Intelligent Decision Support for Aerial Spray Deposition Management

W. D. Potter, W. Bi,
Artificial Intelligence Center
University of Georgia,
Athens, Georgia USA
potter@cs.uga.edu

D. Twardus, H. Thistle, M. J. Twery, J. Ghent,
USDA Forest Service

M. Teske
Continuum Dynamics
Princeton, NJ

Abstract

The AGDISP Aerial Spray Simulation Model is used to predict the deposition of spray material released from an aircraft. The prediction is based on a well-defined set of input parameter values (e.g., release height, and droplet size) as well as constant data (e.g., aircraft and nozzle type). But, for a given deposition, what are the optimal parameter values? This problem is considered to be a parametric design problem or more generally a configuration problem. Attempting to optimize a configuration based on some set of constraints is known to be extremely difficult (NP-Hard). We use the popular Genetic Algorithm to heuristically search for an optimal or near-optimal set of input parameters needed to achieve a certain aerial spray deposition. Having this knowledge can benefit forest managers substantially, especially regarding such issues as cost, environmental safety, and forest treatment accuracy.

Introduction

Determining the parameter value settings to use as input to the AGDISP Aerial Spray Simulation Model [Bila89] in order to produce a desired spray material deposition is considered an instance of a parametric design problem [Davi91]. Parametric design is a specialization or subtype of the more generic design problem. Typically, when working on a design problem, the solution representation is a set of instructions or components for achieving the design goals. This representation can also be called a configuration, especially if the elements comprising the configuration are predefined. For the parametric design problem we are dealing with, these elements correspond to the AGDISP simulation input parameters. Each parameter has its own domain and range of values. If we arrange the parameters in a one-dimensional array or vector, and select some value for each parameter from that parameter's range then we would have an input parameter configuration. Using this configuration (set of values) as input to the AGDISP simulation model would yield a prediction of the spray deposition.

For this type of problem, the total number of possible configurations can be extremely large. For example, we can calculate this value simply by multiplying together the number of possible values for all of the parameters. That is, if there were twelve parameters, and each parameter had a range of twelve values then the total number of configurations would be 12^{12}. This is indeed a very large number. Now, if we wanted to find the best configuration to achieve a desired spray deposition, then we could enumerate all the possible configurations and run the simulation on each one to see which configuration gave the best deposition. Clearly, this sort of computational task is outside the scope of current computing technology. The configuration problem suffers from what is called combinatorial explosion, that is, as the number of elements increases (e.g., add more parameters), the number of possible configurations also increases but at an exponential rate. See the discussion by Mittal and Frayman in [Mitt89] for more on generic configuration tasks and their complexity.

One method we can use to reduce the computational burden of finding a particular configuration is a heuristic search technique. Heuristic search techniques have been shown to be effective techniques for finding acceptable solutions to problems with very large solution spaces (total number of possible solutions or configurations). The major advantage of a heuristic approach is its speed. The major disadvantage is that there is no guarantee that the heuristic search will find the best solution or configuration. However, typically the solution found is of high enough quality that the trade-off is well worth it. In other words, we may not find the best solution but the solution we do find is typically good enough for our purpose and we found it very quickly. The heuristic search technique we use is the Genetic Algorithm. In the following sections we discuss in more detail the configuration problem, the genetic algorithm, the aerial spray deposition problem, and our approach to the problem along with some preliminary results.

Previous Work

In the development of a good heuristic approach, two methods or knowledge-based system approaches are available to us. These are the rule-based (experiential) approach using typical if-then rules, and the functional (deep or associative) approach based on knowledge about the structure and behavior of a system and its components (see [Chan83, Chan91] for more on the two general approaches). The functional approach follows the "reasoning from first principles" paradigm. This is quite different from the rule-based approach driven by "rules of thumb". Regardless of the approach, the major point of emphasis is the heuristic synthesis of a satisfactory solution or, in our case, a satisfactory configuration. Our Spray Advisor Genetic Algorithm (SAGA) approach, however, could be considered a combination of the rule-based and functional paradigms (although we do not have a typical collection of if-then rules, expert knowledge is incorporated into SAGA in the form of the sophisticated AGDISP simulation model). For a configuration application, synthesis means incorporating a given set of parameter settings and a set of constraints associated with the parameters into a configuration that satisfies the deposition goals and constraints. Synthesis can be thought of as the design of a solution.

