STUDIA INFORMATICA
Nr 1-2 (19) Systems and information technology 2015

Anna Wawrzynczak ^{1,2},
Marcin Jaroszynski ^{2},
Mieczyslaw Borysiewicz^{1}

^{1}National Centre for Nuclear Research, Swierk−Otwock, Poland
^{2}Institute of Computer Sciences, Siedlce University, Siedlce, Poland

Recognition of the atmospheric contamination source localization with the Genetic Algorithm

Abstract: We have applied the Genetic Algorithm (GA) to the problem of the atmospheric contaminant source localization. The algorithm input data are concentrations of given substance registered by sensor network. To achieve rapid-response event reconstruction,the fast-running Gaussian plume dispersion model is adopted as the forward model.
The proposed GA scans 5-dimensional parameters space searching for the contaminant source coordinates (x,y), release strength (Q) and the atmospheric transport dispersion coefficients. Based on the synthetic experiment data the GA parameters like population size, number of generations and the genetic operators best suitable for the algorithm performance are 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 rate. The shapes of the probability distribution function of searched parameters values reflect the uncertainty in observed data.

Keywords: Genetic algorithm, source characterization; atmospheric dispersion model

1. Introduction

Accidental atmospheric releases of hazardous material pose high risks to human health and the environment. In the event of an abrupt atmospheric release of chemical or radioactive, biological materials, emergency responders need to undertake the necessary action quickly to reduce the release consequences. In this context, it is important to develop the emergency system which based on the concentration of a dangerous substance reported by the network of sensors, can inform about probable location of the release source. It is crucial to point out the contamination source location as soon as possible.

The most evident way is to perform the simulation producing the material point concentrations comparable to registered by the sensors network. However, it is not trivial to create the model realistically reproducing the real situation based only on the sparse point-concentration data. This work requires the specification of the set of model parameters, which depend on the applied model. Consequently, the event reconstruction problem can be reformulated into a task of sampling an ensemble of simulations, guided by comparisons with data.

A comprehensive literature review of past works on solutions to the inverse problem for atmospheric pollutant releases can be found in (e.g. Keats, 2007). The problem of the source term evaluation was studied in the literature grounded both on the deterministic and probabilistic approach. Pudykiewicz, (1998) implemented an algorithm based on integrating the adjoint of a linear dispersion model backward in time to solve a reconstruction problem. Johannesson et al. (2004, 2005) 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 shown in Borysiewicz et al. (2012ab). Wawrzynczak et al. (2014) presented the advantage of the Sequential Monte Carlo over the MCMC in the estimation of the probable values of the source coordinates.

The task of finding the 'best fitted' model parameters for which a forward atmospheric dispersion model output will reach agreement with the real observations can be considered as an optimization problem.

Metaheuristics, such as genetic algorithms (GAs), are broadly used to solve various optimization problems. GA was designed to imitate some of the processes taking place in the natural environment (Holland, 1992). The GAs are highly relevant for industrial applications. Reason is their capabilities of handling problems with non-linear constraints, multiple objectives, and dynamic components – properties that usually appear in the real-world problems (e.g. Goldberg, 2006). Since GA introduction and propagation (Holland, 1992) it has been successfully applied in a variety of areas as an alternative to the conventional optimization methods. For example, it was used in control engineering (Chwee, 1995), finding hardware bugs (Goodall and Michail, 2009), multiple criteria production scheduling (Bagchi, 1999) and much more. GAs have been also used in environmental sciences problem e.g. in the addressing air quality problem (Allen and Haupt, 2006)

Application of the metaheuristics like GA requires defining several algorithm components and parameters. These parameters have a large impact on performance and efficiency of the algorithm (e.g., Eiben et al., 1999, Saremi et al., 2007, Roeva et al., 2013). Therefore, it is important to estimate the algorithm parameters best suitable for the considered optimization problem. The optimal values of parameters depend mainly on a) the problem; b) the domain of the problem to deal with and c) the computational time that can be spent on solving the problem. Usually in the algorithm parameters tuning should be achieved a compromise between solution quality and search time.

In this paper, we apply the GA to the problem of localization of the abrupt atmospheric contamination source based on the released substance concentration reported by the sensors network. Based on the synthetic experiment data the GA parameters, like population size, mutation, and crossover probability; best suitable for the algorithm performance are identified.

2. Problem formulation

The application of the GA to the issue of localization of the atmospheric contamination source is tested based on the synthetic data.The algorithm input data are the concentrations recorded by 10 sensors randomly distributed over the domain 15 km x 15 km (Figure 1). The synthetic concentrations (Figure 1), used in algorithm testing, were generated by the atmospheric dispersion Gaussian plume model (e.g. Turner, 1994). In the presented experiment, the contamination source was located at x = 3 km, y = 8 km, within the domain at the height $\widetilde{H} = 50\ m$ (Figure 1). The release rate was assumed to be Q = 5000 g/s. The wind was directed along x axis with speed U = 5 m/s.The sensors synthetic concentrations provided to the GA localization algorithm were randomly distorted up to 20% of initial concentration.

