Proceedings of the 2014 Federated Conference on Computer Science and Information Systems pp. 519–527
DOI: 10.15439/2014F272
ACSIS, Vol. 2

Data-driven Genetic Algorithm in Bayesian estimation of the abrupt atmospheric contamination source

A. Wawrzynczak
1) National Centre for Nuclear Research, Świerk-Otwock, Poland
2) Institute of Computer Sciences, Siedlce Univesity, Poland
Email: a.wawrzynczak@ncbj.gov.pl

M. Jaroszynski
Institute of Computer Sciences, Siedlce Univesity, Poland
Email: marcinjaro89@gmail.com

M. Borysiewicz
National Centre for Nuclear Research, Świerk- Otwock, Poland
Email: manhaz@ncbj.gov.pl

***Abstract*—We have applied the methodology combining Bayesian inference with Genetic algorithm (GA) to the problem of the atmospheric contaminant source localization. The algorithms input data are the on-line arriving information about concentration of given substance registered by sensors' network. To achieve rapid-response event reconstructions the fast-running Gaussian plume dispersion model is adopted as the forward model. The proposed GA scan 5-dimensional parameters' space searching for the contaminant source coordinates (x,y), release strength (Q) and atmospheric transport dispersion coefficients. Based on the synthetic experiment data the GA parameters, best suitable for the contamination source localization algorithm performance were identified. We demonstrate that proposed GA configuration can successfully point out the parameters of abrupt contamination source. Results indicate the probability of a source to occur at a particular location with a particular release strength. We propose the termination criteria based on the probabilistic requirements regarding the parameters' value.**

I. Introduction

ACCIDENTAL atmospheric releases of hazardous material pose great risks to human health and the environment. In the event of an atmospheric release of chemical, but also radioactive biological materials, emergency responders need to quickly predict the current and future locations and concentrations of substance in the atmosphere. In this context it is valuable to develop the emergency system, which based on the concentration of dangerous substance by the sensors' network can inform about probable location of the release source. Moreover, the contamination source's location should be found as soon as possible. The most obvious way is to propose the simulation which gives the same substance point concentrations like registered by the sensors. However, to create the model realistically reproducing the real situation based only on the sparse point-concentration data is not trivial. This task requires specification of set of models' parameters, which depends on the applied model. The event reconstruction problem can be reformulated into a solution based on efficient sampling of an ensemble of simulations, guided by comparisons with data.

This work was supported by the Welcome Programme of the Foundation for Polish Science operated within the European Union Innovative Economy Operational Programme 2007-2013

A comprehensive literature review of past works on solutions of the inverse problem for atmospheric contaminant releases can be found in (e.g.[1]). The problem of the source term estimation was studied in literature grounded both on the deterministic and probabilistic approach. [2] implemented an algorithm based on integrating the adjoint of a linear dispersion model backward in time to solve a reconstruction problem. [3] introduced dynamic Bayesian modeling, and the Markov Chain Monte Carlo (MCMC) sampling approaches to reconstruct a contaminant source. The effectiveness of MCMC in the localization of the atmospheric contamination source based on the synthetic experiment data was presented in [4], [5]. The advantage of the Sequential Monte Carlo over the MCMC in the estimation of the probable values of the source coordinates was presented in [6].

The problem of finding the 'best fitted' model's parameters, for which a forward atmospheric dispersion model's output will reach agreement with real observations, can be considered as the optimization problem. Metaheuristics, such as genetic algorithms (GA), are broadly used to solve various optimization problems. GA was designed to imitate some of the processes that people can witness in natural environment [7]. By observing nature people noticed that many beings have evolved diametrically in the relatively short period of time. The concept of GA was to use the power of evolution to create a strong and universal tool reliable of solving optimization problems. The GAs are highly relevant for industrial applications, because they are capable of handling problems with non- linear constraints, multiple objectives, and dynamic components - properties that usually appear in the real-world problems (e.g. [8]). Since GA introduction and propagation the GA have been often used as an alternative to the conventional optimization methods and has been successfully applied in a variety of areas. For example it was used in control engineering [9], finding hardware bugs [11] and much more e.g. [10]. GA has been also used in

978-83-60810-58-3/$25.00

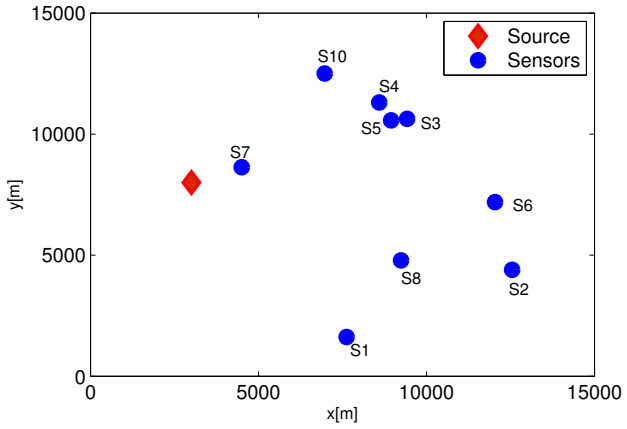

Fig. 1. Distribution of the sensors and the release source within the considered domain.

environmental sciences problem e.g. in the addressing air quality problem [12].

Application of the metaheuristic algorithms like GA requires defining the values of several algorithm components and parameters. These parameters have large impact on performance and efficiency of the algorithm (e.g., [13], [14], [15]). Therefore, it is important to estimate the algorithm's parameters best suitable for the considered optimization problem. The optimal values for the parameters depend mainly on: a) the problem; b) the domain of the problem to deal with; and c) computational time that can be spend for solving the problem. Usually, in the algorithm parameters tuning a compromise between solution quality and computational time should be achieved.

In this paper we apply the GA to the problem of localizing the abrupt atmospheric contamination source based on point-concentrations reported by the sensors network. Using the synthetic experiment data we demonstrate the efficient GA configuration best suitable for the algorithm performance.