Probably the most famous expert system to be developed for design applications is R1 (XCON) which is used to configure computer systems from customer specifications [Bach84, McDe81]. An early example of an engineering design system for configuring networks using heuristics is DESIGNET, developed by Bolt, Beranek, and Newman in the early 1980's [Mant86]. DESIGNET is a rule based design aid that focuses on an iterative user interface approach to configuration based on the process a decision maker goes through during the design process.

Our own experience with configuration deals with designing battlefield communication networks to support specific missions. Our system, called IDA-NET, configures a “shopping list” of communication equipment indicating type of equipment and number of components [Pott92]. The “shopping list” represents the required amount of equipment to support a particular mission. The goal is to minimize the number and types of components yet still satisfy a set of constraints associated with the mission, the equipment connectivity, and the available components in inventory.

Aerial Spray Models

For many years, computer simulation models for predicting what happens to spray material released from aircraft have been a major research interest of the USDA Forest Service [Teske98b]. The Forest Service Cramer-Barry-Grim (FSCBG) aerial spray model [Teske89, Teske93a] and the Agricultural Dispersal (AGDISP) model [Bila89, Teske98a] are examples of this research. AGDISP simulates the effects of aircraft movement and wake on material released from the aircraft. The model predicts the behavior of spray material droplet movement when sprayed from an airplane or helicopter. FSCBG predicts the dispersion of the spray material and the deposition of the material (that is, how much material settles on the ground and where). Both models analyze the movement of the spray material above the forest canopy, the movement among the trees, and the amount of material that actually reaches the ground. Getting the spray material to reach the proper location depends on many factors. These factors include: (1) the altitude of the aircraft when the material is released, (2) the speed of the aircraft, (3) whether the aircraft is an airplane or a helicopter, (4) the type of boom and nozzle system used to discharge the spray material, (5) the swath width of each pass of the aircraft, (6) the type and density of the forest, (7) wind speed and direction, (8) relative humidity, and (9) spray material characteristics. Determining the optimal set of factors in order to provide accurate (getting the spray material exactly where it should be), and inexpensive (using the exact amount of material; not too much and not too little) spraying is the goal of our research. We are currently investigating the use of a genetic algorithm to determine the parameters.

The Genetic Algorithm

Genetic Algorithms [Davi91, Gold89, Holl75] are heuristic search routines that are guided by a model of Darwin's theory of natural selection or the survival of the fittest. Here the fittest means the most highly ranked solution in a large solution space. The basic idea behind the genetic search strategy is to generate solutions that converge on the global maximum (i.e., the best solution in the search space) regardless of the "terrain" of the search space. A typical terrain might resemble the Great Smoky Mountains with many peaks and valleys, an area that is relatively flat, and a highest peak (Clingman's Dome). One characteristic of genetic algorithms is that they are relatively unaffected by hill-climbing or being misled by some local maximum such as ascending Mt. LeConte and assuming that you are on the highest peak in the Smokies since other nearby peaks appear lower, depending on visibility. Likewise, with genetic algorithms the key to finding the global maximum lies in the ability to evaluate and compare possible optimal solutions.

The basic operations involved in a genetic algorithm (GA) are: 1) mate selection, 2) crossover, and 3) mutation. Typically, the major data structure is a binary string representing the possible solutions. In GA terms, a bit string corresponds to an individual, and a set of individuals is called a population. The fitness or strength of an individual is computed using some objective or fitness function, and is used to compare an individual with other individuals in the same population. During mate selection, parent strings are stochastically selected, according to their fitness, from the current population. Then, parent strings are "mated" via crossover to produce offspring for the next generation. Fitter parents contribute more offspring to the next generation than weaker parents because they have a higher probability of being selected for mating. This is the step that models the process of natural selection in nature.

Crossover, the second operation, determines the characteristics of a child or next generation individual. In nature, children inherit good as well as bad features of their parents in varying degrees of dominance. Crossover performs this same function in a GA. One of the simplest crossover approaches is to split each parent string at the same randomly chosen location and swap their tail sections. This ensures a certain amount of inheritance and ideally, the good/strong features will dominate the children. The inheritance of features that produce stronger children throughout the generations is the source of the GA's ability to converge on the global maximum in a relatively short time.

The last basic operation is called mutation. Mutation is that extremely rare "glitch" in the inheritance mechanism that introduces or modifies some feature with unpredictable consequences. Mutation occurs in a GA immediately after the creation of a next generation individual yet before the next generation has become static. Once the new generation becomes static, we move forward in order for it to become the new current generation. Ideally, mutants would contain some useful features that may have been inadvertently lost in earlier generations.

