J1.5

# NON DARWINIAN EVOLUTION FOR SOURCE ESTIMATION

G. Cervone*[1] and P. Franzese[2]
[1]Dept. of Geography and Geoinformation Science, George Mason University
[2]Center for Earth Observing and Space Research, George Mason University

## 1. Introduction

The problem of designing efficient methodologies to locate and characterize the sources of atmospheric releases is attracting much interest. A promising approach consists in coupling artificial intelligence algorithms to dispersion models. Typically, the dispersion model performs a number of simulations from a series of tentative sources. The function of the artificial intelligence algorithm is to determine the series of tentative sources in order to efficiently converge toward the real source. Concentration measurements are necessary to implement the process. For each tentative source, the algorithm analyzes the error between the simulated and the observed values, and prescribes the next tentative source (or the next set of tentative sources). Knowledge of some meteorological variables is not strictly required in principle, but in practice may be essential especially for large scale (i.e., no fixed winds) dispersion scenarios. The main appeal of this paradigm is that the dispersion model does not need to be modified. Therefore, the method is easy to implement with any well established dispersion model independent of its complexity and internal working.

Several popular artificial intelligence algorithms use Bayesian inference coupled with stochastic sampling (Gelman et al., 2003; Johannesson et al., 2004; Senocak et al., 2008). A different approach based on evolutionary algorithms has also been proposed and tested (Haupt et al., 2007; Allen et al., 2007; Cervone and Franzese, 2010b; Cervone and Franzese, 2010a). Evolutionary computation algorithms (of which genetic algorithms are one of the main paradigms, in addition to evolutionary strategy, evolutionary programming and genetic programming) are iterative stochastic methods that evolve in parallel a set of potential solutions through a trial and error process. Potential solutions are encoded as vectors of values, which can include a number of source characteristics such as, e.g., its geometry and size, location, and emission rate. The potential solutions are evaluated according to an objective function (often referred to as fitness function, error function, or skill score), which is a measure of the difference between the concentration field simulated by the dispersion model from a tentative source, and the available observations. The evolutionary process consists of selecting one or more candidate solutions whose vector values are modified to minimize the objective function. A selection process is invoked that determine which of the new solutions survive into the next generation. While the methodologies and algorithms that are subsumed by this name are numerous, most of them share one fundamental characteristic: they use non-deterministic operators such as mutation and recombination as the main engine of the evolutionary process.

These operators are semi-blind, and the evolution is not guided by knowledge learned in the past generations. In fact, most evolutionary computation algorithms are inspired by the principles of Darwinian evolution, defined by "...one general law, leading to the advancement of all organic beings, namely, multiply, vary, let the strongest live and the weakest die" (Darwin, 1859). The Darwinian evolution model is simple and fast to simulate, and it is domain-independent. Because of these features, evolutionary algorithms have been applied to a wide range of optimization problems (Ashlock, 2006).

In this paper we introduce a multi-strategy iterative approach that pairs the traditional Darwinian operators with a non-Darwinian machine learning evolutionary process. This approach provides a different search strategy where new candidate solutions are generated by an inductive inference reasoning process. The main drawback compared with traditional algorithms is higher algorithm complexity and a possible increase of computational needs.

There have been several attempts to extend the traditional Darwinian operators with statistical and machine learning approaches that use history information from the evolution to guide the search process. The main challenges are to avoid local maxima and increase the rate of convergence. The majority of such methods use some form of memory and/or learning to direct the evolution towards particular directions thought more promising (Grefenstette, 1991; Sebag et al., 1997; Reynolds, 1999; Hamda et al., 2002).

Because evolutionary computation algorithms evolve a number of individuals in parallel, it is possible to learn from the 'experience' of entire populations. A similar type of biological evolution does not exist, because in nature there is no mechanism to evolve entire species.

Estimation-of-Distribution Algorithms (EDA) are a form of evolutionary algorithms where an entire population may be approximated with a probability distribution (Lozano, 2006). New candidate solutions are not chosen at random, but using statistical information from the sampling distribution. The aim is to avoid premature convergence and to provide a more compact representation.

Discriminating between best and worst performing individuals could provide additional information on how to guide the evolutionary process. The Learnable Evolution Model (LEM) includes a machine learning rule induction algorithm to learn attributional rules which discriminate between best and worst performing candidate solutions (Cervone et al., 2000; Cervone et al., 2000). New individuals are then generated according to inductive hypotheses

* *Corresponding author address*: Dept. of Geography and Geoinformation Science, George Mason University, Fairfax, VA 22030; gcervone@gmu.edu

discovered by the machine learning program. The main difference with Darwinian-type evolutionary algorithms is the way new individuals are generated. In contrast to Darwinian operators of mutation and/or recombination, Machine Learning (ML) conducts a reasoning process in the creation of new individuals. Specifically, at each step (or selected steps) of evolution, a machine learning method generates hypotheses characterizing differences between high-performing and low-performing individuals. These hypotheses are then instantiated in various ways to generate new individuals The hypotheses indicate the areas in the search space that are likely to contain high-performing individuals. New individuals are selected from these areas and then classified as belonging to either a high-performance or a low-performance group, depending on their fitness value. These groups are then differentiated by the machine learning program, yielding a new hypothesis as to the likely location of the global solution.

The possible advantages in generating new individuals of ML compared to traditional Darwinian operations mainly depend on the evolution length, defined as the number of function evaluations needed to determine the target solution, and the evolution time, defined as the execution time required to achieve this solution. A choice between ML and Darwinian algorithms is based on the tradeoff between the complexity of the population generating operators and the evolution length. In our case, the proposed operations of hypothesis generation and instantiation are more computationally costly than mutation and/or crossover, but the evolution length is typically much shorter than in Darwinian evolutionary algorithms.

In all applications where the evaluation of a new individual is computationally taxing, speeding up the convergence rate is paramount. Therefore ML as engine of evolution is only convenient for problems with high objective function evaluation complexity. *Non-Darwinian evolution is not a replacement for traditional evolutionary algorithms*. It is a new paradigm to speed-up a certain class of problems that contain particularly complex evolution functions.

The problem of source detection of atmospheric pollutants is an ideal problem due to the complexity of the function evaluation which may require expensive numerical simulations.

This paper describes an application of non-Darwinian evolution for the source characterization of atmospheric releases.