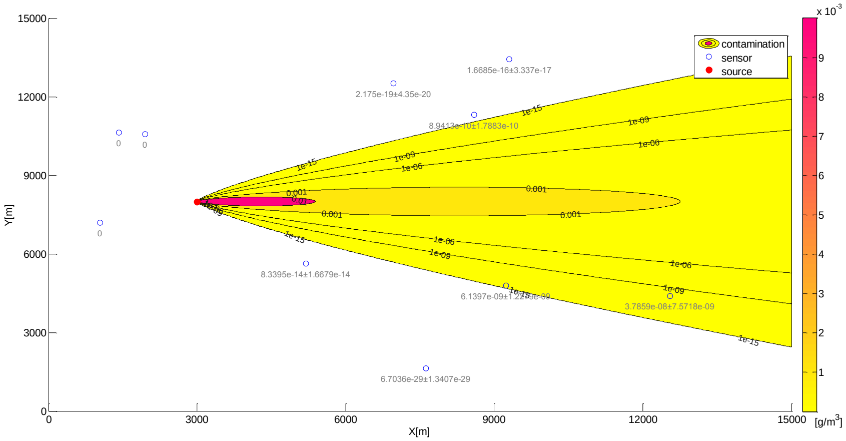

Figure 1. The location of the release source along with the spatial distribution of 10 sensors within the domain. The values for the sensors represent the substance concentration distorted randomly up to 20% percent when passed to the reconstruction procedure.

An atmospheric dispersion model is necessary 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. To satisfy the short computational time requirement, as a forward model we selected the fast-running Gaussian plume dispersion model (e.g. Turner, 1994). For uniform steady wind conditions the concentration $C(\tilde{x}, \tilde{y}, z)$ of the emission (in micrograms per cubic meter) at any point $\tilde{x}$ meters downwind of the source, $\tilde{y}$ meters laterally from the centerline of the plume, and z meters above ground level can be written as follows:

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

where U is the wind speed directed along x axis, Q is the emission rate or the source strength and H is the effective height of the release equal to the sum of the release height and plume rise ($H = \widetilde{H} + h$). In the equation (1) σ_y and $\sigma_z$are the standard deviations of concentration distribution in the crosswind and vertical direction and depends on $\tilde{x}$ (see formula below). These two parameters were defined empirically for different stability conditions by Pasquill (1961) and Gifford (1960). We restrict the diffusion to the stability class C in an urban area (Pasquill type stability for therural area). Thus, in the creation of the synthetic data we have fixed this coefficient as:

$$\sigma_y = 0.22\tilde{x} \cdot (1 + \tilde{x} \cdot 4 \cdot 10^{-5})^{-0.5},\ \sigma_z = 0.2\tilde{x}\ .$$

However, during the reconstruction we assume that we do not know the exact behavior of the plume and consider those coefficients as unknown. Thus, the parameters σ_y, σ_z are taken as:

$$\sigma_y = z_1 \cdot \tilde{x} \cdot (1 + \tilde{x} \cdot 4 \cdot 10^{-5})^{-0.5}, \sigma_z = z_2 \cdot \tilde{x}\,, \tag{2}$$

where values z_1 and z_2 are sampled by algorithm within the physically acceptable interval <0.001,0.35>.

The simple mathematical transformation of the coordinate system is required to apply formula (1) to search for the contamination source position (x,y) within the domain 15km x 15km, in which the sensors measuring concentration of the dispersed substance are located (Figure 1).

To summarize, in this paper the scanned model parameter space is $M = (x, y, Q, z_1, z_2)$, where *x* and *y* are spatial locations of the release within the domain, Q release rate, and z_1, z_2 are stochastic terms in the turbulent diffusion parameterization defined in (2). To restrict the space of the searched parameters the height of the searched release source was fixed during the reconstruction at the sensors height i.e. 20 meters (which differs from the assumed during generation of the synthetic data by 30 meters). The plume rise (h) was estimated to be equal 1 meter.

3. Genetic Algorithm concept

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 meets the real data. In this context, we can say that the problem can be seen as an optimization problem for which GA can be applied.