The simple genetic algorithm described in Goldberg follows these three basic steps [Gold89]. Additional operations and modifications are described as well. One major modification to the simple crossover approach, called two-point crossover, has been shown to be an easily implemented and effective alternate to simple crossover. With two-point crossover, an individual bit string is viewed as a ring and sections of parents are interchanged. This is like cutting equal sized sections from two donuts and swapping the sections to form a new (more appetizing) pair of snacks. Another effective crossover approach is the "greedy" approach described in [Liep90]. They report encouraging results using the "greedy" approach for general set covering problems. Other variations and improvements of the GA operators can be found in [Davi91, Jog89, Pott90, Pott91].

SAGA

Figure 1 (on the last page) shows the architecture for our spray advisor GA. The GA sends a set of AGDISP parameters to the AGDISP simulation model. The AGDISP model calculates and sends back the deposition for each parameter set. Based on the fitness function values mapped from deposition and the coefficient of variation (COV), the GA evolves an improved set of parameters and sends it to AGDISP. This process is repeated from generation to generation for each individual in the population until a satisfactory deposition is found. The corresponding parameter set is returned as the proposed set-up to achieve the desired deposition. Currently, we focus on twelve specific parameters. The twelve parameters used in this study are listed in Table 1. Other less important or more static parameters are kept constant during our experiments. However, they can become part of the variable parameter set (i.e., we can easily include additional parameters to the parameter set we are searching for) by specifying them at the beginning of each SAGA run.

Table 1. SAGA Parameters and Their Ranges

PARAMETER	LOWER	UPPER
Release Height (m)	1	100
Wind Speed (m/s)	0.5	10.0
Drop Size Distribution (mm)	100	200
Wind Direction (deg)	-360	360
Number of Nozzles	1	60
Total Flow Rate (gal/min)	0.1	1000.0
Volatile Fraction	0.0	1.0
Flight Speed (m/s)	10	200
Dry Bulb Temperature (degC)	1.0	51.67
Relative Humidity (%)	5.0	100.0
Number of Swaths	1	20
Width of Swath (m)	5	300

We use AGDISP DOS Version 7.0 for the AGDISP computation engine in SAGA. AGDISP DOS Version 7.0 has the advantage of reading its input parameters from ASCII data files, displaying output information to the screen as the run proceeds, and writing deposition output to a text file. We developed the methodology described below to make full use of this feature in order to establish the interconnections between our GA and the AGDISP simulation model, which is one of the most important facets of this work.

Our approach to connecting the GA with AGDISP is as follows. First, we specify the GA characteristics in the GA input file (saga.inp). We altered the simple GA in order to generate a text file containing the twelve key parameters and all other necessary AGDISP parameters in the format of the input file for AGDISP 7.0. This file is named 'agdisp.inp'. Then AGDISP is initialized by the GA main routine to compute the deposition. Since the GA and AGDISP are two separate programs that run as separate processes, the GA program halts until AGDISP generates and saves the deposition results in an output file, 'agdisp.dep'. This file contains two columns of data, one for downwind distance and the other for deposition. Then the GA continues execution. It reads in the deposition values from 'agdisp.dep'. The COV of depositions would be computed (we have it set to a constant value in the current experiments) and combined with the deposition to map the objective function to form the fitness function. Our long-term goal is to maximize the deposition and minimize the COV. Based on the fitness value, the GA evolves an improved set of parameters to send back to AGDISP. This process is repeated for each individual in every generation until a satisfactory deposition and acceptable parameter set are found.

Modified Simple Genetic Algorithm Used in SAGA

The Genetic Algorithm driver in this study originated from the Simple Genetic Algorithm (SGA) described by Goldberg [Gold89]. We are using a shareware Fortran version of the SGA implemented by David L. Carroll [http://www.staff.uiuc.edu/~carroll/ga.html]. The GA initializes the first population with individuals generated at random. An individual corresponds to a set of AGDISP parameters. We use a binary representation for the individuals. The selection scheme is tournament selection with a shuffling technique for choosing random pairs for mating. We have the option of using jump mutation or creep mutation, and the option for single-point or uniform crossover. Other features are included such as niching, using a micro-population, and variable offspring production. We added roulette wheel selection as another selection scheme option, two-point crossover, intermediate output file generation for AGDISP input, and changed the standard I/O formats to meet our project requirements.