The AQ4SD program (Cervone et al., 2010) was used as main engine of evolution to generate the hypotheses. The proposed system differs significantly from earlier LEM applications in three ways: a) The reasoning process generates hypotheses (rules) that discriminate between high and low performing individuals; b) The native real value encoding of variables; and c) The selection mechanism that determines the makeup of the next generation.

The new system was tested in two cases: simulated data generated by synthetic releases, and the real concentration measurements from the Prairie Grass controlled field experiment. Initial experiments show that guiding evolutionary processes by hypotheses generation and instantiation can dramatically speed up convergence in the source detection problem. (Barad, 1958).

## 2. Methodology

The proposed methodology is based on an iterative process guided by hypotheses generation and instantiation. Similarly to an evolutionary computation methodology, it evolves solutions in parallel, and evaluates them according to an objective function. However, the traditional Darwinian birth operators are paired with a machine learning process that learns hypotheses, in the form of rules, to describe the characteristics of the highest performing candidate solutions. Specifically, at each step of evolution, a population is divided into High-performing (H-group) and Low-performing individuals (L-group), according to the fitness score. These groups are selected from the current population, or a combination of the current and past populations. Then a learning program creates general hypotheses distinguishing between these two groups, which are instantiated in various ways to produce new candidate individuals. New births occur in the areas of the search space identified as regions most favorable for good solutions.

### 2.1 *Hypotheses Generation*

The central core of the system is the machine learning rule induction algorithm to generate the hypotheses. In this study, the AQ4SD program was specifically designed and optimized to be run iteratively and to drive a Non-Darwinian evolutionary process (Cervone et al., 2010). AQ is a machine learning classifier methodology which generalizes sets of examples with respect to one or more sets of counter-examples (Michalski, 1969; Michalski, 1983; Mitchell, 1997; Cervone et al., 2001; Cervone et al., 2010).

The input data for AQ is therefore made of labeled data, or in other words data which is already assigned to a particular class or group.

AQ is a form of supervised learning, wherein classified data are generalized to identify the characteristics of the entire class. More details on the AQ classifier are given in (Cervone et al., 2010)

In general, a multivariate description is a classified event of type $x_1, ..., x_k$, and $c$, where each $x$ is an attribute value and $c$ is the class it belongs into. For each class $c$, AQ considers as positive all the events that belong to class $c$, and as negative all the events belonging to the other classes. In its simplest form, given two sets of multivariate descriptions $P_1, ..., P_n$ (positive events) and $N_1, ..., N_m$ (negative events), AQ finds rules that cover all $P$ examples, and do not cover any of the $N$ examples.

(negatives), patterns of attribute-values (or rules) that discriminate the characteristics of the positive events with respect to the negative events. Such patterns are generalizations of the individual positive events, and depending on AQ's mode of operation may vary from being totally complete (covering all positives) and consistent (not covering any of the negatives), to accepting a tradeoff of coverage to gain simplicity of patterns.

Table 1: Sample candidate solutions generated

| X | Y | Wind Direction | Height | Q | Stability Class |
|---|---|---|---|---|---|
| 101.26 | 88.12 | 128 | 11 | 33 | D |
| 101.29 | 70.99 | 128 | 15 | 77 | A |
| 101.31 | 121.22 | 128 | 13 | 13 | C |

AQ uses a highly descriptive representation language to represent the learned knowledge. In particular it uses rules to describe patterns in the data. A prototypical AQ *rule* is defined as the following logical equation:

$$\texttt{Consequent} \longleftarrow \texttt{Premise} \sqsubset \texttt{Exception} \quad (1)$$

where *Consequent*, *Premise* and *Exception* are conjunctions of *Conditions*. A *Condition* is simply a relation between an attribute and a set of values it can take:

$$\texttt{[Attribute . Relation . Value(s)]} \quad (2)$$

Whereas Premise and Condition are mandatory, the Exception is optional and used only in very special circumstances. Although Exception has been implemented in AQ4SD, it is not being used, because it often leads to over fitting in the presence of very noisy data.

### 2.2 *Hypotheses Instantiation*

Once rules are generated, they can be easily instantiated to create new candidate solutions by generating attribute-value pairs. This is arguably one of the main advantages in using attributional rulesets to drive the evolutionary process.

The learned hypotheses (attributional rules) are used to generate new individuals by randomizing variables within the ranges of values defined by the rule conditions. If a rule does not refer to some variable, it means that this variable was not needed for distinguishing between the H-group and the L-group. A problem then arises as to what values should be assigned to such variables when generating new individuals. There can be different methods for handling this problem. Early experimental results showed that values can be assigned randomly within the range of the individuals in the current population. This is a conservative method that does not introduce values not already present in the population. We will investigate alternative approaches, as for example using previous rules, or selecting only high performing individuals.

Assuming an optimization problem with six variables *Longitude*, *latitude*, *Wind Direction*, *Height*, *Q*, *Stability Class*, new sample candidate solutions generating through the instantiation of rule are shown in Table 1. The values for the first three variables (X, Y, Wind Direction) are assigned by taking a random value in the range of the possible values cited in the rules. The other variables, are set according to the values of randomly chosen candidate solutions in the current population.

### 2.3 *Evolutionary Algorithm*

The evolutionary process can use hypothesis generation and instantiation as the sole engine or evolution, or pair it with traditional Darwinian operators of mutation and recombination. The two strategies are paired because hypothesis creation and instantiation is more powerful than mutations and/or recombination, but computationally much more costly. By allowing the interchangeable execution of both Darwinian and non-Darwinian operators, our method can utilize the best features of both paradigms. figure 1 presents a general flow diagram of the methodology

The *Initialize population* module creates individuals randomly or according to a given distribution. Select Mode determines if new individuals are generated according to non-Darwinian (default) or Darwinian operators, and alternates depending on the convergence rate. The toggling between two modes continues until the termination condition is satisfied.

In *Machine learning mode*, two methods of selecting H- and L- groups are supported: Fitness-based and Population-Based. In the Fitness-based method, the H-group and L-group consist of individuals whose fitness is above the High Fitness Threshold (HFT) and below the Low Fitness threshold (LFT), respectively. In the Population-based method, the H-group and L-group consist of portions of the population defined by the High Population Threshold (HPT) and the Low Population Threshold (LPT), respectively. HPT defines the percentage of the highest performing individuals, and LPT the percentage of the lowest performing individuals to be selected for the H- and L-group, respectively. HFT, LFT, HPT and LPT are controllable parameters.

The *generate new individuals via hypothesis generation and instantiation* module employs the AQ4SD learning system (Cervone et al., 2010) for generating hypotheses distinguishing between H- and L-groups.