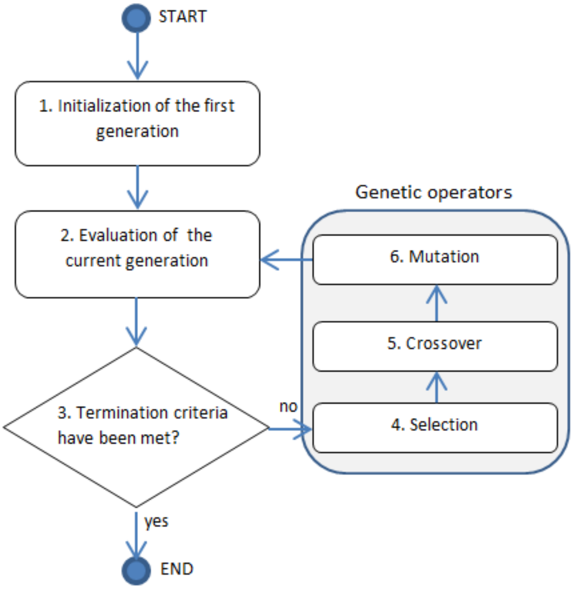
START
1. Initialization of the first generation
2. Evaluation of the current generation
3. Termination criteria have been met?
no
yes
END
Genetic operators
6. Mutation
5. Crossover
4. Selection

Figure 2. Flowchart of the GA

Figure 2 presents the concept of the GA (e.g. Goldberg, 2006). The algorithm starts with defining the initial population. The population is composed of the predefined number of chromosomes, $P(t) = c_1^t, \cdots, c_n^t$ in the generation *t.* The chromosomes in the initial population were randomly drawn from the admissible set of values explicitly defined by the space of explored parameters. The chromosome is configured as a binary value representing the real value of searched parameters. The quality of each chromosome in the current population is evaluated based on the cost/objective function. Various objective functions can be applied; its form depends upon the problem being solved. The 'improvement' of the current population can be done by applying the genetic operators.

The information on the quality of each population chromosome is used to perform a selection. Then the crossover is implemented. As a result, the pairs of parents in the current population are replaced by their children. Children are created by blending of the parents' bits at the randomly chosen crossover point. The crossover probability determines the number of crossovers that occurs within the population. Subsequently, the current population is mutated by changing some chromosome features. Possibilities of changing chromosome individual bits allow the algorithm to search for the entire solution space and not to converge to local extremes. The mutation probability determines the number of going on mutations. After performing the selection, crossover and mutation the new generation (*t+1*), being subject to the new evaluation, is established. After some number of generations the algorithm converges – it is expected that the best chromosome in the population represents a near-optimum (reasonable) solution. The process stops when the termination criterion is fulfilled. The most common termination criterion is a limited number of generations.

3.1. Adjustment of the GA to the problem of source localization

In the problem of the localization of the contamination source GA should find the applied atmospheric dispersion model parameters that fit the model output to the on-line arriving concentrations of given substance. In this paper, the scanned parameters space *M* is five-dimensional i.e. $M \equiv \{x, y, Q, z1, z2\}$.

Correspondingly, each population chromosome stores the following information:

- x, y– coordinates of contamination source in meters,
- Q – strength of the release in grams per second,
- z1, z2 – turbulent dispersion coefficients.

We assume that initially we have no *a priori* information about the parameters values. Accordingly, for each dispersion model parameter we randomize a real number draw from the predefined interval with the use of the uniform distribution. The next step is to encode all real numbers to a binary form. One of the most efficient ways is to perform particular conversion to an integer number and then to the binary number.

Conversion of the real number to an integer number.

Let consider the real number *xReal* from the interval <A, B> with the required precision *P*. The conversion of *x Real* to *xInt* can be achieved by:

1. Finding the lowest value for parameter l that fulfills the following inequality:

$$2^l - 1 \geq \frac{B-A}{P} + 1.$$

Right side of inequality stands for a number of all possible real numbers from the interval <*A, B*> with precision *P*.

2. Flattening the entire interval to such form that its boundaries will be between <0, 1>, by formula:

$$flattenReal = \frac{xReal - A}{B - A}$$
$$xInt = round[flattenReal * (2^l - 1)].$$

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

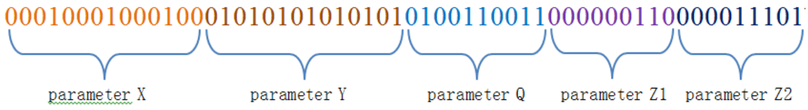

Figure 3. Example of the chromosome representing the searched model parameters

3.2. Evaluation of the population quality

The population evaluation is done with the use of the objective function reflecting the quality of population's chromosomes. This function compares the concentrations predicted by the model and registered at the sensor locations as:

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

where *f* is the objective function, C_i^M are the concentrations predicted by the forward atmospheric dispersion Gaussian plume model 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 (Monache et al., 2008) chosen accordingly to expected errors in the observations for given sensors, assumed here equal to 0.2. It is evident that the greater is the objective function value for the given model setup *M*, the better is estimated the set of searched parameters *M*.

3.3. Genetic algorithm operators – 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, all mentioned methods were tested. The best results were achieved with a selection based on hard tournament, which can be expressed by following pseudocode:

```
FOR i=1 to N LOOP
        FOR j=1 to TS LOOP
                tournamentGroup(j)=drawChromosomeFromPopulation();
        END LOOP
        sortTournamentGroupByObjectiveFunction();
        newPopulation(i) = getBestTournamentChromosome();
END LOOP
```

where N is the size of population and TS is tournament size, equal 2 in this paper. As the result of the tournament from each pair of the selected chromosomes, one with a better objective function value passes to the next generation.