A. Synthetic data

Our main goal is to conduct dynamic inference of an unknown atmospheric release. To test the proposed methods we require some concentration data. To satisfy this requirement we have performed the simulation with use of the atmospheric dispersion second-order Closure Integrated PUFF model (SCIPUFF) [16]. SCIPUFF is an ensemble mean dispersion model designed to compute the time-dependent field of expected concentrations resulting from one or more sources. The model solves the transport equations using a second-order closure scheme and treats releases as a collection of Gaussian puffs. In simulation we assumed that we have 10 sensors distributed over 15km x 15km area, the location of sensors was chosen randomly within the domain (Fig. 1). The atmospheric contamination source was located at $x = 3$ km, $y = 8$ km, $H = 25$ m within the domain. The simulated release was continuous with rate $Q = 8000 g/s$ and started one hour before first sensors measurements. The wind was directed along x axis with speed $5 m/s$. Further, in this paper we assume that the only algorithm input information we have, are reported every 15 minutes (in subsequent time steps) during 1.5 hour concentrations of dispersed substance registered by 10 sensors (Table I). We run algorithm searching for the source of contamination just after first information from sensors (t=1) and update the obtained probabilities with use of the developed algorithms by subsequent sensors registrations.

TABLE I
CONCENTRATION $[g/m^3]$ REPORTED BY SENSORS IN SUBSEQUENT TIME INTERVALS

Sensor	t=1	t=2	t=3	t=4	t=5	t=6
S1	0	0	0	0	0	0
S2	0	3.62E-09	4.93E-09	6.98E-09	4.15E-09	6.65E-09
S3	9.15E-09	2.88E-08	1.97E-08	1.88E-08	1.69E-08	1.62E-08
S4	3.83E-12	1.77E-11	4.89E-12	6.53E-12	2.31E-12	7.77E-12
S5	1.14E-08	1.83E-08	1.25E-08	1.20E-08	1.10E-08	1.03E-08
S6	2.91E-06	4.85E-04	4.77E-04	4.71E-04	4.43E-04	4.49E-04
S7	3.28E-05	3.27E-05	3.21E-05	3.13E-05	3.01E-05	2.87E-05
S8	2.29E-11	2.15E-10	1.05E-10	1.17E-10	7.56E-11	1.14E-10
S9	0	0	0	0	0	0
S10	0	0	0	0	0	0

II. RECONSTRUCTION PROCEDURE

A. Bayesian inference

The Bayes' theorem, as applied to an emergency release problem, can be stated as follows:

$$P(M|D) \propto P(D|M)P(M) \tag{1}$$

where M represents possible model configurations or parameters and D are observed data. For our application, Bayes' theorem describes the conditional probability $P(M|D)$ of certain source parameters (model configuration M) given observed measurements of concentration at sensor locations (D). This conditional probability $P(M|D)$ is also known as *a posteriori* distribution and is related to the probability of the data conforming to a given model configuration $P(D|M)$, and to the possible model configurations $P(M)$, before taking into account the measurements. The probability $P(D|M)$, for fixed D, is called the *likelihood* function, while $P(M)$- *a priori* distribution [17]. To estimate the unknown source parameters M using (1), the posteriori distribution $P(M|D)$ must be sampled. $P(D|M)$ quantifies the likelihood of a set of measurements D given the source parameters M.

Value of likelihood for a sample is computed by running a forward dispersion model with the given source parameters M and comparison of the model predicted concentrations in the points of sensors location (within a considered domain) with actual observations D. The closer the predicted values are to the measured ones, the higher is the likelihood of the sampled source parameters.

As the sampling procedure we use an GA to obtain the posterior distribution $P(M|D)$ of the source term parameters given the concentration measurements at sensor locations. This way we completely replace the Bayesian formulation with a

sampling procedure to explore the model parameters' space and to obtain a probability distribution for the source location.

B. The likelihood function

A measure indicating the quality of the current GA population is expressed in terms of a likelihood function. This function compares the predicted from model and observed data at the sensor locations as:

$$\lambda(M) = -\frac{\sum_{i=1}^{N}[log(C_i^M) - log(C_i^E)]^2}{2\sigma_{rel}^2}, \quad (2)$$

where λ is the likelihood function, C_i^M are the predicted by the forward model concentrations at the sensor locations i, C_i^E are the sensor measurements, N is the number of sensors, σ_{rel}^2 is an error parameter which can be updated accordingly to the expected errors in the observations at given observational time interval, here fixed to 0.2.

C. Posterior distribution

The posterior probability distribution (1) is computed directly from the resulting GA generations and is estimated as:

$$P(M|D) = \frac{1}{n}\sum_{i=1}^{n}\delta(M_i - M), \quad (3)$$

which represents the probability of a particular model configuration M giving results that match the observations at sensor locations. Equation (3) is a sum over the entire GA generation. Thus $\delta(M_i - M) = 1$ when $M_i = M$, and 0 otherwise. If in the generation many chromosomes have the same configuration $P(M|D)$ increases through the summation increasing the probability for those contamination source parameters.

D. Forward dispersion model

A forward model is needed to calculate the concentration C_i^M at the points i of sensor locations for the tested set of model parameters M at each GA step. As a testing forward model we selected the fast-running Gaussian plume dispersion model (e.g. [18]).