In *Darwinian mode* new individuals are generated by selecting representative individuals, and mutating and/or recombining them. The *Select parents* operator selects representative individuals (parents) from the current population according to some selection method, such as fitness proportional selection, uniform selection, tournament selection, etc. In the current work we have used a method based on fitness proportional selection, where candidate solutions with a smaller error are more likely to be selected, but at the same time, leaving a chance to inferior solution to create offspring. The *Generate new individuals* module creates new individuals by mutation and/or recombination (Bäck, 1996). The mutation operator takes one individual (the parent) and generates one offspring (the child) by varying one or more of its variables. The recombination operator, also known as crossover, takes two individuals (the parents) as input, and it generates one offspring (the child). The variables of the offspring are a mix of the variables of the two parents. In the current work we have not used the recombination operator, but only an adaptive mutation operator as described in (Cervone et al., 2010). The next operations are common for both modes of operation.

The *Evaluate new individuals* module determines the fitness of each individual. In the proposed research, evaluating an individual involves running a numerical transport and dispersion simulation, and comparing the simulated and the observed concentrations. This is a very computationally expensive operation, and one of the driving forces behind the proposed research.

The *Population survival selection* combines previous individuals with the newly generated ones. The proposed

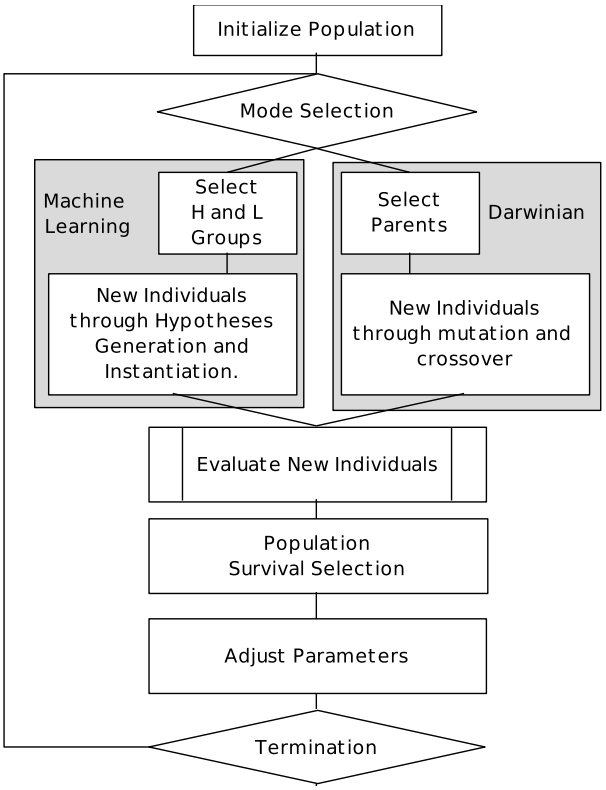


FIG. 1: Flowchart of the non-Darwinian evolutionary algorithm.

methodology uses binary tournament as to select new individuals. Specifically, old and new candidate solutions are merged them together in a single population. Iteratively, two solutions are selected at random and compared. The one with the higher error is removed from the population, while the one with smaller error is kept for subsequent selections. The process continues until the population is resized to the proper value.

The *Adjust parameters* module sets the settings for both non-Darwinian and Darwinian mode based on statistics of previous generations. Adjustments involve varying the generality of the rules, control parameters of the learning algorithm, or adjusting the mutation rate.

The *Terminating condition* determines when to stop the execution of the program. The condition can be defined by the maximal number of function evaluations, by the number of generations without improvement, or by a limit on the execution time

### 2.4 *Transport and Dispersion Simulations*

The dispersion model which will be used to perform the forward simulations is a simple analytical Gaussian model, which is adequate to give satisfactory results for simple and short range dispersion configurations such as Prairie Grass. More complex scenarios may require more sophisticated numerical models. We use the following Gaussian model with a single reflection at the ground, which determines the predicted mean concentration $C_p$ at a location $x$, $y$ and $z$ generated by a source located at $x_s$, $y_s$, and $z_s$ as:

$$C_p(x, y, z, x_s, y_s, z_s) = \frac{Q g_y g_z}{2\pi U[(\sigma_s^2 + \sigma_y^2)(\sigma_s^2 + \sigma_z^2)]^{1/2}} \quad (3)$$

with

$$g_y = \exp\left[-\frac{(y - y_s)^2}{2(\sigma_s^2 + \sigma_y^2)}\right]; \quad (4)$$

$$g_z = \exp\left[-\frac{(z - z_s)^2}{2(\sigma_s^2 + \sigma_z^2)}\right] + \exp\left[-\frac{(z + z_s)^2}{2(\sigma_s^2 + \sigma_z^2)}\right] \quad (5)$$

where $Q$ is the source mass emission rate, $U$ is the wind speed, $\sigma_y(x, x_s; \psi)$ and $\sigma_z(x, x_s; \psi)$ are the crosswind and vertical dispersion coefficients (i.e. the plume spreads) where $\psi$ describes the atmospheric stability class (i.e., $\psi = A$ to $\psi = F$), and $\sigma_s^2 = \sigma_y^2(x_s, x_s, \psi) = \sigma_z^2(x_s, x_s, \psi)$ is a measure of the area of the source. The result of the simulation is the concentration field generated by the release along an arbitrary wind direction $\theta$. The dispersion coefficients were computed from the tabulated curves of Briggs (Arya, 1999). In this study, $U$ and $\psi$ are assumed to be known, and their values set according to the observations reported in the Prairie Grass dataset. Each candidate solution is thus comprised of the 6 variables $x_s$, $y_s$, $z_s$, $Q$, $\sigma_s$ and $\theta$.

### 2.5 *The Prairie Grass field dispersion experiment*

The Prairie Grass experiment is a well documented campaign of dispersion measurements in controlled conditions (Barad, 1958). The experiment consists in 68 releases of trace gas $SO_2$ from a ground level point source. Mean concentration samplers were deployed along five arcs located at 50 m, 100 m, 200 m, 400 m and 800 m from the source. Extensive meteorological measurements were collected during the experiments. The stability of the atmosphere depends on the day and time of the release, and experiments were conducted under all stability classes.

figure 2 shows reconstructed concentration fields averaged over all releases in each atmospheric class, with the wind direction aligned with the horizontal axis.