3.4. Genetic algorithm operators - crossover

There can be distinguished many methods of dealing with GA cross over e.g. single point crossover, multi-point crossover, uniform crossover, arithmetic crossover. For a given problem, the best results were achieved 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. The 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. Pseudocode can be expressed in the following way:

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

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

      xCrossoverPoint = drawNumberFrom0ToParameterXLength();
      yCrossoverPoint= drawNumberFrom0ToParameterYLength();
      qCrossoverPoint= drawNumberFrom0ToParameterQLength();
      z1CrossoverPoint= drawNumberFrom0ToParameterZ1Length();
      z2CrossoverPoint= 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,crossoverPoint)+

      tmpXBin2(crossoverPoint+1);
      firstChildY  =tmpYBin1(0,crossoverPoint)+

      tmpYBin2(crossoverPoint+1);
      firstChildQ  =tmpQBin1(0,crossoverPoint)+
        tmpQBin2(crossoverPoint+1);
        firstChildZ1 = tmpZ1Bin1(0,crossoverPoint)+

        tmpZ1Bin2(crossoverPoint+1);
        firstChildZ2 = tmpZ2Bin1(0,crossoverPoint)+

        tmpZ2Bin2(crossoverPoint+1);

        secondChildX= tmpXBin2(0,crossoverPoint)+

        tmpXBin1(crossoverPoint+1);
        secondChildY= tmpYBin2(0,crossoverPoint)+

        tmpYBin1(crossoverPoint+1);
        secondChildQ= tmpQBin2(0,crossoverPoint)+

        tmpQBin1(crossoverPoint+1);
        secondChildZ1= tmpZ1Bin2(0,crossoverPoint)+

        tmpZ1Bin1(crossoverPoint+1);
        secondChildZ2= tmpZ2Bin2(0,crossoverPoint)+

        tmpZ2Bin1(crossoverPoint+1);

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

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

where N is the size of population and CP is a crossover probability.

3.5. Genetic algorithm operators – mutation

The latter applied genetic operator is mutation. The most frequently are used uniform mutation and non-uniform mutation. For a given problem, the best results were achieved with uniform mutation in which all chromosome bits are mutated with the mutation probability MP. Pseudocode can be expressed in the following way:

```
FOR i=1 to N LOOP
        FOR j=1 to L LOOP
                IF drawNumberFrom0To1() <= MP
                        currentPopulation(i).swapBitValue(j);
                END IF
        END LOOP
END LOOP
```

where N is the size of population, L is the length of chromosome binary representation and MP is a mutation probability.

4. Numerical results

Described GA was implemented in the MatLab environment and tested based on the synthetic data outlined in section 2. Various setups of the sensors distribution within the considered domain were tested to confirm the correctness of the applied methodology. However, for clarity results for one configuration are presented. The initial testing of the algorithms efficiency allowed to determine the adequate size of the population to be equal 150 and the number of generations to 50.

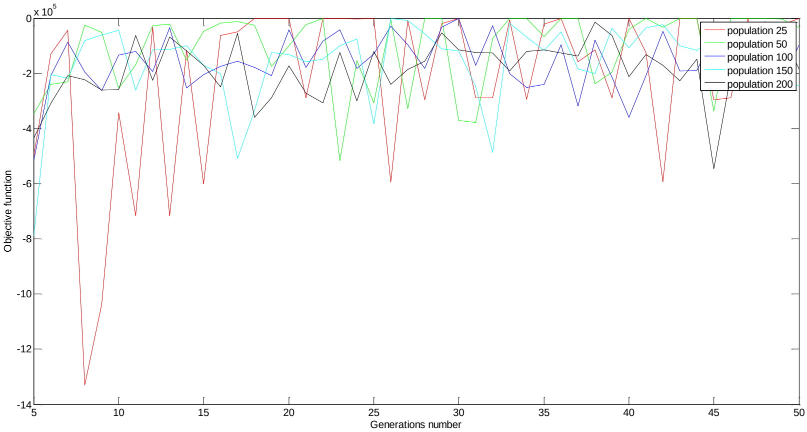

Figure 4. The average values of the objective function obtained during the 30 algorithm runs for 25, 50, 100, 150 and 200 chromosomes in the population

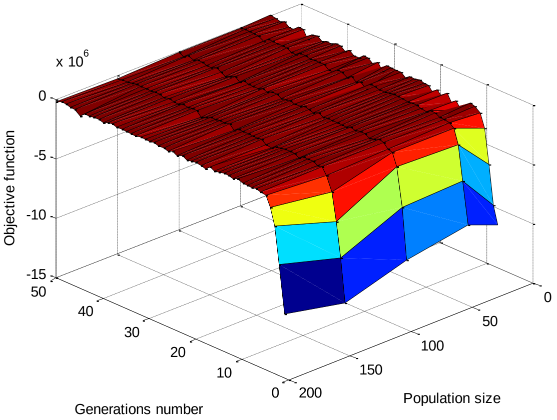

Figure 5. Influence of the population size and number of generations on the value of the objective function. The value of the objective function was averaged over 30 runs

