Recall that in the synthetic data runs without noise, the NMDS algorithm could be applied to further improve the computed solution after the 100th GA iteration. In the runs with noise, however, application of NMDS after the 100th GA iteration typically did not improve the solution. The average skill score of the GA-produced solutions across all SNRs and receptor grids was 1.578, whereas the average skill score after the application of NMDS was 1.582. This result is not surprising, because after the receptor data is contaminated with noise, the fitness landscape has become more rugged. Therefore, fine-tuning the solution with the NMDS can push the solution into the incorrect sub-basin. While NMDS may find a lower cost function value than the GA, the objective skill score compares the result to the known correct solution in the noise-free solution space. Thus, application of NMDS may not result in a lower skill score for noisy data.

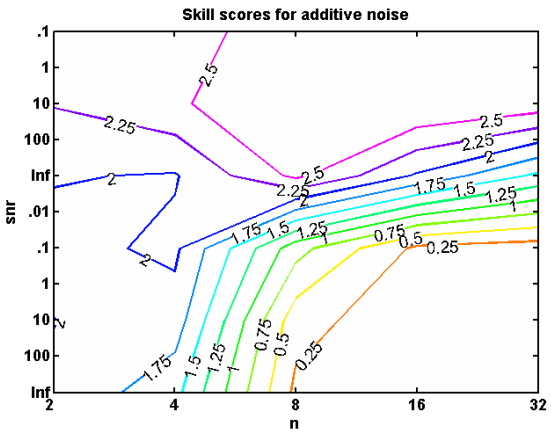

Figure 3. Contour plot of median skill score for various *n*-by-*n* receptor grids and signal-to-noise ratios (SNRs) for additive noise. The median skill scores are taken over 12 runs. Lower skill scores denote better solutions.

IV. Analysis of GA Configuration

The GA model successfully found correct wind directions and source configurations in most cases. The next question is which configuration of GA parameters and receptor grid works best for this problem.

A. Mating

Best GA performance was obtained using a mating scheme where all parameters are blended according to uniform crossover. The superiority of this method to the single-point crossover scheme used in our previous work [3-5] is most likely due to correlations between the effects of the parameters, specifically the response of plume structure to wind direction and source location. The location of the plume centerline is uniquely determined by source location and wind direction. Changes in either of these parameters will modify the location of the plume centerline. Therefore, it is advantageous to modify these parameters simultaneously when.searching for an improved solution. Blending all parameters improves the average skill score across six runs from 0.613 to 0.061, a remarkable improvement from single-point crossover.

B. Population Size

For this problem, the GA requires a larger population size than used in our prior work [3-5] in order to adequately sample the solution space. With a population size of 1,200, the GA can find the solution in a single run in 100 iterations or less. All GA runs in Table 1 produced a solution close to the actual, and some even produced a solution within tolerance, defined as correct within 0.01° in wind direction, 1% of source strength, and 1 meter from the actual source location..

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, which is proportional to the population size times the number of iterations. Smaller population sizes often converge too quickly and thus reach an incorrect solution, even with a high mutation rate. Numerous population sizes and iteration numbers were tested to determine the best compromise between computing time and performance. Table 2 shows how many of six runs returned the solution within the specified tolerance after the application of the hybrid GA/NMDS for 16 combinations of population size and number of iterations. A population size of 1,200 and 100 iterations resulted in the prescribed solution for all six runs made with the least computing time. These are the same runs from Table 1, but Table 1 shows the results before the application of the downhill simplex.

C. Receptor Grid

In Table 1 we saw that the GA's performance is exceptional. Numerous additional runs (not shown) were able to find an accurate solution repeatedly, demonstrating the consistency of the GA. Accurate solutions, however, can only be found when using at least

TABLE 2. 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.

Iteration/ Population Size	50	100	150	200
400	3	4	4	5
800	4	4	4	5
1200	5	6	6	6
1600	5	6	6	6

an 8×8 grid of receptors. For a 2×2 receptor grid, solutions were basically random. For a 4×4 grid, solutions were better, but nowhere near the exactness of the 8×8 grid solutions. This suggests that a 4×4 grid of receptors does not provide enough receptor data to distinguish the effects of wind direction from those of source location and source strength. Only two or three of the receptors in a 4×4 grid provide useful data since the others are outside the plume or nearly so. In that case there are four parameters to be tuned (wind direction, source strength, and two for source location). The poor results should not be surprising as there are more unknowns than inputs. In contrast, for an 8×8 grid, the number of receptors inside the plume exceeds the number of unknowns, so the model is successful.