The Gaussian plume dispersion model for uniform steady wind conditions can be written as follows:

$$C(x,y,z) = \frac{Q}{2\pi\sigma_y\sigma_z U} exp\left[-\frac{1}{2}\left(\frac{y}{\sigma_y}\right)^2\right] \times \quad (4)$$
$$\{exp\left[-\frac{1}{2}\left(\frac{z-H}{\sigma_z}\right)^2\right] + exp\left[-\frac{1}{2}\left(\frac{z+H}{\sigma_z}\right)^2\right]\}$$

where $C(x,y,z)$ is the concentration at a particular location, U is the wind speed directed along x axis, Q is the emission rate or the source strength and H is the height of the release; y and z are the distance along horizontal and vertical direction, respectively. In the equation (4) σ_y and σ_z are the standard deviation of concentration distribution in the crosswind and vertical direction. These two parameters are defined empirically for different stability conditions [19], [20]. In this case we restrict the diffusion to the stability class C (Pasquill type stability for rural area). In scanning algorithm we assumed

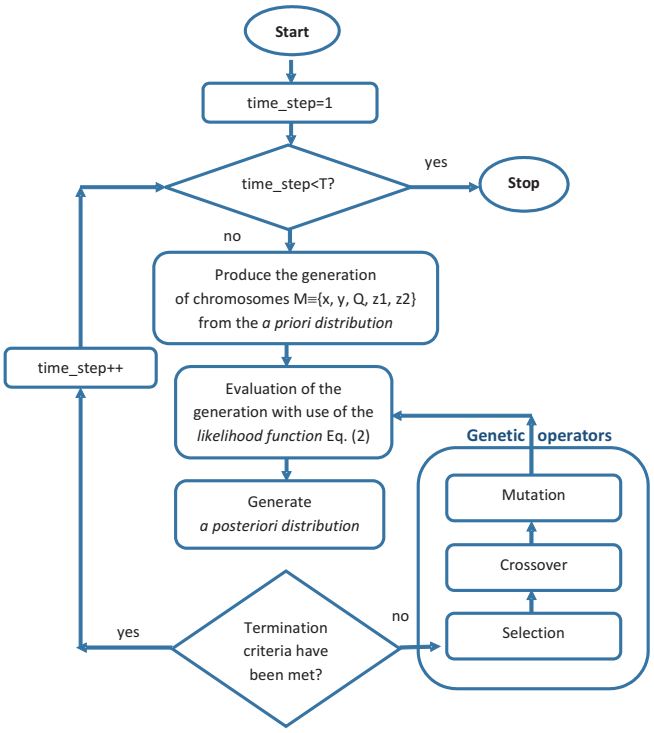

Fig. 2. Flow chart of the stochastic reconstruction procedure

that we do not know exact behavior of the plume and consider those coefficients as unknown. Thus, the parameters σ_y and σ_z are taken as: $\sigma_y = z_1 \cdot x \cdot (1 + x \cdot 4 \cdot 10^{-5})^{-0.5}, \sigma_z = z_2 \cdot x$ where values z_1 and z_2 are sampled by algorithm within interval $[0.001, 0.35]$.

To summarize, in this paper the searched model's parameter space is

$$M = (x, y, Q, z_1, z_2) \quad (5)$$

where x and y are coordinates of the release's source, Q release strength and z_1, z_2 are terms in the turbulent diffusion parametrization.

E. Genetic algorithm

The localization of the contamination source within the predefined domain requires the recognition of the atmospheric dispersion model parameters for which the model output at the sensors location meet the real data. In this context we can say that the problem can be seen as the optimization problem for which GA can be applied.

Fig. 2 presents the concept of GA's application in the Bayesian estimation of the unknown model parameters. The algorithm starts with the defining the initial population. The population is composed from the predefined number of chromosomes, $P(\tau) = x_1^\tau, \ldots, x_n^\tau$, for the generation τ, being initially randomly drawn from the admissible set of values. This set is explicitly defined by the space of explored parameters. GA chromosome is configured as binary value representing the real value of searched parameters. The quality

Algorithm 1 Rank Selection

```
ascSortMByLikelihoodFunction();
MProbabilityRange = 0;
FOR i=1 to N LOOP %N–population size
    M(i).rank = i-1; %M–chromosome
    probability = 2*(N-M(i).rank)/N*(N+1);
    MProbabilityRange += probability;
    M(i).probability = MProbabilityRange;
END LOOP
FOR i=1 to N LOOP
    randVal = drawNumberFrom0To1();
    FOR j=1 to N LOOP
        IF M(j).probability >= randVal
            newPopulation(i) = M(j);
            break;
        END IF
    END LOOP
END LOOP
```

Algorithm 2 Hard tournament selection

```
FOR i=1 to N LOOP
  FOR j=1 to TS LOOP
        tournamentGroup(j)=
                =drawSpecimenFromPopulation();
  END LOOP
  sortTournamentGroupByLikelihoodFunction();
  newPopulation(i) = getBestTournamentSpecimen();
END LOOP
```

of each chromosome in current population is evaluated based on the cost, or objective/likelihood function. Various objective functions can be applied; its form depends upon the problem being solved. We use the function presented by eq. (2). The 'improvement' of the current population can be done by the various genetic operators.

Information on the quality of population's chromosomes is used to perform selection. The portion of the population that is replaced in each generation is done based rank on the likelihood function (Eq.2) value obtained during the evaluation of the population (various in each algorithm iteration). Then, the crossover is performed. Crossover is process of replacing parents by their children in the current population. Children are created by blending of the parents at the randomly chosen crossover point. The number of crossovers that occurs within the population is determined by the crossover probability. Subsequently the current population is mutated. It changes the chromosome's features. By giving a chance of changing chromosome's individual bits mutation allows the algorithm to search for the entire solution's space and not to converge to local extremes. The number of mutations that occurs is determined by the mutation probability. After performing the selection crossover and mutation the new generation $(\tau + 1)$, being subject to the new evaluation, is established. After some number of generations the algorithm converges - it is expected that the best chromosome represents a near-optimum (reasonable) solution. The process stops when the termination criterion is fulfilled. The most common termination criterion is limited number of generations, but in this paper we present