Figure 4 presents the increasing of the objective function value with the growing number of generations. One can also see that the variance of the objective function value decreases with increasing population size. However, there is no distinct difference between the population size 150 and 200. Consequently, in the problem solution we took the population size equal to 150 chromosomes. Figure 5 presents how the objective function value changes with respect to the population size and the generation number. It is visible that for the chosen configuration the profile of the objective function is acceptable, i.e. increasing the population size from 150 to 200 does not improve the objective function value in the 50th generation significantly.

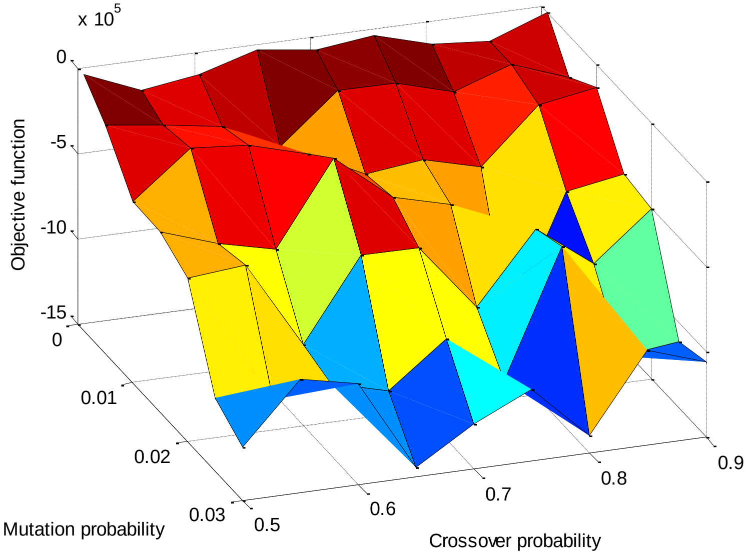

Figure 6. Influence of the mutation probability and crossover probability on the objective function value. The value of the objective function is averaged over 30 runs

Influence of the various crossover and mutation probabilities on the objective function value was also tested. Figure 6 presents the averaged over the population (150 chromosomes) objective function value for 50th generation. Figure 6 illustrates that the objective function value increases with the decreasing of the mutation probability. The impact of the crossover probability on the objective function value is not apparent. However, performed tests endorsed that in the presented optimization problem effective is to apply mutation with the probability MP = 0.005 and crossover with the probability CP = 0.7.

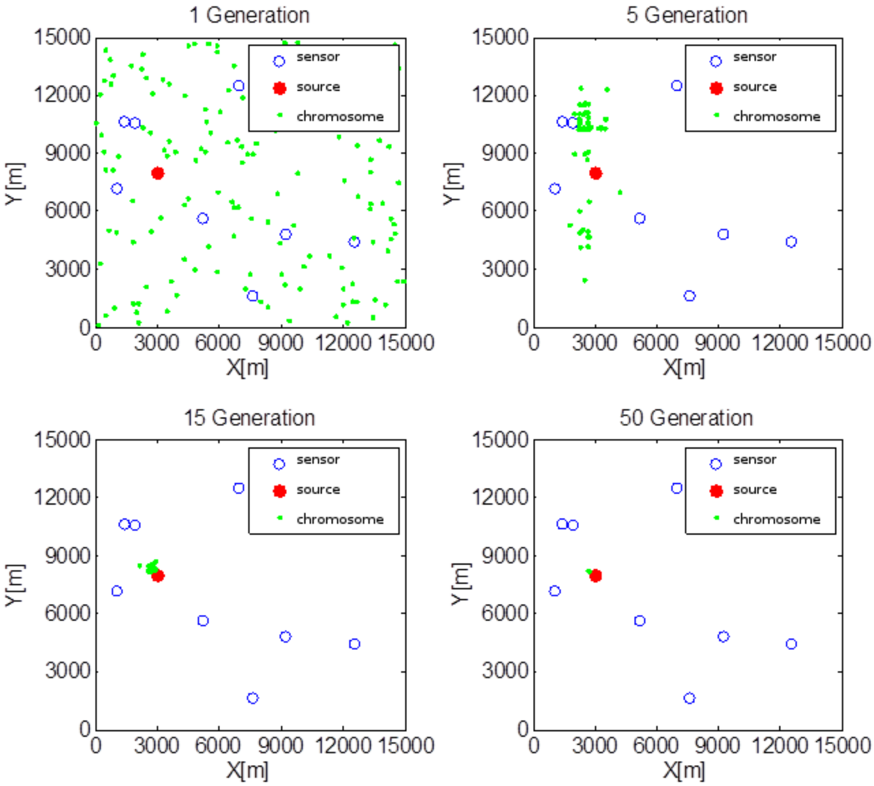

Figure 7. Distribution of the x and y coordinates estimate while the GA runs for the 1, 5, 15 and 50th generation (for optimal GA setup MP = 0.005; CP = 0.7)

Based on the undertaken analysis the following GA configuration was selected as appropriate:

- Number of generations = 50;
- Size of population N = 150;
- Selection based on hard tournament of size 2;
- Multi-point crossover with probability CP = 0.7, with 5 crossover points (5 is the number of searched parameters);
- Uniform mutation with probability MP = 0.005.