The GA parameters currently in use are: population size between 40 and 100, generations between 50 and 200, crossover probability between 0.6 and 0.9, jump mutation probability set between 0.005 and 0.05, and creep mutation probability set between 0.002 and 0.05.

Results and Discussion

In our initial testing stages, we focused on the determination of (hopefully) optimal spray parameter settings. Some preliminary results are shown in Table 2 and are based on the GA parameters specified earlier. Keep in mind that we are dealing with two sets of parameters: one set for the genetic algorithm which includes population size and crossover probability, and one set for the aerial spray advisor which includes release height and drop size. From the evolution of the fitness values, we can see that the GA is doing a good job of improving the parameter values in order to obtain better depositions. For example, comparing the depositions at the edge of the spray block, we can see that the deposition has improved from 98.34 mg/m^2 in the first generation to 146.53 mg/m^2 after 70 generations.

Table 2. Preliminary Results

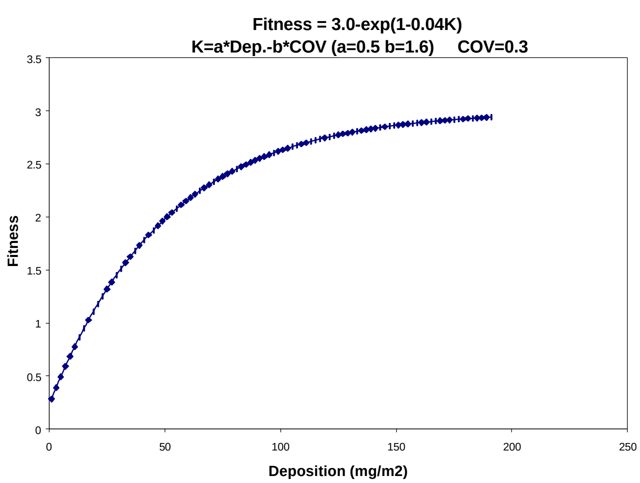
GENERATION	DEPOSITION (mg/m^2)
1	98.34
5	99.46
10	102.56
20	108.25
30	116.84
40	119.25
50	124.29
60	137.58
70	146.53

There are a few simplifications that we embedded during these testing stages such as setting the COV to a constant value of 0.3, and limiting the droplet size range. The primary reason for these simplifications is that it allows us to begin the spray parameter optimization process fairly quickly after setting up the genetic algorithm. The computation of the COV is somewhat tricky within AGDISP. We are in the process of implementing another routine that will determine COV. We will incorporate it into SAGA very soon. The other simplification deals with droplet size distribution. Here we set the range for droplet size to be between 100µm and 200µm. This range is subdivided into ten droplet size categories with an increment of 10µm. Each droplet size category is assigned a mass fraction of 0.1. We are continuing to investigate these issues in order to arrive at a more accurate evolutionary approach to setting these parameters.

We ran numerous experiments to determine which GA parameters seemed to produce the best results. The selection of GA parameters such as population size, number of generations, crossover type and probability, and mutation probability is a key facet of the speed and success of the evolutionary process. These parameters are typically domain dependent. With SAGA, we are, to a certain extent, limited by the runtime of AGDISP. This makes it inconvenient to change the population size and generations freely. The runtime of the AGDISP module typically varies from 5 to 45 seconds for each run depending on the aerial spray parameters. The runtime of the main GA program is negligible compared to the AGDISP runtime. Thus for example if we set the population size to 50 and number of generations to 100, then use an average AGDISP runtime length of 15 seconds, it will take about 20 hours to complete the SAGA run. During our experiments, we usually let SAGA run overnight and collect data the next morning. Therefore, the number of generations was accordingly set to around 50 and the population size was set to between 50 and 100. Table 3 shows some comparisons of the results obtained with different GA population sizes. Similar experiments were run to help determine values for other GA parameters. Our current GA parameter setup includes a population size of 50, between 50 and 100 generations, a crossover probability between 0.8 and 0.9, and a mutation probability between 0.02 and 0.06.

