Nature Inspired Heuristics in Aerial Spray Deposition Management

L. Wu, W.D. Potter, K. Rasheed J. Ghent, D. Twardus, H. Thistle M. Teske
University of Georgia USDA Forest Service Continuum Dynamics

Abstract

AGDISP (Aerial Spray Simulation Model) is used to predict the deposition of spray material released from an aircraft. Determining the optimal input values to AGDISP in order to produce a desired spray material deposition is extremely difficult (NP hard). SAGA, an intelligent optimization method based on the simple genetic algorithm, was developed to solve this problem. In this paper, we apply several nature inspired heuristics to this problem. The first method still uses the genetic algorithm, but changes its type, selection method, crossover and mutation operator. The second method applies a neural network to improve the initial population, crossover and mutation. The third method uses GADO, a general-purpose approach to solving the parametric design problem. The fourth method applies simulated annealing to this problem. Finally, we compare their performance with SAGA and discuss their applications to the aerial spray deposition problem.

Keywords: Genetic Algorithm, Neural Network, Simulated Annealing, Aerial Spray Deposition

To appear (late 2002 or early 2003) in the *Journal of Applied System Sciences*, special issue on "Real Life Applications of Nature Inspired Combinatorial Heuristics", guest editor Dr. Hussein A. Abbass.

Nature Inspired Heuristics in Aerial Spray Deposition Management

L. Wu, W.D. Potter, K. Rasheed J. Ghent, D. Twardus, H. Thistle M. Teske
University of Georgia USDA Forest Service Continuum Dynamics

Abstract

AGDISP (Aerial Spray Simulation Model) is used to predict the deposition of spray material released from an aircraft. Determining the optimal input values to AGDISP in order to produce a desired spray material deposition is extremely difficult (NP hard). SAGA, an intelligent optimization method based on the simple genetic algorithm, was developed to solve this problem. In this paper, we apply several nature inspired heuristics to this problem. The first method still uses the genetic algorithm, but changes its type, selection method, crossover and mutation operator. The second method applies a neural network to improve the initial population, crossover and mutation. The third method uses GADO, a general-purpose approach to solving the parametric design problem. The fourth method applies simulated annealing to this problem. Finally, we compare their performance with SAGA and discuss their applications to the aerial spray deposition problem.

Keywords: Genetic Algorithm, Neural Network, Simulated Annealing, Aerial Spray Deposition

Introduction

Aerial spray pest control has always been an important application in forest management the world over. Computer simulation programs are frequently used to facilitate the spraying practice. AGDISP (Aerial Spray Simulation Model) is one of the main models developed for this purpose by the USDA Forest Service. It simulates the behavior of spray material released from an aircraft, and predicts the spray deposition and drift (of both herbicides and pesticides).

The prediction is based on a well-defined set of input parameter values as well as constant data (Bilanin et al., 1989). There are eleven primary input parameters: aircraft ID number, block width, boom height, boom length, relative humidity, number of nozzles, nonvolatile fraction, swath width, temperature, volume median diameter input and wind speed (the details are in Table 1). In practical spray applications, some of these can be fixed based on the requirements or situation.

Table 1. Spray Parameters and Their Range

Spray Parameters	Lower	Upper
Aircraft ID	1	124
Block Width (m)	50.0	1000.0
Boom Height (m)	8.0	15.0
Boom Length	0.3	1.0
Humidity (%)	15.0	100.0
Nozzles	1	60
Nonvolatile Fraction	0.001	1

Swath Width	0.3	3.0
Temperature (C)	1.0	30.0
VMD Input (μm)	100.0	400.0
Wind Speed (m/s)	0.23	4.47

AGDISP returns three spray output values: VMD (the deposition composed of Volume Median Diameter), drift fraction, and COV (the Coefficient of Variance). VMD is a measure of spray material droplet size. Drift fraction identifies the amount of spray material deposited outside the spray block. COV gives an indication of the uniformity of the deposited spray material. The goal is to minimize the drift fraction, minimize the COV, and minimize the difference between the output VMD and the desired VMD. That is, get the exact amount of spray material evenly distributed over the spray block with the least loss due to evaporation and attrition.

It is a difficult problem to identify ideal spray parameters to achieve a desired deposition, reduce spray material evaporation or drift, and save time and money devoted to the spray process (Teske et al., 1989). The difficult part lies in that there are dozens of spray parameters in spray practice and each of them has many possible values. The total combination of possible spray parameters generates a huge search space. In order to avoid this combinatorial explosion in the parameter space, heuristic search techniques have been applied to get a near-optimal solution.