Algorithm 3 Multi-point Crossover.

```
FOR i=1 to N LOOP %N–population size
    IF drawNumberFrom0To1() <= CP
      currentPopulation(i).isParrent(true);
    END IF
END LOOP

WHILE existsTwoNotUsedParents() LOOP
  firstParent  = popParent();
  secondParent = popParent();

  xCrossPoint = drawNumberFrom0ToParameterXLength();
  yCrossPoint = drawNumberFrom0ToParameterYLength();
  qCrossPoint = drawNumberFrom0ToParameterQLength();
  z1CrossPoint= drawNumberFrom0ToParameterZ1Length();
  z2CrossPoint= drawNumberFrom0ToParameterZ2Length();

  tmpXBin1 = firstParent.getXParameterBinaryForm();
  tmpYBin1 = firstParent.getYParameterBinaryForm();
  tmpQBin1 = firstParent.getQParameterBinaryForm();
  tmpZ1Bin1= firstParent.getZ1ParameterBinaryForm();
  tmpZ2Bin1= firstParent.getZ2ParameterBinaryForm();

  tmpXBin2 = secondParent.getXParameterBinaryForm();
  tmpYBin2 = secondParent.getYParameterBinaryForm();
  tmpQBin2 = secondParent.getQParameterBinaryForm();
  tmpZ1Bin2= secondParent.getZ1ParameterBinaryForm();
  tmpZ2Bin2= secondParent.getZ2ParameterBinaryForm();

  firstChildX  = tmpXBin1(0,CrossPoint)+
               + tmpXBin2(CrossPoint+1);
  firstChildY  = tmpYBin1(0,CrossPoint)
               + tmpYBin2(CrossPoint+1);
  firstChildQ  = tmpQBin1(0,CrossPoint)+
               + tmpQBin2(CrossPoint+1);
  firstChildZ1 = tmpZ1Bin1(0,CrossPoint)+
               + tmpZ1Bin2(CrossPoint+1);
  firstChildZ2 = tmpZ2Bin1(0,CrossPoint)+
               + tmpZ2Bin2(CrossPoint+1);

  secondChildX = tmpXBin2(0,CrossPoint)+
               + tmpXBin1(CrossPoint+1);
  secondChildY = tmpYBin2(0,CrossPoint)+
               + tmpYBin1(CrossPoint+1);
  secondChildQ = tmpQBin2(0,CrossPoint)+
               + tmpQBin1(CrossPoint+1);
  secondChildZ1= tmpZ1Bin2(0,CrossPoint)+
               + tmpZ1Bin1(CrossPoint+1);
  secondChildZ2= tmpZ2Bin2(0,CrossPoint)+
               + tmpZ2Bin1(CrossPoint+1);

  firstChild = firstChildX+firstChildY+firstChildQ
             + firstChildZ1+firstChildZ2;
  secondChild= secondChildX+secondChildY+secondChildQ
             + secondChildZ1+secondChildZ2;

  currentPopulation(firstParent.getId())=firstChild;
  currentPopulation(secondParent.getId())=secondChild;
END LOOP
```

other possibility.

In this paper the scanned parameters space M is five-dimensional i.e. $M \equiv \{x, y, Q, z1, z2\}$. Correspondingly each population's chromosome $M(i)$ stores the following information:

- x, y - coordinates of contamination's source in [m],
- Q - strength of release in [g/s],
- $z1$, $z2$ - terms in the turbulent diffusion parametrization.

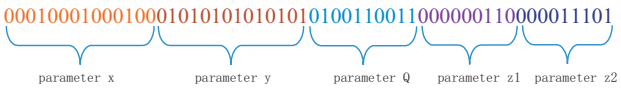

Fig. 3. Example of the chromosome representing the searched model's parameters

In the problem presented in this paper the parameters M are searched within the intervals $x \in \langle 0, 15000\rangle$, $y \in \langle 0, 15000\rangle$, $Q \in \langle 1, 8000\rangle$, $z1 \in \langle 0.001, 0.350\rangle$ and $z2 \in \langle 0.001, 0.350\rangle$. The parameters value precision P for parameters x,y equals $P_{x,y} = 1$ [m], for Q: $P_Q = 1$ [g/s], and $P_{z1} = P_{z2} = 0.001$. The example of the encoded chromosome presents Fig. 3.

1) Selection: There are many ways of dealing with GA selection e.g. roulette selection, rank selection, hard and soft tournament. For the problem presented in this paper the all mentioned methods were tested. The best results were achieved with selection based on rank and hard tournament selection. Results obtained applying these two selections are compared further in this paper. In the rank selection the better likelihood function results in the lower rank value leading to higher probability of being drawn to the next population. Pseudo code presents Algorithm 1. In the case of hard tournament selection of size 2, as the result of the tournament from each pair of the selected chromosomes one with the better objective function value passes to the next population. Pseudo code presents Algorithm 2.

2) Crossover: Similarly to the previous operator there are many ways of dealing with GA crossover e.g. single point crossover, multi point crossover, uniform crossover, arithmetic crossover. For a given problem the best results were achieved with by applying the multi-point crossover. Procedure begins with performing, for each chromosome, the test for being a parent according to the crossover probability CP. From the parents' population the unexploited pair is chosen, then one crossover point for each parameter encoded in the chromosome is drawn, i.e. five points for the problem presented. Parents are split at the crossover points for each encoded parameter, then (in term of each encoded parameter) bits are swap resulting in two children. Pseudo code presents Algorithm 3.

3) Mutation: The latter applied genetic operator is mutation. The most frequently used are uniform mutation and not-uniform mutation. For the given problem the best results were achieved with uniform mutation in which all chromosome's bits are mutated with the mutation probability MP. Pseudo code presents Algorithm 4.

In the reconstruction of the atmospheric contamination source the following GA configuration was applied:

- Size of population N=150;
- Selection:
  - rank selection,
  - hard tournament of size 2;
- Multi-point crossover with probability $CP = 0.75$, with 5 crossover points (5 is a number of searched parameters);
- Uniform mutation with probability $MP = 0.02$.