In addition of the observed Prairie Grass measurements, we have created a synthetic dataset simulating each of the 68 releases using the Gaussian model Eq. (3). The synthetic dataset is generated using the source, wind characteristics, and atmospheric class of each of the original Prairie Grass experiment, and the simulated concentrations are recorded at the corresponding sensor locations for the original experiments. The synthetic dataset is generated for two reasons. First, it allows the assessment of the accuracy of the model simulations with respect to the observed measurements. This gives an indication of how much of the error in the predicted source location of the real case is attributable to the search algorithm and how much is intrinsic to the dispersion model; second, it provides an ideal dataset against which the search algorithm can be tested and assessed in the absence of noise and of modeling uncertainties.

figure 3 illustrates the differences between the simulated and observed concentrations for two of the Prairie Grass

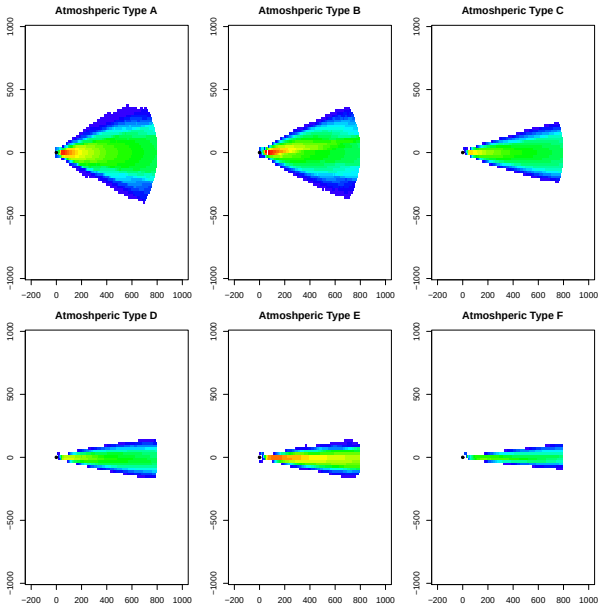


FIG. 2: Reconstructed concentration fields averaged over all releases in each atmospheric class, with the wind direction aligned with the horizontal axis.

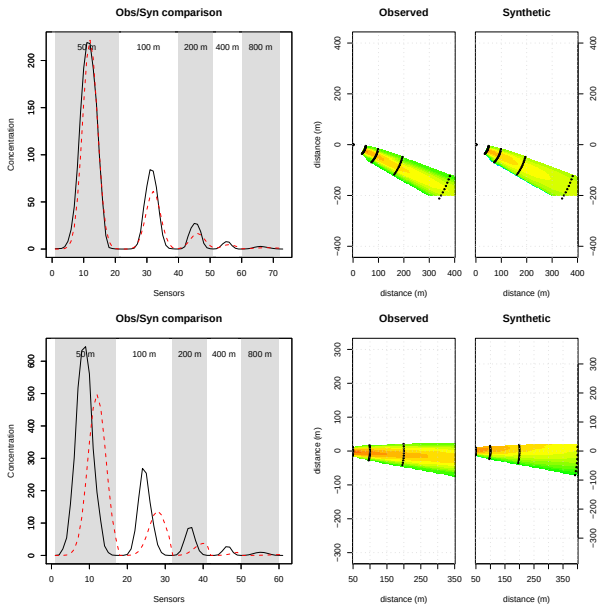


FIG. 3: Comparison between two synthetic and observed concentration fields. A small error in wind direction leads to large errors in the reconstructed field.

releases. The left graphs show the concentration observed at each sensor, plotted along with the simulated concentration profile. The sensors are positioned along the five concentric arcs indicated by alternate white and gray background. Within each arc, the sensors are sorted counter-clockwise. The middle and right graphs show the concentration field reconstructed from the measurements and from the simulated values respectively. The top graphs refer to release 55, which shows a very good agreement between the observed and simulated values. The bottom graphs show release 17, where there is a large discrepancy between the observed and simulated values. The shift in the location of the peak concentration is most likely due to errors in the measurement of the wind direction.

The accuracy of the simulations of all Prairie Grass experiments was assessed by calculating the following measure of the error between simulations and observations (Allen et al., 2007):

$$\Delta_c = \sqrt{\frac{\overline{[\log_{10}(C_o+1) - \log_{10}(C_p+1)]^2}}{\overline{[\log_{10}(C_o+1)]^2}}} \tag{6}$$

where $C_o$ and $C_p$ are the observed and predicted concentrations at the sensors locations, respectively, and the bar indicates an average over all the observations. The function $\Delta_c$ was shown to be a suitable objective function for source detection algorithms (Cervone and Franzese, 2010b). In this study, it will also be used as the objective function to be minimized by the search algorithm.

Equation (6) was calculated for each of the 68 Prarie Grass releases. Table 2 reports $\Delta_c$ and the relevant characteristics observed at the time of the release. The first column indicates the ID for the release (as assigned in the original dataset). The second column is the sum of all the concentrations measured at different sensors. The third column is the number of observations for each experiment, namely the number of sensors detecting concentration. In general, unstable atmospheric conditions (class A) corresponds to a large number of sensors because of enhanced plume spread. Conversely, unstable atmospheric conditions (class F) result in the smallest number of ground measurements, because the footprint of the dispersion is small and fewer sensors are active compared to the other stability classes. The errors associated with the experiments in atmospheric class F are mainly attributable to two factors:

- The fewer number of measurements compared to other atmospheric classes generates a larger uncertainty in the reconstructed field.

- Small errors in the observed wind direction cause large errors in the reconstructed field (e.g., see figure 3).