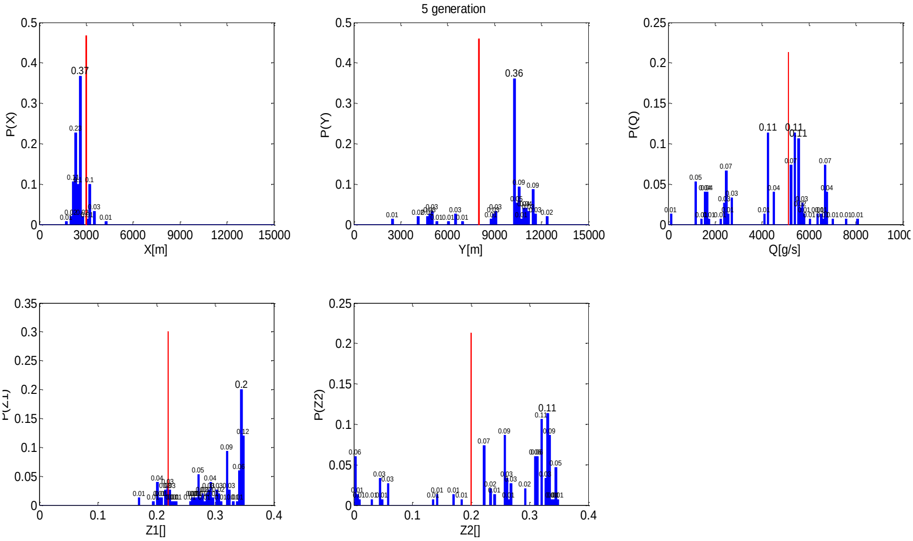

Figure 8. Probability distributions of the model parameters M for the 5th generation (for optimal GA setup with MP = 0.005 and CP = 0.7). The red vertical line represents the target value

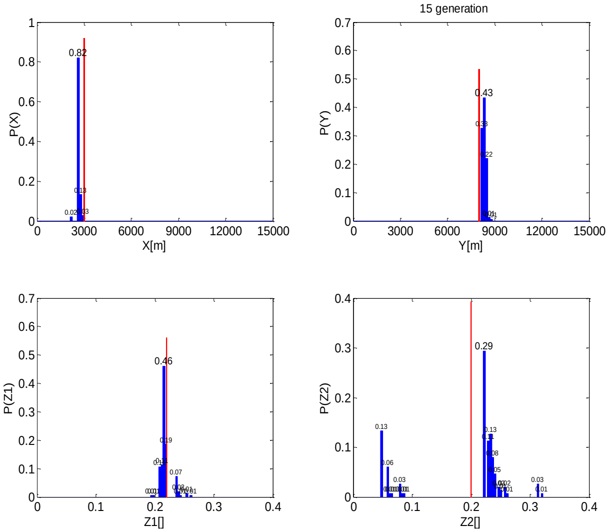

Figure 9. Probability distributions of the model parameters M for the 15th generation (for optimal GA setup with MP = 0.005 and CP = 0.7). The red vertical line represents the target value

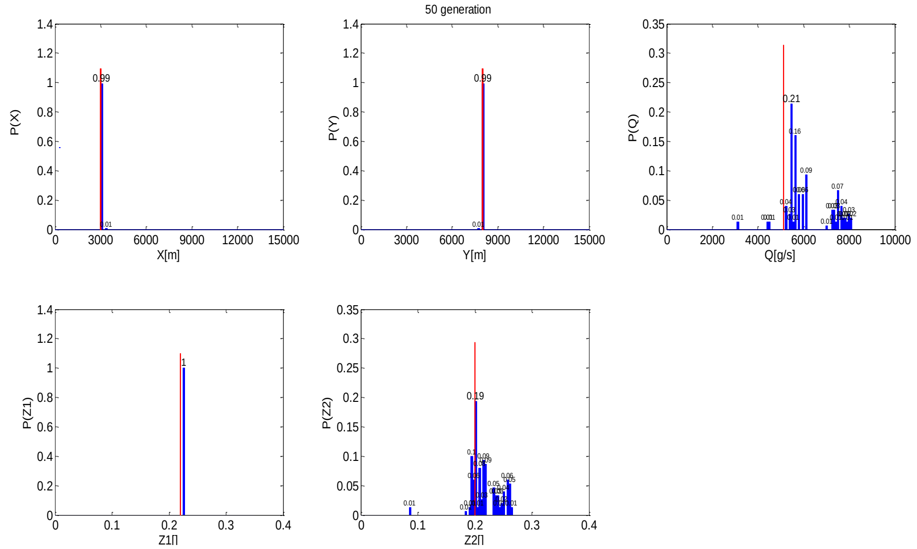

Figure 10. Probability distributions of the model parameters M for the 50th generation (for optimal GA setup with MP = 0.005 and CP = 0.7). The red vertical line represents the target value