Previous work

SAGA (Spray Advisor using Genetic Algorithm) was developed, in cooperation with experts from the USDA Forest Service, to heuristically search for a near-optimal set of input parameters needed to achieve a certain aerial spray deposition (Potter et al., 2000). The Simple Genetic Algorithm (SGA) was used for the optimization process (Goldberg, 1989). SAGA sends a set of spray parameters to the AGDISP simulation model. AGDISP calculates and sends back the spray results for each parameter set. Based on the fitness function values mapped from the spray results, the genetic algorithm attempts to evolve an improved set of parameters.

This is a multi-objective optimization problem since there are three output values to optimize. A weighted-sum approach is applied, which assigns weights to each output value and combines them into a single-objective function (Zadeh, 1963). The fitness function suggested by the USDA Forest Service experts is shown below. (VMDCenter is the desired VMD value specified by the user, usually an entomologist, before the run.)

$Fitness = 100\times[50\times(1.0-DriftFraction)+25\times(1-COV)+25\times VMDTerm]$, *where*
$VMDTerm = 1.0-abs(1.0-VMD/VMDCenter)$.

As can be seen, this is a typical parametric design problem, in which the genetic algorithm evolves the spray parameters in a continuous domain, and the simulation engine evaluates the solution. The simulation engine takes 2-5 seconds (on a Pentium III 600 MHZ PC) to evaluate the input parameters for an individual, so the total number of simulations cannot be too high in order to get a solution in a reasonable time (normally SAGA allows 5000 simulations). The population size is set to 100, the crossover rate is 1, and the mutation rate is set to 0.1 initially and decreases with time. Experiments were done on various spray parameter settings. The results were evaluated by the spray application experts and regarded as excellent predictions with high practical importance.

The goal of this paper is to briefly report additional attempts to improve the SAGA results since pest control via aerial spraying is such an important real-world application. For example, environmentalists are concerned with drift and its possible contamination of areas not meant to be sprayed while conservationists and the public are concerned with the severe damage caused by forest pests (e.g., the Gypsy Moth and Spruce Budworm). The forestry industry is also very much concerned with controlling pest damage. In addition, SAGA takes some time to produce its results so attempts were made to shorten the time required to achieve useful results. The following sections describe the various modifications/attempts we have made so far in our aerial spray research.

SAGA2

In the aerial spray deposition problem, the search space is huge and the simulation number is limited, so SAGA has difficulty converging to a global optimum. We developed SAGA2 (Spray Advisor using Genetic Algorithm version 2) from SAGA, and hoped to get better results. SAGA uses a generational genetic algorithm, replacing an entire set of parents by offspring (Grefenstette, 1986). The convergence speed of this scheme is rather slow. Actually, after 5000 simulations (100 individuals × 50 generations) as specified by the program, the population has typically not converged. SAGA2 uses a steady-state genetic algorithm, replacing only the n worst parents by offspring (Davis, 1991). We expected it to speed up the convergence process and reach a better result.

The original SAGA selection method was roulette wheel selection, which determines selection probability for each chromosome proportional to the fitness value (Holland, 1975). In SAGA2, we replaced roulette wheel selection with tournament selection, which randomly chooses a set of chromosomes and picks out the best chromosome for reproduction (Goldberg et al., 1989). According to Goldberg and Deb, tournament selection is robust and less prone to premature convergence (Goldberg et al. 1991).

Instead of using a single type of crossover and mutation operator, in SAGA2 we combined several kinds of crossover and mutation operators and applied them with different probabilities. The three kinds of crossover operators are blend crossover (0.6), direction-based crossover (0.3), and uniform crossover (0.1) (Eshelman et al., 1993, Michalewicz et al., 1994, Syswerda, 1989). The three kinds of mutation operators are greedy mutation (0.1), shrinking window mutation (0.5) and nonuniform mutation (0.4) (Rasheed, 1998, Michalewicz, 1996). By improving the crossover and mutation methods, our goal was to increase the search capability of the genetic algorithm.

The main interface of SAGA2 is shown in Figure 1. The user can press the button “Preset Parameters” to preset certain spray parameters by selecting the ones to preset and then fill in appropriate values. The rest of the parameters are left open to evolution by the genetic algorithm. The interface to preset spray parameters is shown in Figure 2. Depending on the user’s knowledge of the genetic algorithm and the spray application scenario, the user can press the button “Customized Parameters“ to modify a set of genetic algorithm parameters and the value of VMDCenter. The interface to customize SAGA2 parameters is shown in Figure 3.