The remaining columns indicate the amount of material released, the wind velocity and direction, and the heat flux properties recorded at the time of the Prairie Grass field experiment. The last column indicates the atmospheric class associated with the conditions at the time of the experiment.

| $\psi$ | ID | HeatFlux (W/m$^2$) | Q (g/s) | U (m/s) | $\theta$ (deg) | # Measurements | $\Sigma_C^{ID}$ (mg/m$^3$) | $\Delta_c$ |
|---|---|---|---|---|---|---|---|---|
| A | 15 | 129.90 | 95.5 | 2.90 | 209 | 135 | 5554 | 0.38 |
| A | 16 | 213.54 | 93.0 | 2.96 | 192 | 158 | 3621 | 0.31 |
| A | 25 | 94.99 | 101.4 | 2.48 | 177 | 224 | 5747 | 0.47 |
| A | 47 | 217.33 | 103.1 | 3.02 | 243 | 148 | 4543 | 0.21 |
| A | 52 | 293.13 | 104.0 | 4.04 | 132 | 196 | 3999 | 0.44 |
| B | 1 | 81.50 | 81.5 | 2.39 | 150 | 223 | 4882 | 0.65 |
| B | 2 | 37.12 | 83.9 | 1.74 | 100 | 202 | 5231 | 0.88 |
| B | 7 | 265.13 | 89.9 | 4.02 | 188 | 260 | 3482 | 0.75 |
| B | 10 | 203.11 | 92.1 | 4.15 | 225 | 155 | 3407 | 0.38 |
| B | 48S | 140.17 | 104.0 | 2.77 | 216 | 206 | 3714 | 0.68 |
| C | 5 | 202.83 | 77.8 | 5.15 | 176 | 138 | 2759 | 0.35 |
| C | 8 | 127.33 | 91.1 | 4.06 | 184 | 139 | 4138 | 0.30 |
| C | 9 | 194.07 | 92.0 | 6.11 | 204 | 154 | 3101 | 0.42 |
| C | 19 | 187.53 | 101.8 | 5.33 | 166 | 126 | 3645 | 0.38 |
| C | 27 | 239.83 | 98.8 | 5.40 | 184 | 133 | 3559 | 0.39 |
| C | 43 | 229.18 | 98.9 | 4.68 | 170 | 189 | 4146 | 0.49 |
| C | 44 | 263.20 | 100.7 | 5.39 | 158 | 177 | 3774 | 0.51 |
| C | 49 | 270.34 | 102.0 | 6.03 | 199 | 159 | 3504 | 0.40 |
| C | 50 | 331.86 | 102.8 | 6.06 | 215 | 138 | 3498 | 0.35 |
| C | 62 | 106.64 | 102.1 | 4.61 | 212 | 127 | 4266 | 0.26 |
| D | 6 | 80.10 | 89.5 | 5.99 | 183 | 110 | 3268 | 0.47 |
| D | 11 | 157.05 | 95.9 | 6.77 | 184 | 97 | 2963 | 0.16 |
| D | 12 | 284.41 | 99.1 | 7.21 | 194 | 105 | 2834 | 0.39 |
| D | 17 | -15.71 | 56.5 | 2.87 | 184 | 61 | 5079 | 0.58 |
| D | 20 | 396.31 | 101.2 | 8.26 | 178 | 121 | 2879 | 0.55 |
| D | 21 | -29.26 | 50.9 | 5.31 | 181 | 74 | 2563 | 0.53 |
| D | 22 | -45.95 | 48.4 | 6.39 | 176 | 69 | 1847 | 0.45 |
| D | 23 | -26.78 | 40.9 | 5.37 | 128 | 79 | 1664 | 0.19 |
| D | 24 | -18.40 | 41.2 | 5.21 | 141 | 78 | 1642 | 0.14 |
| D | 26 | 200.05 | 97.6 | 5.68 | 190 | 161 | 3202 | 0.63 |
| D | 29 | -43.29 | 41.5 | 3.40 | 220 | 132 | 3568 | 0.51 |
| D | 30 | 243.64 | 98.4 | 6.28 | 196 | 131 | 3443 | 0.57 |
| D | 31 | 155.75 | 96.0 | 6.91 | 225 | 140 | 3102 | 0.54 |
| D | 33 | 180.74 | 94.7 | 6.90 | 181 | 116 | 2621 | 0.47 |
| D | 34 | 254.66 | 97.4 | 8.46 | 146 | 99 | 2518 | 0.30 |
| D | 35S | -22.23 | 41.8 | 3.40 | 135 | 67 | 3176 | 0.39 |
| D | 37 | -23.12 | 40.3 | 4.00 | 187 | 81 | 2045 | 0.29 |
| D | 38 | -18.13 | 45.4 | 3.70 | 170 | 63 | 2918 | 0.30 |
| D | 42 | -37.95 | 56.4 | 5.27 | 212 | 78 | 2440 | 0.27 |
| D | 45 | 62.81 | 100.8 | 5.31 | 163 | 103 | 4593 | 0.34 |
| D | 46 | -31.19 | 99.7 | 4.86 | 134 | 99 | 5612 | 0.26 |
| D | 48 | 198.21 | 104.1 | 6.91 | 214 | 105 | 2525 | 0.36 |
| D | 51 | 242.16 | 102.4 | 6.18 | 245 | 150 | 3755 | 0.58 |
| D | 54 | -32.86 | 43.4 | 3.40 | 140 | 62 | 2981 | 0.38 |
| D | 55 | -41.27 | 45.3 | 5.17 | 156 | 73 | 2039 | 0.21 |
| D | 56 | -28.52 | 45.9 | 4.15 | 153 | 73 | 2668 | 0.20 |
| D | 57 | 48.22 | 101.5 | 6.42 | 200 | 119 | 3601 | 0.40 |
| D | 60 | -36.20 | 38.5 | 4.04 | 198 | 66 | 2059 | 0.34 |
| D | 61 | 305.16 | 102.1 | 7.00 | 203 | 158 | 3153 | 0.61 |
| D | 65 | -47.79 | 44.1 | 3.93 | 178 | 59 | 2476 | 0.55 |
| D | 67 | -22.77 | 45.0 | 3.85 | 185 | 69 | 2600 | 0.19 |
| E | 18 | -24.52 | 57.6 | 2.68 | 187 | 68 | 5460 | 0.51 |
| E | 28 | -17.25 | 41.7 | 2.12 | 174 | 74 | 4832 | 0.31 |
| E | 41 | -32.58 | 39.9 | 3.16 | 198 | 49 | 2782 | 0.38 |
| E | 66 | -33.58 | 43.1 | 2.56 | 166 | 94 | 4357 | 0.35 |
| E | 68 | -16.88 | 42.8 | 2.19 | 174 | 75 | 5236 | 0.24 |
| F | 3 | -0.68 | 56.3 | 0.66 | 150 | 289 | 5550 | 1.00 |
| F | 4 | -4.54 | 50.5 | 0.91 | 216 | 443 | 10619 | 0.86 |
| F | 13 | -4.38 | 61.1 | 0.92 | 190 | 99 | 7062 | 0.68 |
| F | 14 | -2.13 | 49.1 | 0.74 | 170 | 153 | 8990 | 0.52 |
| F | 32 | -17.72 | 41.4 | 1.60 | 171 | 64 | 7167 | 0.32 |
| F | 35 | -7.45 | 38.8 | 1.10 | 132 | 66 | 7497 | 0.71 |
| F | 36 | -10.00 | 40.0 | 1.37 | 160 | 70 | 7530 | 0.47 |
| F | 39 | -17.95 | 40.7 | 1.69 | 140 | 79 | 3919 | 0.48 |
| F | 40 | -15.39 | 40.5 | 1.58 | 180 | 102 | 4011 | 0.73 |
| F | 53 | -20.44 | 45.2 | 1.56 | 132 | 47 | 6625 | 0.54 |
| F | 58 | -18.44 | 40.5 | 1.65 | 178 | 42 | 6881 | 0.50 |
| F | 59 | -22.24 | 40.2 | 2.02 | 174 | 47 | 5583 | 0.29 |

Table 2: Characteristics of the 68 Prarie Grass field experiments, and the calculated error $\Delta_c$ between the simulated and observed concentrations.