Fig. 3 gives information on the grid requirements in the presence of noise. It is clear that with a 32×32 grid we can invert the problem as long as the noise does not exceed the signal. For grids on the order of 4×4 to 8×8, the GA model has difficulty identifying the correct solution as more noise is added.

D. 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 (1) would weight the outliers more heavily. Conversely, lower powers consider the outliers as less important. To evaluate how this might impact the results, we look at alternate formulations for the cost function**.**

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. In general, for this problem, the more GA iterations performed, the lower will be the final value of 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:

$$\mathrm{SqRoot} = \frac{\left(\sum_{m=1}^{M}\sqrt{|C\bullet S - R|}\right)^2}{\left(\sum_{m=1}^{M}\sqrt{|R|}\right)^2} \tag{8}$$

$$\mathrm{AbsVal} = \frac{\sum_{m=1}^{M}|C\bullet S - R|}{\sum_{m=1}^{M}|R|} \tag{9}$$

$$\mathrm{FourthRoot} = \frac{\sqrt[4]{\sum_{m=1}^{M}(C\bullet S - R)^4}}{\sqrt[4]{\sum_{m=1}^{M}(R)^4}} \tag{10}$$

$$\mathrm{EighthRoot} = \frac{\sqrt[8]{\sum_{m=1}^{M}(C\bullet S - R)^8}}{\sqrt[8]{\sum_{m=1}^{M}(R)^8}} \tag{11}$$

$$\mathrm{RMSAbs} = \mathrm{RMS} + \mathrm{AbsVal} \tag{12}$$

These cost functions are compared in the context of our original GA coupled model formulation [3-5]. Table 3 summarizes the results 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 and normalized to 0.0 for the spiral case).
3. Min: The minimum calibration factor each run, averaged over the six runs. We again hope to see this as close to the actual 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.

As seen in the Table 3, the metrics for the different cost functions vary little, although the higher power cost functions perform somewhat worse than the SqRoot, AbsVal, RMS, and RMSAbs. For a circular configuration, the AbsVal function works best, closely followed by the SqRoot. For a spiral 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 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 is reasonable and easy to compare with other methods that are based on RMS differences.

TABLE 3. EVALUATION OF DIFFERENT COST FUNCTION FORMULATIONS FOR A CIRCULAR GEOMETRY.

Metric/ Cost Func.	RMS	Max	Min	In 0.01
RMS	0.050919	1.02305	0.97063	10.5
SqRoot	0.048137	1.02045	0.97270	11.2
AbsVal	0.044658	1.02457	0.97757	11.3
FourthRoot	0.056269	1.02503	0.97215	8.0
EighthRoot	0.063798	1.03520	0.97195	9.8
RMSAbs	0.049764	1.02443	0.97546	11.2

E. *GA vs. Random search*

Does solving the inversion problem require the GA? To answer this question, the GA's performance is compared to the performance of a random search method. Fig. 4 shows the minimum cost found by the GA (dashed) and a random search (solid), averaged over 5 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 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 (within the tolerance) in about 7000 iterations. Thus, we conclude that a random search is inefficient and that more sophisticated optimization methods such as a GA are required for this problem.

F. Refinement via a Hybrid GA

Because the solution after the 100th iteration is often close to the optimal solution (i.e. global minimum of the cost function), we investigate whether a traditional gradient descent method such as NMDS [10] further improves the solution. The method begins with a first guess solution on a multi-dimensional surface and finds a local minimum in the vicinity of the starting point.

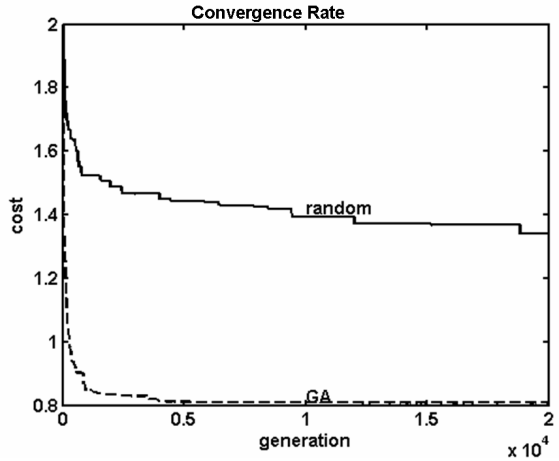

Figure 4. Minimum cost function value as a function of iteration number for the GA (dashed) versus a random search method (solid), carried out to 20,000 iterations.