Algorithm 4 Uniform Mutation

```
FOR i=1 to N LOOP   %N-population size
  FOR j=1 to L LOOP %L-length of chromosome
                      %binary form
    IF drawNumberFrom0To1() <= MP
      currentPopulation(i).swapBitValue(j);
    END IF
  END LOOP
END LOOP
```

TABLE II
NUMBER OF GENERATIONS USED IN THE RECONSTRUCTION ALGORITHM WITH THE RANK SELECTION, $CP = 0.75$ AND $MP = 0.02$.

Time step	Generation's number	Forward dispersion model's runs
t=1	14	21 000
t=2	12	18 000
t=3	1	1 500
t=4	17	25 500
t=5	1	1 500
t=6	21	31 500
Summary	66	99 000

TABLE III
NUMBER OF GENERATIONS USED IN THE RECONSTRUCTION ALGORITHM WITH THE HARD TOURNAMENT OF SIZE 2 SELECTION, $CP = 0.75$ AND $MP = 0.02$.

Time step	Generation's number	Forward dispersion model's runs
t=1	140	210 000
t=2	124	186 000
t=3	62	93 000
t=4	97	145 500
t=5	113	169 500
t=6	216	324 000
Summary	752	1 128 000

The size of population, crossover probability and mutation probability were selected based on the numerical tests presented in [21].

III. RESULTS

We assume that the concentration from the sensors arrives subsequently in six time steps (Table I). We start to search for the source location (x, y), release rate (Q) and model parameters z_1 and z_2 after first sensors' measurements. Thus, reconstruction algorithm is run with obtaining the first measurements from the sensors ($t = 1$ at Table I). We assume that initially we have no *a priori* information about the parameters' values. So, the initial value of each parameter is draw randomly from the predefined interval with use of the uniform distribution.

Then generation is evaluated with use of the likelihood function (Eq. 2). The subsequent generations are iteratively updated by the applied genetic operators until the stop criterion is met. Of course there arises question how to specify the termination criteria? The usual criterion applied in GA is fixed number of generations. For the problem presented in this paper the time of giving the answer is crucial, so the constant number of generations is not optimal. In the task of the estimation of the source of the atmospheric contamination the most important is to

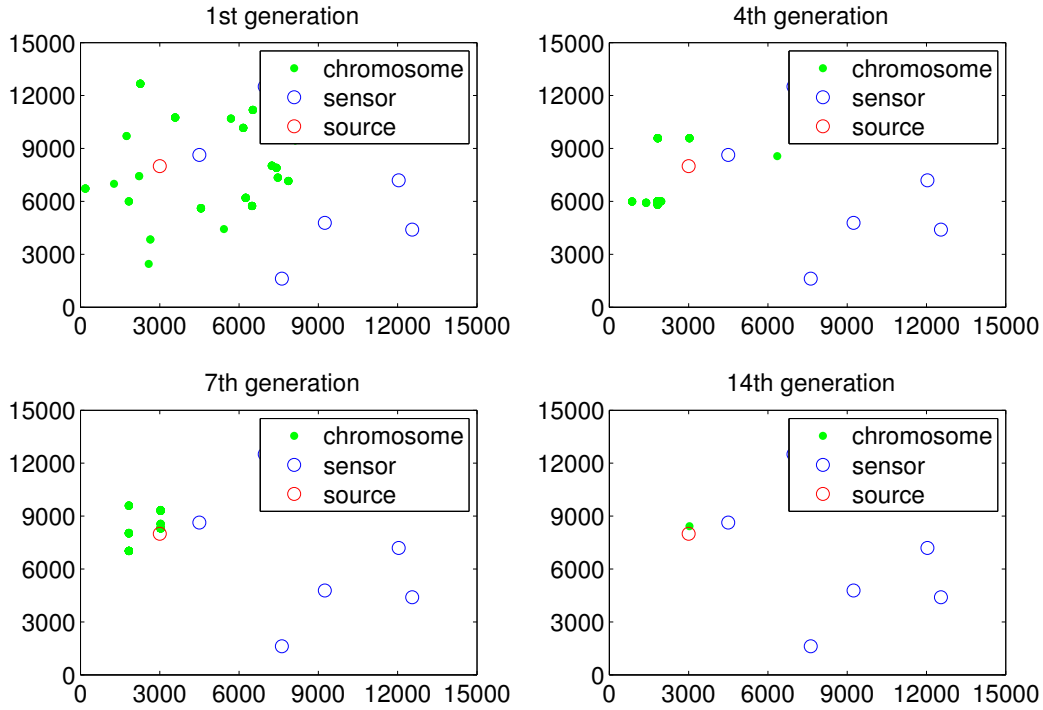

Fig. 4. Distribution of the x and y coordinates estimates during the GA runs for the given generation in 1st time step (rank selection).

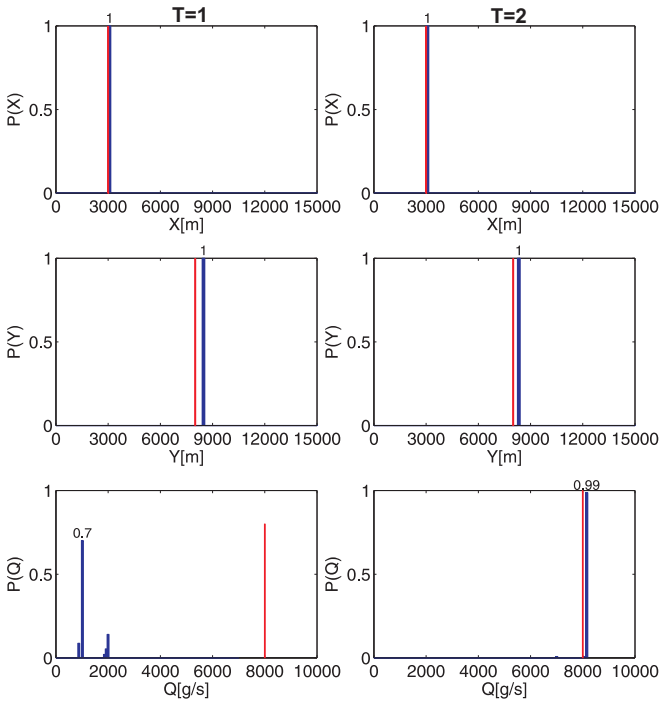

Fig. 5. Probability distributions of the models parameters x, y, and Q for the last generations in 1st and 2nd time step (rank selection). Vertical red lines represent the target value.