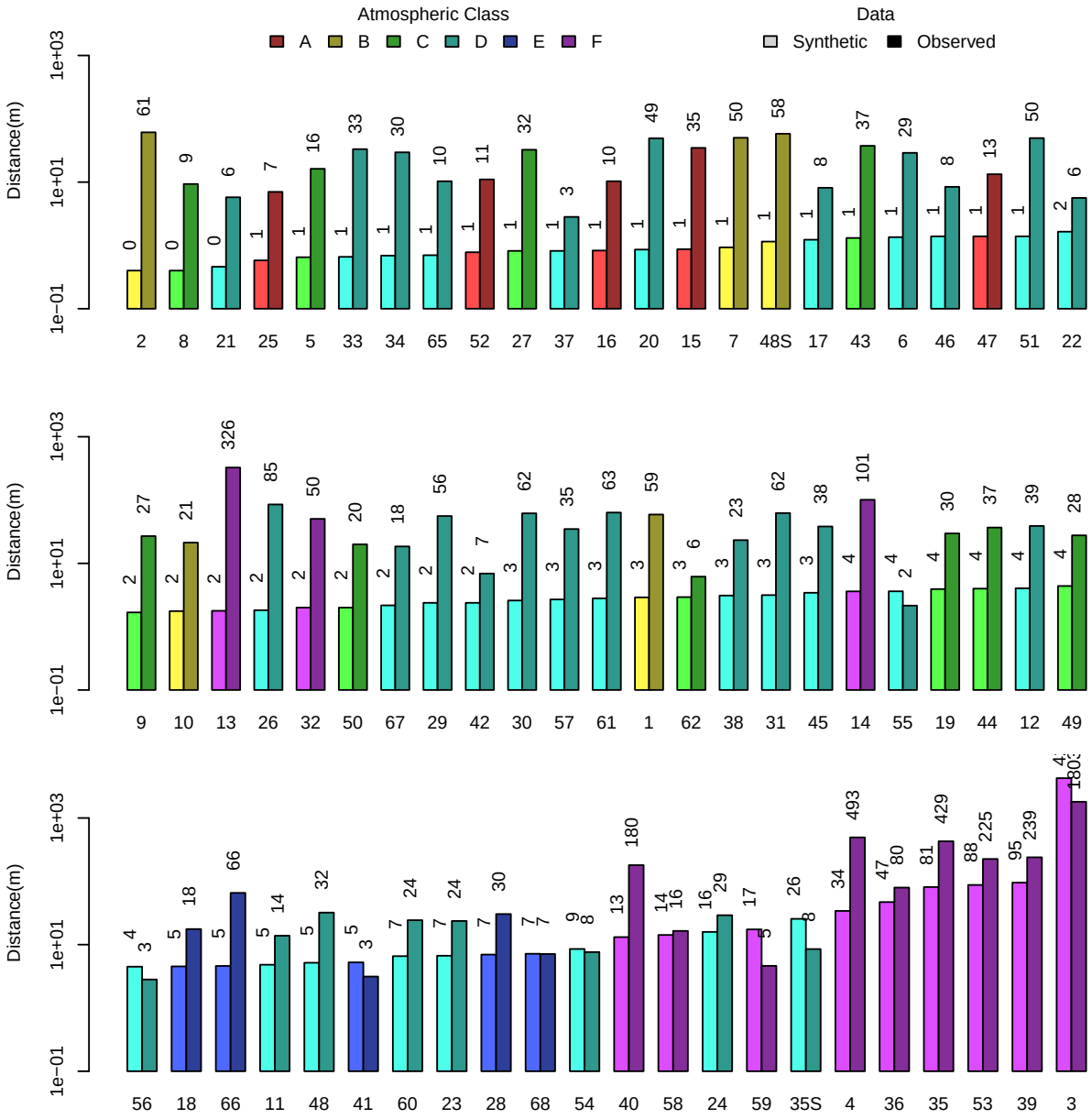


FIG. 4: Summary of the predicted distance from the real source using the observed Prairie Grass data and the synthetic data

Table 3: Summary of mean predicted distance from the real source for the synthetic dataset and for the Prairie Grass dataset

| Stability Class | A | B | C | D | E | F |
|---|---|---|---|---|---|---|
| Mean d (m), Synthetic | 0.89 | 1.42 | 2.21 | 3.97 | 5.70 | 387.88 |
| Mean d (m), Prairie Grass | 15.27 | 49.92 | 24.26 | 27.88 | 24.76 | 328.94 |

## 3. Results

The proposed methodology was tested to reconstruct the source characteristics for both the synthetic and the observed dataset. A total of 136 reconstructions were performed (68 for each dataset). Each reconstruction is the average of 30 runs, where each run differs only from the initial randomly generated population of candidate solutions. A total of 4080 runs are performed, requiring under 6 hours of CPU time on a dual core E8400 3.00GHz computer. figure 4 shows a summary of the predicted distance from the real source using the observed Prairie Grass data and the synthetic data. The atmospheric class of each experiment is also shown. The light color in each pair of columns indicates the synthetic dataset, the darker tone indicates the observed Prairie Grass dataset. The numbers at the base of each column indicate the Prairie Grass experiment identifier, while the numbers at the top of the columns report the value of the predicted distance from the source in meters. In most cases, the difference in results are very small. Worst results are consistently obtained with atmospheric class F. This is to be expected as it is consistent with the inferior performance of the dispersion model with the cases of class F, as shown in Table 2. In general, the results show that the proposed method performs consistently better with the synthetic data, indicating that it is able to perfectly characterize the source in the absence of noise and error measurements. However, even with the observed data, the errors are usually very small, in most cases about or smaller than 50 m, which is the distance from the source at which the closest ground measurements are made. Table 3 summarizes the mean predicted distance from the real source for both the synthetic and the Prairie Grass datasets.