Traditional gradient descent methods such as NMDS are ineffective if we cannot obtain a good first guess. As discussed earlier, the NMDS method can only find the exact solution if it is close enough initially to be in the same basin as the global minimum. If we use the GA to provide a good first guess, the NMDS method is very efficient at further tuning the solution.

The NMDS method was run on each of the solutions from Table 1. Each time, the NMDS method returned a very accurate solution. Thus, this method can be used effectively to further improve the accuracy of the solution after the termination of the GA. In this mode the GA is used to locate the basin of the global minimum of the cost function and the NMDS method to find the bottom of the basin. Thus, the entire optimization process is a hybrid GA. While the simplex is not designed to find global minima due to its requirement for a sufficiently close first guess, it provides an efficient means of refining trial solutions that are in the same basin as the actual solution.

In light of the discussion above, another issue to consider is whether the NMDS method, by itself, could solve the problem. We recognize that the NMDS method is sensitive to the initial guess; therefore, we choose to initialize it with random initial guesses and average the number of cost function evaluations required to solve the problem directly. Table 4 shows the number of function calls (a uniform unit of computing time) required to find the solution within a tolerance of 0.01° for wind direction, 1% of source strength, and 1 meter of source location for the GA and for the random initialization NMDS method. The results are averaged over two runs for each receptor and wind direction configuration for a total of twelve runs. The number of function calls required in any individual run using the simplex method is highly dependent on random initialization. Therefore, it is not surprising that in some instances, NMDS found the solution faster than the GA. The performance of the GA, however, is far more consistent than the NMDS method over the twelve runs performed, because it is able to overcome a bad start to 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 the solution, while NMDS took an average of 78,725 function calls. Thus, running the NMDS method from random starting points until the solution is found is inefficient compared to the GA, and particularly the GA-NMDS hybrid.

TABLE 4. NUMBER OF COST FUNCTION EVALUATIONS REQUIRED TO FIND THE SOLUTION FOR THE GA AND THE NMDS METHOD, AVERAGED OVER TWO RUNS FOR EACH CONFIGURATION.

Configuration	Nelder-Mead function calls	GA function calls
8×8, θ=180°	17180	19200
8×8, θ=225°	123235	1200
16×16, θ=180°	60874	13800
16×16, θ=225°	121035	3600
32×32, θ=180°	16996	10200
32×32, θ=225°	133034	23400

.IV. CONCLUSIONS

This data assimilation and source characterization problem is an example of how computational intelligence can be applied in real-world problems of practical interest that extend to operational applications. Homeland Defense and DoD agencies have a need to assimilate concentration and wind data that is being monitored in the field and use it to characterize the contaminant source. The source information could then be used to initialize predictions of transport and dispersion of the contaminant. In addition, this method provides the data necessary for assimilating wind direction into meteorological forcing models for transport and dispersion. It is related to sensor data fusion in that it uses data obtained in the field to obtain appropriate modeling data.

The GA model system has shown promise in characterizing the wind direction as well as the strength and two-dimensional location. The success of this method has required careful formulation of the cost function and the solution methodology as discussed in section V.

This work is just the beginning of what can be done with AI-type techniques to blend sensor data into real-world computational problems. Our own continuing efforts are now focusing on using these same techniques to additionally back-calculate source height, time of release, and wind speed. As for this work, those calculations will be validated using identical twin experiments then analyzing the robustness of the technique when noise is added. Additionally, we will test the model in more realistic frameworks, such as sensor data simulators and on field test data.

We are also looking at integrating these methods with assimilation and sensor data fusion tools to provide further means of using these inverted data for subsequent transport and dispersion modeling. We also expect to analyze how much data are necessary to perform our inversions, and how this result changes when system noise is considered.

ACKNOWLEDGMENT

This work was supported, in part, by the Defense Threat Reduction Agency under grant number W911NF-06-C-0162, and also by the PSU Applied Research Laboratory.

REFERENCES

[1] National Research Council, *Tracking and Predicting the Atmospheric Dispersion of Hazardous Material Releases. Implications for Homeland Security*, The National Academies Press, Washington, D.C., 2003.

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

[3] S.E. Haupt, "A Demonstration of Coupled Receptor/Dispersion Modeling with a Genetic Algorithm," *Atmospheric Environment*, vol. 39, 2005, pp. 7181-7189.

[4] S. E. Haupt, G. S. Young, and C. T. Allen, "Validation of a Receptor/Dispersion Model Coupled with a Genetic Algorithm Using Synthetic Data*," J. Appl. Meteor and Clim..,* **45**, 476–490.