Another key issue in the development of SAGA is the mapping of the deposition and the COV onto the fitness function. Our goal is to maximize the deposition and minimize the COV. That is, get the exact amount of spray material evenly distributed over the spray block. We follow the rule of thumb suggested in [Park82] and set the COV to 0.3. We tested and compared different mapping functions having linear and exponential characteristics, and are currently using the exponential function formulated below.

$$Fitness = 3.0 - \exp(1 - 0.04(a \times Deposition - b \times COV))$$



It should be noted that COV is dependent on swath width in most cases, but in the above formulation, the deposition is less than two orders of magnitude higher than that of COV, therefore we can set COV constant without changing the nature of the mapping.

In addition, some other work we are carrying out is to test the parameter sensitivity of AGDISP. The approach we take is to set one of the twelve SAGA parameters constant and test the impact of this change on the deposition evolution. Release height, wind direction, and wind speed are the three main parameters we have tested thus far. The results are presented in Table 4. As we can see, setting the release height has a large impact on the deposition evolution. Likewise, keeping the wind parameters constant also has a considerable impact on SAGA results. The trend is consistent with the results obtained by Teske and Barry [Teske93b], namely that the input parameters for aerial spray can be ranked in order of importance where release height is more important than any other parameter. The approach they took to measure the relative importance was to change an input variable linearly and measure the corresponding relative sensitivity of the results. Two parameter values, Figure of Merit and Mean Horizontal Position were used to measure the effectiveness of swath width deposition and the level of off-target drift, respectively. Compared to their approach, our tests indicate that we need further technical verification of our results, and additional tweaking of our approach. But the trend indicated by our results coincides with the important roles of these key parameters and roughly their relative importance.

We are currently working to incorporate AGDISP parameter dependencies and practical application considerations (spray knowledge) into a revised fitness measure. Also, a user-friendly interface is being set up to facilitate the use of SAGA. The main interface is designed to enable a user to specify the GA parameters. An additional interface feature will allow the user to specify certain AGDISP parameters other than the primary twelve parameters evolved by SAGA. With these new features incorporated, we expect SAGA to produce more satisfactory results in the near future and therefore become a new instrumental aid to aerial spray applications.

References

[Bach84] Bachant, J., and J. McDermott (1984). “R1 Revisited: Four Years in the Trenches,” in *AI Magazine*, Vol. 5, No. 3.

[Bila89] Bilanin, A. J., M. E. Teske, J. W. Barry and R. B. Ekblad (1989). “AGDISP: The Aircraft Spray Dispersion

Model, Code Development and Experimental Validation". *Transactions of the ASAE* 32(1):327-334.

[Chan83] Chandrasekaran, B., and S. Mittal (1983). "Deep versus Compiled Knowledge Approaches to Diagnostic Problem Solving," in the *International Journal of Man-Machine Studies*, Vol. 19, No. 5, pp. 425-436, November.

[Chan91] Chandrasekaran, B. (1991). "Models Versus Rules, Deep Versus Compiled, Content Versus Form," in *IEEE Expert*, Vol. 6, No. 5, April.

[Davi91] Davis, L., (ed.) (1991). *Handbook of Genetic Algorithms*, Van Nostrand Reinhold, New York.

[Gold89] Goldberg, D.E. (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley Publishing Co.

[Holl75] Holland, J.H. (1975). *Adaptation in Natural and Artificial Systems*, Ann Arbor: The University of Michigan Press.

[Jog89] Jog, P., J.Y. Suh, and D.V. Gucht (1989). "The Effects of Population Size, Heuristic Crossover and Local Improvement on a Genetic Algorithm for the Traveling Salesman Problem," in the *Proceedings of the 3rd International Conference on Genetic Algorithms*, Morgan Kaufmann Publishing, San Mateo, CA.

[Liep90] Liepins, G.E., M.R. Hilliard, J. Richardson, and M. Palmer (1990). "Genetic Algorithm Applications to Set Covering and Traveling Salesman Problems," in *OR/AI: The Integration of Problem Solving Strategies*, (Brown, ed.).

[Mant86] Mantelman, L. (1986). "AI Carves Inroads: Network Design, Testing, and Management," in *Data Communications*, pp. 106-123.

[McDe81] McDermott, J. (1981). "R1: The Formative Years," in *AI Magazine*, Vol. 2, No. 2.

[Mitt89] Mittal, S., and F. Frayman (1989). "Towards a generic model of configuration tasks," in the *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence*, Vol. 2, pp. 1395-1401.

[Park82] Parkin, C.S., and J.C. Wyatt (1982). "The Determination of Flight-Lane Separations for the Aerial Application of Herbicides," in *Crop Protection*, 1 (3), pp. 309-321.