### 3.1 *Sensitivity to the number of sensors and their measured concentrations*

In the following, we investigate the relationship between the amount of information provided to the search algorithm and the final error in the characteristics of the source. The first test we performed consist in a series of runs using the observed dataset, but only using a limited number of sensors. The runs were performed using an increasing number of sensors, selected at random. For each number of sensors, 30 different combinations of sensors were used. Each combination was run 30 times with different initial random candidate solutions. All 68 releases were simulated. figure 5 shows the cumulative error for all 68 Prairie Grass experiments as a function of the number of sensors actually used, and as a function of the atmospheric class of the experiments. Each bar in figure 5 was generated by plotting the results of 61200 runs, and shows the results obtained for 2, 3, 4, 5, 6, 8, 10, 12, 15 and 20 sensors, grouped by atmospheric class. The error decreases

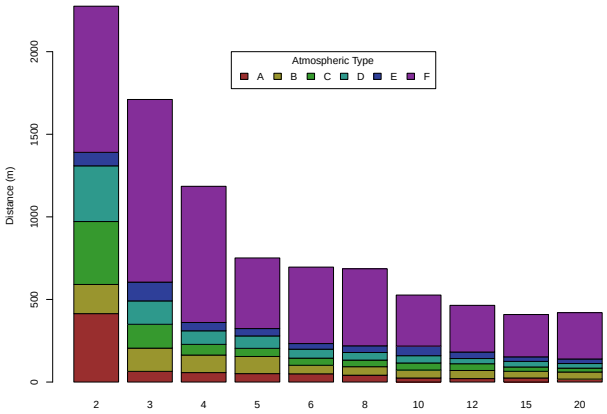


FIG. 5: Cumulative error for all the 68 Prarie Grass cases as a function of number of sensors and atmospheric stability class.

non-linearly with the number of sensors. In particular, the test shows that beyond a minimum number of sensors, there is little gain using a large amount of information. For instance, acceptable results are obtained with as little as 5 sensors selected at random. As expected, figure 5 also shows that the reconstructions of experiments conducted in atmospheric class F are noticeably worse than the other cases.

Further insight into the relation between amount of information required and performance of the search algorithm is gained by another experiment, in which the measurements were not chosen at random, but in such a way that the sum of the concentration over the selected sensors, i.e. the cumulative concentration, would be within a threshold range. figure 6 shows the predicted distance to the real source location as a function of the cumulative concentration reading, when the search algorithm is run using only 3, 4, 6, 8, 10 and 12 sensors. Using a larger number of sensors gives results essentially comparable with the case of 12 sensors and are not reported. In order to eliminate the effects of different atmospheric classes, we performed the experiments using only releases of class D, indicating a neutral atmosphere. The results clearly show that the number of sensors itself is not a determining factor in the performance of the algorithm. While there is an overall slight improvement in the results obtained with large number of sensors, figure 6 shows that the fundamental quantity governing performance is in fact the cumulative concentration over the sensors. Simulations conducted using only three sensors give results comparable to simulations using twelve or more sensors, as long as the cumulative distribution over the sensors is the same. In particular, the distance $d$ of the predicted source from the real source is found to decreases with an approximate slope $-2/3$, independent of the number of sensors considered:

$$d \approx k \left( \sum_n C_o^{(n)} \right)^{-2/3} \tag{7}$$

where $k$ is a constant, and $C_o^{(n)}$ is the measurement of concentration at the sensor $n$. At this stage it is not possible to determine a minimum number of sensors over which

Equation (7) describes the behavior of the algorithm. This will likely depend on the characteristics of the problem, and perhaps on the dispersion model used. In our case, the trend is observed already using 3 sensors, but for more complex scenarios it is possible that the minimum number of sensors necessary to the source characterization algorithm may be larger than 3.

## 4. Conclusions

This paper presented an innovative non-Darwinian evolutionary algorithm applied to the problem of identifying the location and characteristics of an unknown source of atmospheric contaminant. The proposed method is particularly advantageous in problems with complex evaluation functions, where the additional computational complexity introduced by the learning process is offset by the smaller number of evolutions required for convergence.

The methodology was tested using the real concentration and meteorology measurements from the Prairie Grass field experiment. An additional synthetic data was also generated to conduct sensitivity studies in a noiseless environment. The solutions were generally very accurate, e.g. the source was often located within a few meters from the real source.

The relationship between the number of sensors, and the accuracy of the solutions was investigated. A power law relationship was found characterizing the calculated distance from the real source, and the cumulative concentrations measured at the sensors. The power law relationship was found to be independent of the number of sensors used, beyond a minimum threshold of 3 sensors.

## 5. Acknowledgments

Work performed under this project has been partially supported by the NSF through Award 0849191 and by George Mason University Summer Research Funding.

## REFERENCES

Allen, C. T., G. S. Young, and S. E. Haupt, 2007: Improving pollutant source characterization by better estimating wind direction with a genetic algorithm. *Atmospheric Environment*, **41**(11), 2283–2289.

Arya, P. S., 1999: *Air pollution meteorology and dispersion*. Oxford University Press, 310 pp.

Ashlock, D., 2006: *Evolutionary computation for modeling and optimization*. Springer-Verlag.

Bäck, T., 1996: *Evolutionary Algorithms in Theory and Practice: Evolutionary Straegies, Evolutionary Programming, and Genetic Algorithms*. Oxford University Press.

Barad, M., 1958: Project Prairie Grass, a field program in diffusion. Technical Report Geophysical Research Paper, No. 59, Report AFCRC-TR-58-235, Air Force Cambridge Research Center, 218pp.

Cervone, G., and P. Franzese, 2010a: Machine learning for the source detection of atmospheric emissions. *Proceedings of the 8th Conference on Artificial Intelligence Applications to Environmental Science*, number J1.7.

Cervone, G., and P. Franzese, 2010b: Monte Carlo source detection of atmospheric emissions and error functions analysis. *Computers & Geosciences*, **36**(7), 902–909.

Cervone, G., P. Franzese, and A. Gradjeanu, 2010: Characterization of atmospheric contaminant sources using adaptive evolutionary algorithms. *Atmospheric Environment*, **44**, 3787–3796.