[5] C. T. Allen, S.E. Haupt, and G. S. Young, "Source Characterization With a Genetic Algorithm-Coupled Receptor/Dispersion Model Incorporating SCIPUFF", 2007, *J. Appl. Meteor and Clim..,* in press.

[6] J.C Chang, P. Franzese, K. Chayantrakom, and S. R. Hanna, "Evaluations of CALPUFF, HPAC, and VLSTRACK with Two Mesoscale Field Datasets", *J. Appl. Meteor.*, **42**, 2003, 453-466.

[7] C.T. Allen, G.S. Young, and S.E. Haupt, "Improving Pollutant Source Characterization by Optimizing Meteorological Data with a Genetic Algorithm *Atmospheric Environment,* 2007, **41**, 2283-2289.

[8] M.R. Beychok, Fundamentals of Stack Gas Dispersion, 3rd Ed. Milton Beychok, pub., Irvine, CA, 1994.

[9] R. L. Haupt and S. E. Haupt, Practical Genetic Algorithms, 2nd edition with CD. John Wiley & Sons, New York, NY, 2004.

[10] J. A. Nelder and R. Mead, "A Simplex Method for Function Minimization", *Computer Journal*, **7**, 1965, 308-313.

Sue Ellen Haupt earned a B.S. in meteorology and marine science certificate from The Pennsylvania State University (University Park, PA) in 1978, M.S. in engineering management from Western New England College (Bedford, MA) in 1982, M.S. in mechanical engineering from Worcester Polytechnic Institute (Worcester, MA) in 1984, and Ph.D. in atmospheric science from the University of Michigan (Ann Arbor, MI) in 1988.

She is a Senior Research Associate in the Computational Mechanics Division of the Applied Research Laboratory and Associate Professor of Meteorology at The Pennsylvania State University (State College/University Park, PA). Her prior affiliations include National Center for Atmospheric Research, University of Colorado/Boulder, US Air Force Academy, Utah State University, University of Nevada/Reno, New England Electric System, and GCA Corporation. She is coauthor of *Practical Genetic Algorithms* (Wiley and Sons, NY, NY, 1998, second edition 2004) and is currently editing *Artificial Intelligence Methods in the Environmental Sciences* (to be published by Springer in 2007). She has authored over 100 book chapters, journal articles, conference papers, technical reports, and workshop proceedings. Her specialty is in applying novel numerical techniques to problems in fluid dynamics.

Dr. Haupt chairs the Committee on Artificial Intelligence Applications to Environmental Science of the American Meteorological Society (AMS), and is on the AMS Scientific and Technical Activities Commission and Program Organizing Committee for the Annual Meetings in 2007 and 2008. She is a member of DTRA's Sensor Data Fusion Working Group and is faculty advisor for the PSU Section of the Society of Women Engineers (SWE). In addition to AMS and SWE (Senior Member) she is a member of the American Society of Mechanical Engineers, Society for Industrial and Applied Mathematics, American Society of Engineering Educators, the American Geophysical Union and three honor societies: Tau Beta Pi (Engineering), Phi Mu Epsilon (Mathematics), and Chi Epsilon Pi (Meteorology).

George S. Young earned a B.S. in meteorology from Florida State University (Tallahassee, FL) in 1979, M.S. in meteorology from Florida State University (Tallahassee, FL) in 1982, and Ph.D. in atmospheric science from Colorado State University (Fort Collins, CO) in 1986.

He is a Professor in the Meteorology Department at The Pennsylvania State University (University Park, PA) where he has been on the faculty since 1986. He has authored 170 book chapters, journal articles, conference papers, technical reports, and workshop proceedings. His specialty is application of artificial intelligence methods to weather forecasting and satellite image analysis.

Dr. Young is a member of the Information Technology Committee of the National Weather Association (NWA). In addition to the NWA he is a member of the American Meteorological Society and four honor societies: Phi Beta Kappa (National Honor Society), Sigma Xi (Scientific Research Society), Phi Mu Epsilon (Mathematics), and Chi Epsilon Pi (Meteorology).

Christopher Allen earned dual B.S. degrees in meteorology and computer science from the Florida State University (Tallahassee, FL) in 2004, and an M.S. in meteorology from The Pennsylvania State University (University Park, PA) in 2006.

He is currently a systems analyst with Computer Sciences Corporation, Research Triangle Park, NC. He has previously worked as a research assistant for the meteorology departments of both the Florida State University and The Pennsylvania State University. He has authored three prior journal articles: two in the *Journal of Applied Meteorology and Climatology* and one in *Atmospheric Environment*. His research interests include numerical modeling (specifically air pollution modeling) and numerical weather prediction.