Results obtained from the recommended GA configuration are presented in Figures 7, 8, 9 and 10. Figure 7 illustrates the distribution of the estimated by the GA contamination source coordinates *x* and *y* in subsequent generations. It is seen for the 1st generation that the chromosomes are equally distributed within the scanned domain. However, the used genetic operators improve the population quality in subsequent generations and chromosomes gradually focus nearby the actual source location. Finally, for 50thgeneration the contamination source location determined by the GA reach the target position. The development of the chromosomes for subsequent generations for all examined parameters is clearly seen from Figures 8, 9 and 10 where the probability distributions of all five searched parameters are shown. The presented probability distributions were calculated based on the distribution of chromosomes in given generation. The red vertical line marks the target value. Figure 8 shows the probability distribution of searched parameters for the 5th generation. It is seen that, at this stage none of the parameters target value was reached. This is changing with the growth of the population quality by applying genetic operators. Then, for the 15th generation (Figure 9) the near real value of parameter *x* is reached with probability *P(x) = 0.37*; parameter *y* with *P(y) = 0.43* and parameter *z1* with probability *P(z1)= 0.46*. Still, the target value for *Q* and *z2* parameter is not marked as the most likely. Finally, for the 50th generation (Figure 10) as the most probable values of all searched parameters GA marks values close to the target value i.e.:

P(x=3075±75)=0.99, while target value is x=3000;
P(y=8025±75)=0.99, while target value is y=8000;
P(Q=5480±40)=0.21, while target value is Q=5000; *P(z1=0.22575±0.00175)=1,* while target value is z1=0.22*; P(z2=0.20125±0.00175)=0.19,* while target value is z2=0.2.

As the estimated parameter value, we provide the central value of the histogram bar with the highest probability and as the error the half of the bar width. The target parameters value lies within the intervals pointed by the GA as the most probable. Only for the release rate the algorithm overestimated Q by ~440 g/s. However, considering the dispersion of the substance over the domain 225 km^2 and taking into account that the source height was fixed 30 m below its real location, this difference is not physically significant.

The profile of the objective function value during the subsequent generations presents Figure 11.

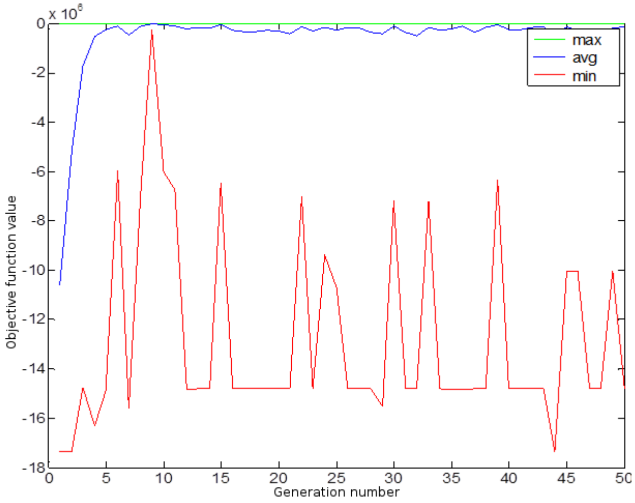

Figure 11. The objective function value for the subsequent generations (for optimal GA setup with MP = 0.005 and CP = 0.7)

For the comparison Figure 12 presents the probability distributions of all parameters in 50th generation derived from not optimal GA configuration in which the mutation probability was increased to MP = 0.03 and crossover probability decreased to CP = 0.5. It is obvious that the results are worse than obtained for the preferred parameters (i.e. MP = 0.005; CP = 0.7). Yet, although the target values of *Q* and *z2* were not marked as the most probable, the most important (in practical application) the contamination source coordinates *x* and *y* relatively close to the target value were marked as the most probable i.e. *P(x = 3525±75)=0.14* and *P(y = 7725±75) = 0.24.*