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

Cervone, G., K. Kaufman, and R. Michalski, 2000: Experimental validations of the learnable evolution model. *Evolutionary Computation, 2000. Proceedings of the 2000 Congress on Evolutionary Computation*, volume 2.

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

Cervone, G., L. Panait, and R. Michalski, 2001: The development of the AQ20 learning system and initial experiments. *Proceedings of the Fifth International Symposium on Intelligent Information Systems, June 18-22, 2001, Zakopane, Poland*, Physica Verlag, 13.

Darwin, C., 1859: *On the Origin of Species by Means of Natural Selection, or the Preservation of Favoured Races in the Struggle for Life*. Oxford University Press, London.

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

Grefenstette, J., 1991: Lamarckian learning in multi-agent environments. *Proceedings of the Fourth International Conference on Genetic Algorithms*.

Hamda, H., F. Jouve, E. Lutton, M. Schoenauer, and M. Sebag, 2002: Compact unstructured representations for evolutionary design. *Applied Intelligence*, **16**(2), 139–155.

Haupt, S. E., G. S. Young, and C. T. Allen, 2007: A genetic algorithm method to assimilate sensor data for a toxic contaminant release. *Journal of Computers*, **2**(6), 85–93.

Johannesson, G., B. Hanley, and J. Nitao, 2004: Dynamic bayesian models via monte carlo - an introduction with examples -. Technical Report UCRL-TR-207173, Lawrence Livermore National Laboratory.

Lozano, J., 2006: *Towards a New Evolutionary Computation: Advances in the Estimation of Distribution Algorithms*. Springer.

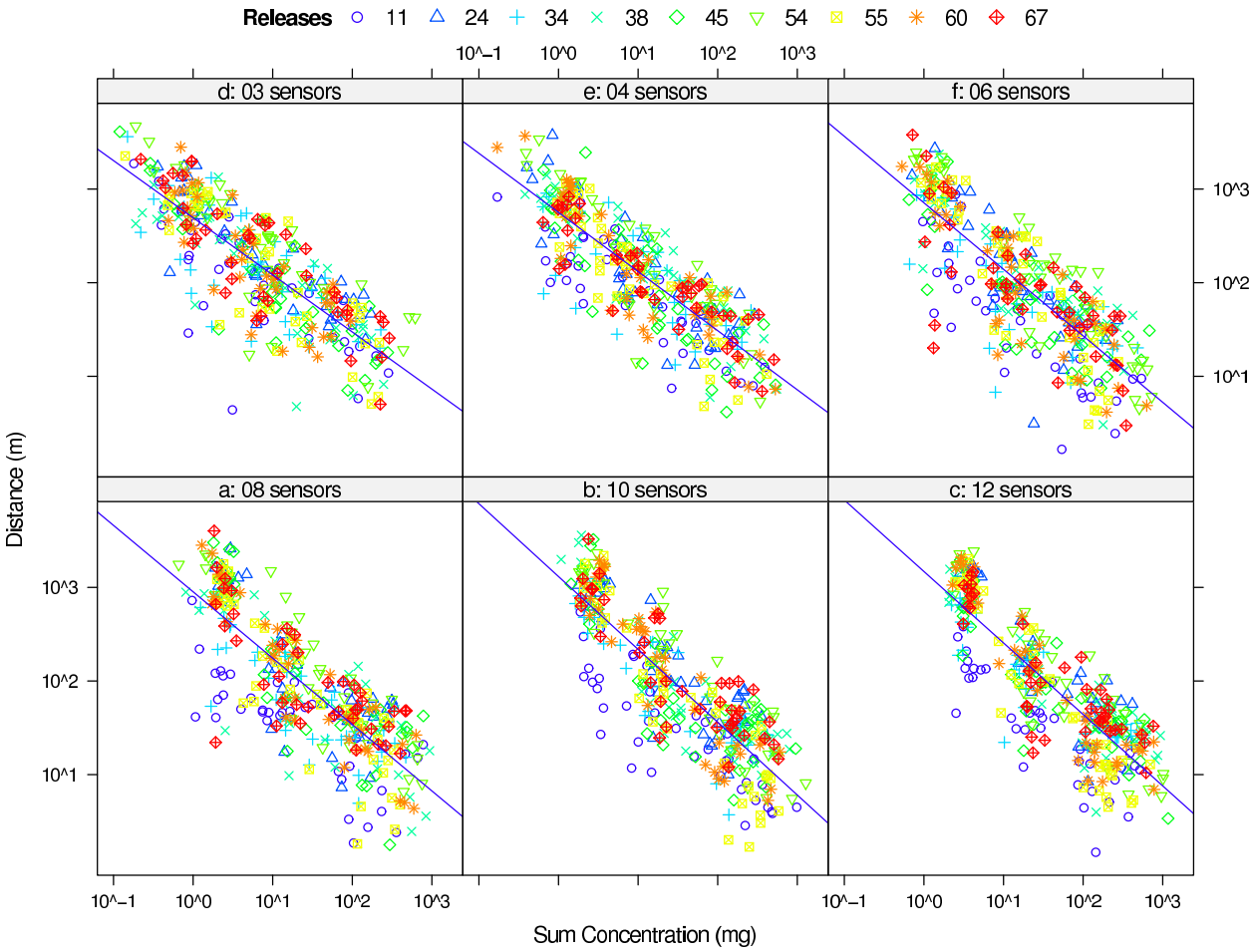


FIG. 6: shows the predicted distance to the real source location as a function of the cumulative concentration reading, when the search algorithm is run using only 3, 4, 6, 8, 10 and 12 sensors.

Michalski, R., 1969: On the quasi-minimal solution of the general covering problem. *Proceesings of Fifth International Symposium on Information Processing (FCIP 69)*, volume A3, 125–128.

Michalski, R., 1983: A theory and methodology of inductive learning. *Machine Learning: an Artificial Intelligence Approach*, **1**, 83–134.

Mitchell, T., 1997: Machine learning. *Mac Graw Hill*.

Reynolds, R., 1999: Cultural algorithms: Theory and applications. *Mcgraw-Hill'S Advanced Topics In Computer Science Series* 367–378.

Sebag, M., M. Schoenauer, and C. Ravise, 1997: Inductive learning of mutation step-size in evolutionary parameter optimization. *Lecture Notes in Computer Science* 247–261.

Senocak, I., N. Hengartner, M. Short, and W. Daniel, 2008: Stochastic event reconstruction of atmospheric contaminant dispersion using Bayesian inference. *Atmospheric Environment*, **42**(33), 7718–7727.