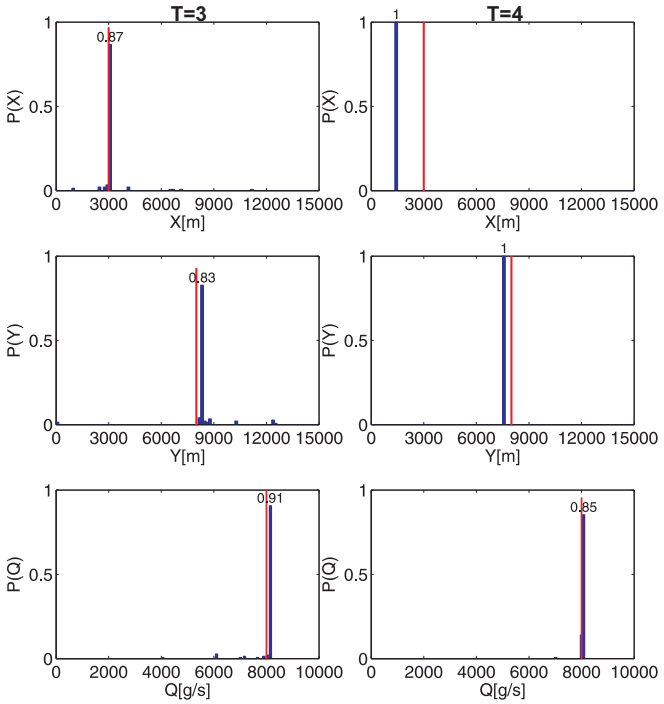

Fig. 6. Probability distributions of the models parameters x, y, and Q for the last generations in 3rd and 4th time step (rank selection). Vertical red lines represent the target value.

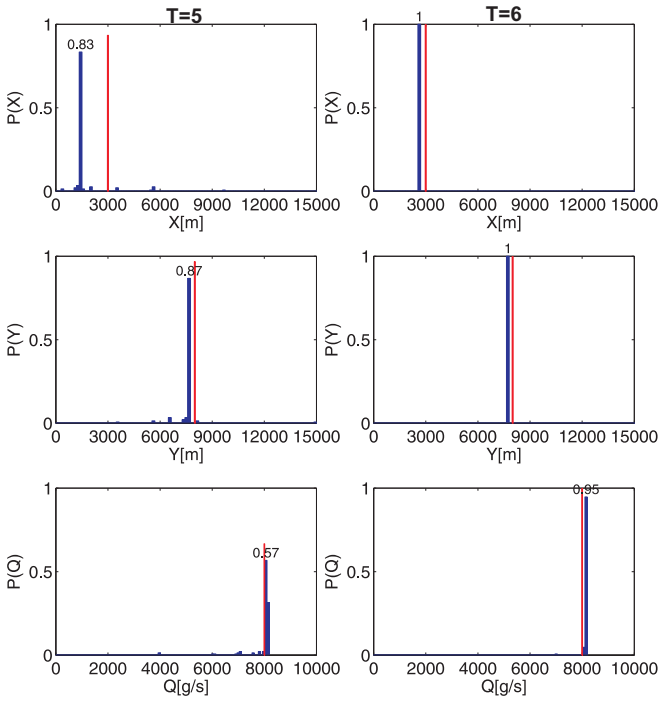

Fig. 7. Probability distributions of the models parameters x, y, and Q for the last generations in 5th and 6th time step (rank selection). Vertical red lines represent the target value.

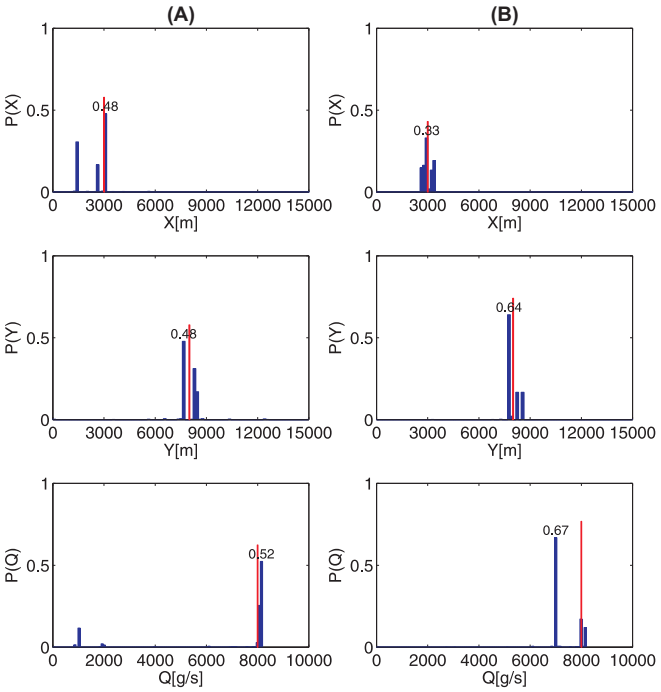

Fig. 8. Cumulative probability distributions of the models parameters x, y, and Q averaged over all time steps (A) rank selection, (B) hard tournament selection. Vertical red lines represent the target value.

estimate its location, to undertake the necessary action. Thus, crucial is assessment of the x and y coordinates of the source. Applying the Bayesian approach we can ask what probability of estimation of these parameters will be acceptable. So, after applying the last genetic operator, i.e. mutation, the histograms of x and y parameters encoded in the current chromosomes generation are evaluated. If many chromosomes have the same parameters configuration the probability of certain parameter's value increases. Consequently, the reconstruction algorithm is terminated when certain values of parameters x and y will be obtained with probability greater than 0.8. If this condition is fulfilled the *a posteriori* distributions of all parameters are calculated. Obtained *a posteriori* distributions are considered as the *a priori* distributions in the subsequent time step. Consequently, in the next time step, when new data from the sensors arrive the initial population is drawn uniformly from the *a priori* distribution i.e. *a posteriori* distribution from previous time step.