[Pott90] Potter, W.D., J.A. Miller, and O.R. Weyrich (1990). "A Comparison of Methods for Diagnostic Decision Making," in *Expert Systems with Applications: An International Journal*, vol. 1, pp. 425-436.

[Pott91] Potter, W.D., J.A. Miller, B.E. Tonn, R.V. Gandham, and C.N. Lapena (1991). "Improving the Reliability of Heuristic Multiple Fault Diagnosis Via The Environmental Conditioning Operator," in the *International Journal of Applied Intelligence*, vol. 2, pp. 5-23.

[Pott92] Potter, W.D., R. Pitts, P. Gillis, J. Young, and J. Caramadre (1992) "IDA-NET: An Intelligent Decision Aid for Battlefield Communication Network Configuration," in the *Proceedings of the Eighth IEEE Conference on Artificial Intelligence for Applications* (CAIA'92), pp. 247-253.

[Teske89] Teske, M.E., Curbishley, T.B. (1989), "Forest Service Aerial Spray Computer Model FSCBG 4.0 User Manual", C.D.I. Report No. 90-06.

[Teske93a] Teske, M.E., Bowers, J.F., Rafferty, J.E., and Barry, J.W., (1993). "FSCBG: An Aerial Spray Dispersion Model for Predicting the Fate of Released Material Behind Aircraft" in *Environmental Toxicology and Chemistry*, Vol. 12, pp. 453-464.

[Teske93b] Teske, M.E. and J.W. Barry (1993). "Parameter Sensitivity in Aerial Application" in *Transactions of the ASAE,* Vol.36(1), pp. 27-33.

[Teske98a] Teske, M.E. (1998) "AGDISP DOS Version 7.0 User Manual".

[Teske98b] Teske, M. E., H. W. Thistle and B. Eav. (1998a) "New Ways to Predict Aerial Spray Deposition and Drift," in *Journal of Forestry* 96(6):25-31.

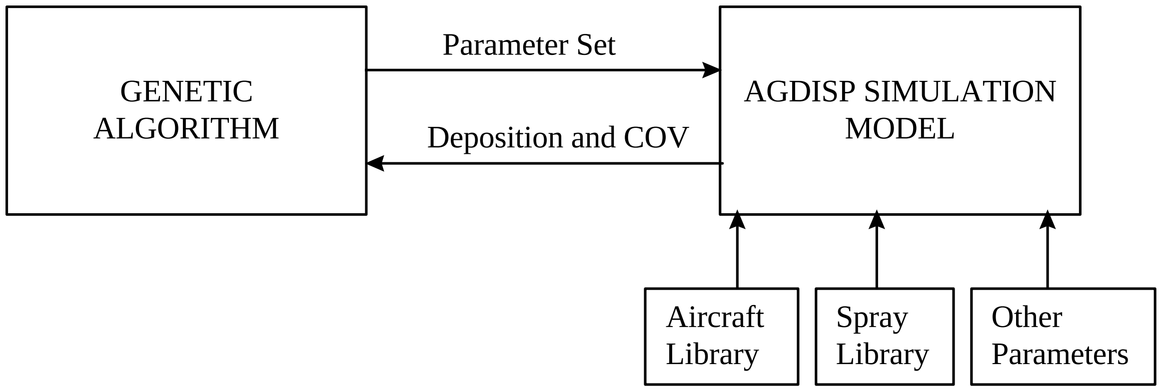


Figure 1. SAGA Architecture.

Table 3. SAGA Results at Different Population Size

GENERATION	DEPOSITION (mg/m^2)	DEPOSITION (mg/m^2)	DEPOSITION (mg/m^2)
	Popsize = 50	Popsize = 40	Popsize = 20
1	98.34	98.34	98.34
20	108.25	107.36	105.42
50	124.29	122.68	116.35

Table 4. Testing of the SAGA Parameter Importance

GENERATION	DEPOSITION (mg/m^2)	DEPOSITION (mg/m^2)	DEPOSITION (mg/m^2)	DEPOSITION (mg/m^2)
		Release Height = 75m	Wind Direction = 150 degree	Wind Speed = 5.0m/s
1	98.34	97.38	96.52	96.82
10	102.56	100.25	100.34	101.25
20	108.25	104.39	103.95	103.49
40	119.25	112.65	115.87	114.58
60	137.58	120.87	125.75	124.68