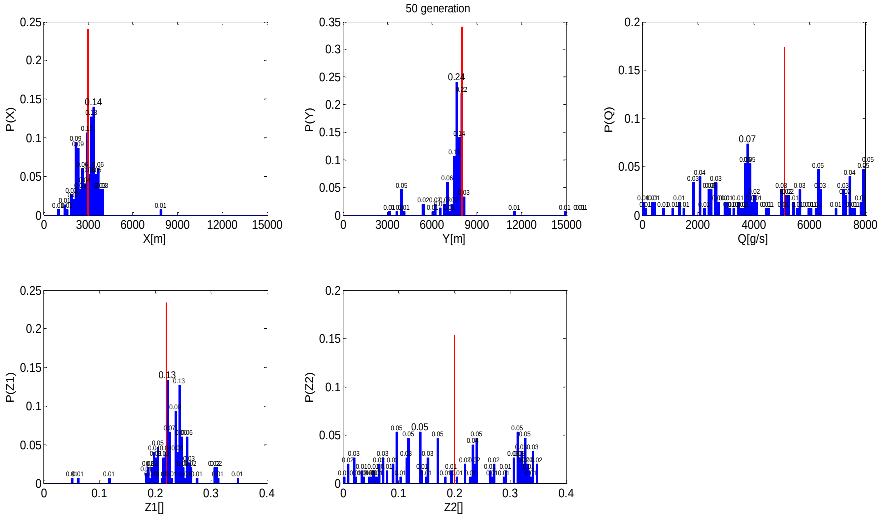

Figure 12. Probability distributions of the models parameters M for the 50th generation (for NOT optimal GA setup with MP = 0.03 and CP = 0.5). The red vertical line represents the target value

5. Conclusion

We can conclude that the applied GA configuration effectively localized the contamination source parameters based on the sparse point concentrations data. The presented algorithm successfully provided the solution to the stated inverse problem i.e. having the downwind concentration measurements and knowledge of the wind field; the algorithm found the most probable location of the source and its strength. The presented probability distributions of all searched parameters encoded in the population's chromosomes allow reflecting the level of confidence in the obtained results. The probabilistic aspect of the solution optimally combines a probable answer to the uncertainties of the available concentrations data driving the calculations.

Acknowledgement

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.

References

1. Allen C.T, Haupt S.E., Source Characterization with a Genetic Algorithm–Coupled Dispersion–Backward Model Incorporating SCIPUFF, Department of Meteorology, The Pennsylvania State University, 2006.
2. Bagchi T.P., *Multiobjective Scheduling by Genetic Algorithms,* 1999.
3. Borysiewicz M., Wawrzynczak A., Kopka P., Stochastic algorithm for estimation of the model's unknown parameters via Bayesian inference, *Proceedings of the Federated Conference on Computer Science and Information Systems*, 501-508, IEEE Press, (2012a).

4. Borysiewicz M., Wawrzynczak A., Kopka P., 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, (2012b).
5. Chwee K., *Switching Control Systems and Their Design Automation via Genetic Algorithms,* PhD Thesis, University of Glasgow, 1995.
6. Eiben A.E., Hinterding R. and Michalewicz Z., 'Parameter Control in Evolutionary Algorithms', *IEEE Transactions on Evolutionary Computation*, Vol. 3, No. 2, 1999.
7. Gifford F.A. Jr., Atmospheric dispersion calculation using generalized Gaussian Plum model, *Nuclear Safety*, 1960, 2(2): 56-59, 67-68.
8. Goldberg D.E., 'Genetic Algorithms in Search, Optimization and Machine Learning', Addison Wesley Longman, London, 2006.
9. Goodall R.M., Michail K., Whidborne J.F. Zolotas A.C, *Optimised Configuration of Sensing Elements for Control and Fault Tolerance Applied to an Electro-Magnetic Suspension*, PhD Thesis, Loughborough University, UK, 2009
10. Holland J.H., "Adaptation in Natural and Artificial Systems", 2nd Edn. Cambridge, MIT Press, 1992.
11. Johannesson G. et al., 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, 2005, 73–80.
12. Johannesson G., Hanley W., and Nitao J., *Dynamic Bayesian models via Monte Carlo - An introduction with examples*, Lawrence Livermore National Laboratory Tech. Rep., 2004, 53.
13. Keats A., E. Yee, and F.S. Lien, Bayesian inference for source determination with applications to a complex urban environment. *Atmos. Environ*., 41, 2007, 465–479.
14. Monache L.D. et al., Bayesian Inference and Markov Chain Monte Carlo Sampling to Reconstruct a Contaminant Source on a Continental Scale, *Journal of Applied Meteorology And Climatology*, 47, 2008, 2600-2613.
15. Pasquill F., The estimate of the dispersion of windborne material, *MeteorolMag*., 90, 1063, 1961, 33-49.
16. Pudykiewicz, J.A., Application of adjoint tracer transport equations for evaluating source parameters. *Atmos. Environ*., 32, 1998, 3039-3050.
17. 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.
18. Saremi A., Mekkawy T. Y. E. and Wang G.G., "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, 2007, pp. 206-219.
19. Turner D. Bruce, *Workbook of Atmospheric Dispersion Estimates*, Lewis Publishers, USA, 1994.
20. Wawrzynczak A., Kopka P., Borysiewicz M., 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, 407-417, 2014, doi:10.1007/978-3-642-55195-6_38