The number of generations required to fulfill the termination criterion in subsequent time steps for the rank selection is presented in Table II and for the hard tournament selection in Table III. Comparing the Tables it is obvious that the rank selection is much more effective. Fig. 4 illustrates the distribution of the estimated by the GA contamination source coordinates x and y in subsequent generations in the first time step. It is seen that at the beginning for the 1st generation the chromosomes are equally distributed within the scanned domain. However, the applied genetic operators improve population quality for further generations and the chromosomes gradually focus around the true source location. Finally, for 19th generation the estimated by the GA contamination source location approaches to the target location. Figs. 5, 6 and 7 present the a posteriori distributions for x, y and Q parameters obtained in the succeeding time steps. This distributions were obtained based on the chromosomes configurations in the last generation at given reconstruction algorithm iteration. Based on the searched parameters value, encoded in the final chromosomes population, the histogram for each parameter has been assessed. Obtained histograms shows which values of the parameters were the most frequent in the final generation, which directly is reflected in its probabilities.

Fig. 5 presents that the first sensors measurements allow to estimate the x and y parameters close to the target values, while the release strength Q is approached in the second time step. The probability distributions in subsequent time steps reflect how the sensor's data support or not the obtained distributions. The exact values of parameters differ in subsequent time steps. Below, as the estimated parameter value we provide the central value of the histogram bar with highest probability and as the error the half of the bar width. In the 6th time step the following parameters are estimated $P(x = 2625 \pm 75) = 1$, $P(y = 7725 \pm 75) = 1$ and $P(Q = 8120 \pm 40) = 0.95$. To effectively compare the results given by all proposed algorithms we have estimated the joint marginal distribution of x, y and Q parameters. Fig. 8ab present the *a posteriori* distribution averaged over all time steps for the GA algorithm

with rank selection and with the hard tournament selection, respectively. The algorithm applying rank selection as the most probable has pointed the parameters $P(x = 3075 \pm 75) = 0.48$, $P(y = 7725 \pm 75) = 0.48$ and $P(Q = 8120 \pm 40) = 0.52$, while the algorithm applying the hard tournament the parameters $P(x = 2925 \pm 75) = 0.33$, $P(y = 7725 \pm 75) = 0.64$ and $P(Q = 7000 \pm 40) = 0.67$. Fig. 9ab presents the probability distributions of the $z1$ and $z2$ parameters for the both selection methods. The algorithm applying rank selection returned the following values $P(z1 = 0.04375 \pm 0.00175) = 0.48$, $P(z2 = 0.00175 \pm 0.00175) = 0.79$ and algorithm applying the hard tournament $P(z1 = 0.05075 \pm 0.00175) = 0.48$, $P(z2 = 0.00175 \pm 0.00175) = 0.8$. We do not know the target values for these coefficient, as far the SCIPUFF model used to generate the synthetic concentration data do not allows to specify its directly. In the reconstruction procedure we could of course fix these coefficients according to the stability class pointed by the terrain and wind speed which in this case could be the stability class C for which $z1 = 0.22$ and $z2 = 0.2$. But our numerical tests showed that we obtain better results when we do not restrict the dispersion coefficients to the one given value. The 'freed' the dispersion coefficients in some acceptable interval assumption allows to better fit the Gaussian plum to the 'real' data.

Comparison of the obtained results leads to the conclusion that algorithms applying both selection methods return similar results for the x and y parameters, at the same time the algorithm using the hard tournament selection as the most probable denotes $Q = 7000$ which differs from the true release rate for $1000g/s$, while for the rank selection algorithm hits the target value. Consequently, we can pointed the algorithm applying the rank selection as more effective, as far it requires ~ 11 times less computational time than the hard tournament selection to return comparable results.

IV. Conclusion

We have presented a methodology to reconstruct a source causing an area of contamination, based on a set of measurements. The method combines Bayesian inference with the genetic algorithm and produces posterior probability distributions of the parameters describing the unknown source. Developed dynamic data-driven event reconstruction model couples data and pollutant dispersion simulations through Bayesian inference. This approach successfully provide the solution to the stated inverse problem i.e. having the downwind concentration measurement and knowledge of the wind field algorithm found the most probable location of the source and its strength.

We have proposed the termination criteria reflecting the probabilistic aspect of the obtained solution i.e. the GA is terminated when some of the searched parameters are pointed with satisfactorily probability. This approach allows to optimize the algorithm's computational time. We show that in the presented problem the rank selection is more efficient than the hard tournament selection.

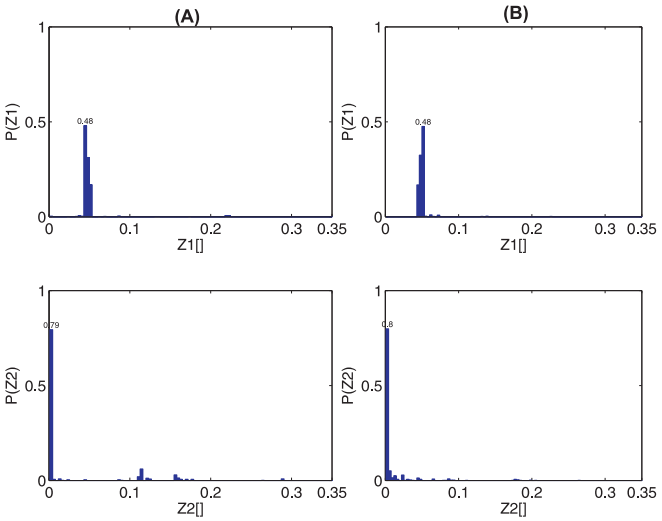

Fig. 9. Cumulative probability distributions of the models parameters $z1$ and $z2$ averaged over all time steps (A) rank selection, (B) hard tournament selection. Vertical red lines represent the value usually accepted in the Pasquili stability class C.

The probabilistic aspect of the solution optimally combines a probable answer with the uncertainties of the available data. Among several possible solutions, the Bayesian source reconstruction is solely able to find values of the model parameters that are more consistent with the currently available data.

Acknowlegements

Authors would like to thank the reviewers for helpful suggestions.

References

[1] Keats, A., E. Yee, and F.-S. Lien, (2007): Bayesian inference for source determination with applications to a complex urban environment. *Atmos. Environ.*, 41, 465-479, $doi : 10.1016/j.atmosenv.2006.08.044$.

[2] Pudykiewicz, J. A., (1998): Application of adjoint tracer transport equations for evaluating source parameters. *Atmos. Environ.*, 32, 303-3050, $doi : 10.1016/S1352 - 2310(97)00480 - 9$.

[3] Johannesson, G. et al., (2005): Sequential Monte-Carlo based framework for dynamic data-driven event reconstruction for atmospheric release.,*Proc. of the Joint Statistical Meeting*, Minneapolis, MN, American Statistical Association and Cosponsors, 73-80.

[4] Borysiewicz, M., Wawrzynczak A.,Kopka P,(2012):Stochastic algorithm for estimation of the model's unknown parameters via Bayesian inference, Proceedings of the Federated Conference on Computer Science and Information Systems pp. 501–508, IEEE Press, Wroclaw, ISBN 978-83-60810-51-4.

[5] Borysiewicz M., A.Wawrzynczak, P.Kopka,(2012): Bayesian-Based Methods for the Estimation of the Unknown Model's Parameters in the Case of the Localization of the Atmospheric Contamination Source, Foundations of Computing and Decision Sciences, 37, 4, 253–270, $doi : 10.2478/v10209 - 011 - 0014 - 9$.

[6] Wawrzynczak A., P. Kopka, M. Borysiewicz, (2014): Sequential Monte Carlo in Bayesian assessment of contaminant source localization based on the distributed sensors measurements, Lecture Notes in Computer Sciences 8385, PPAM 2013, Part II, ch.38, 407–417 , $doi : 10.1007/978 - 3 - 642 - 55195 - 6_38$.

[7] Holland J. H., (1992): Adaptation in Natural and Artificial Systems, 2nd Edn. Cambridge, MIT Press, 1992.

[8] Goldberg D. E., (2006): Genetic Algorithms in Search, Optimization and Machine Learning, Addison Wesley Longman, London, 2006.

[9] Fleming P. J., Fleming P. J., Purshouse R. C., and Purshouse R. C., (2001): Genetic Algorithms In Control Systems Engineering , In:Proceedings of the 12th IFAC World Congress, 383–390.

[10] Edited by Rustem Popa, (2012) Genetic Algorithms in Applications , ISBN 978-953-51-0400-1, InTech, Chapters published March 21, 2012 under CC BY 3.0 license doi : $10.5772/2675$

[11] Goodall R.M., Michail K., Whidborne J.F. Zolotas A.C, (2009): Optimised Configuration of Sensing Elements for Control and Fault Tolerance Applied to an Electro-Magnetic Suspension, PhD Thesis, Loughborough University, UK.

[12] Allen C.T, Haupt S. E., (2006): Source Characterization with a Genetic Algorithm-Coupled Dispersion-Backward Model Incorporating SCIPUFF, Department of Meteorology, The Pennsylvania State University, doi : $10.1175/JAM2459.1$.

[13] Eiben A . E., R. Hinterding and Z. Michalewicz, (1999): Parameter Control in Evolutionary Algorithms", IEEE Transactions on Evolutionary Computation, Vol. 3, No. 2, doi : $10.1109/4235.771166$

[14] Saremi A., T. Y. E. Mekkawy and G. G. Wang, (2007) Tuning the Parameters of a Memetic Algorithm to Solve Vehicle Routing Problem with Backhauls Using Design of Experiments", International Journal of Operations Research, Vol. 4, No. 4, 206–219.

[15] Roeva O., Stefka Fidanova, Marcin Paprzycki , Influence of the Population Size on the Genetic Algorithm Performance in Case of Cultivation Process Modelling , Proceedings of the 2013 Federated Conference on Computer Science and Information Systems, pages 371 - 376, 2013

[16] Sykes, R.I. et.al., (1998): PC-SCIPUFF Version 1.2PD Technical Documentation. ARAP Report No. 718. Titan Corporation,

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

[18] Turner D. Bruce, (1994): Workbook of Atmospheric Dispersion Estimates, *Lewis Publishers*, USA

[19] Pasquill, F. (1961): The estimate of the dispersion of windborne material, *Meteorol Mag.*,90, 1063,: 33-49

[20] Gifford, F. A. Jr. (1960): Atmospheric dispersion calculation using generalized Gaussian Plum model, *Nuclear Safety*, 2(2):56-59,67-68

[21] Wawrzynczak A et al. (2014): Recognition of the atmospheric contamination source localization with the Genetic Algorithm, Studia Informatica, UPH, Siedlce (*submitted*)