SAGA2NN

We developed SAGA2NN (Spray Advisor using Genetic Algorithm version 2 with Neural Network) as a variant of SAGA2. SAGA2 uses random numbers to produce the initial population, so the average fitness of the initial population is not high and the individuals are rather diverse. SAGA2NN generates the initial population from a large pool of individuals. For example, if the population size is 100, then the GA randomly produces 2000 individuals in the process of generating the initial population. It uses a neural network to approximate the fitness values, and selects 100 individuals with the highest fitness as its initial population. Then it uses the simulation engine AGDISP to get the accurate fitness values for these 100 individuals. Compared to the time cost of the 100 simulations, the time to compute 2000 individuals using a neural network is negligible, but the average fitness of the initial population will be much higher than that randomly produced. SAGA2NN also applies this method in the process of crossover and mutation. For each crossover and mutation, it actually does twenty crossover and mutation operations, uses the neural network to approximate the results' fitness, and selects the one with the highest fitness as the candidate. This idea comes from Rasheed et al. (2000). By improving the initial population, crossover, and mutation, we expected the genetic algorithm to require fewer simulations in order to converge and to converge to better results.

We ran 26000 simulations to collect the data, used 20000 of these as the training and testing set, and 6000 as the production set. The input variables are the eleven spray parameters (randomly generated, different in each simulation), and the output variables are drift fraction, COV and VMD, which are computed by the simulation engine AGDISP. It took 24 hours to collect these data (on a Pentium III 600 MHZ PC). The cost is worthwhile since the genetic algorithm will run many times for different spray parameter settings in the future (every run needs 4 to 6 hours). We developed a program to train the neural network. The learning rule we used was backpropagation with momentum (Gallant, 1993). The sigmoid function was the logistic function: $g(u) = 1/(1+1/e^u)$. The main interface of the program is shown in Figure 4.

We used 75% of the data as the training set, 0.1 as the learning rate, and 0.8 as the momentum. The neural network tries to minimize the MSE (median squared error) of the three output values in the production set (in a file NNProduction.txt). It turns out that the number of hidden nodes has a significant influence on the performance of the neural network. It was found that 50 hidden nodes produced the best result (the details are in Table 2).

Table 2. Influence of hidden nodes on neural network

Hidden Nodes	20	30	40	50	60
MSE of Drift Fraction	0.000840	0.001077	0.000843	0.000768	0.001071
MSE of COV	0.015327	0.015289	0.015317	0.013275	0.016260
MSE of VMD	465.59	462.39	448.93	438.88	458.27
Learning Epochs	2524	711	1002	3471	352

The weights of the neural network will be read by SAGA2NN. Without using the expensive simulation engine, SAGA2NN relies on the weights to map the values of drift fraction, COV and VMD from a set of spray parameters. Experiments show that the error of the fitness calculated by the neural network from the actual fitness from AGDISP is within 5%. Originally, the range of fitness values in the initial population was from 1000 to 8500. After applying the neural network to select individuals with high fitness from a large pool, the range of fitness values in the initial population greatly increased, from 8000 to 9300.

The main interface of SAGA2NN, the interface to preset spray parameters, and the interface to customize SAGA2NN parameters are similar to those of SAGA2.

SAGADO

GADO (Genetic Algorithm for Design Optimization) was developed to solve the general engineering design optimization problem (Rasheed, 1998, Rasheed et al., 1999). It uses a steady-state GA. One offspring is produced from two parents selected from the population via some selection scheme, and replaces an existing individual in the population via some replacement strategy. The selection scheme is ranking selection, which sorts the population from best to worst and assigns the selection probability of each chromosome not according to its raw fitness but according to its ranking. The replacement strategy is a crowding technique, which takes into consideration both the fitness and the proximity of the points in the population. Several crossover and mutation operators are used, in which the most important one is guided crossover. The idea of guided crossover is to form different search directions by joining pairs of previously evaluated points, rank these directions, and take a small step in the best direction. It endows the GA with gradient-like capabilities without actually computing any gradients.

We developed SAGADO (Spray Advisor using Genetic Algorithm for Design Optimization) by adapting GADO to the aerial spray parameter problem. Besides passing the set of spray parameters needed to GADO, we added some routines to calculate the fitness of an individual, update the evolution process, and save the result. The main interface of SAGADO and the interface to preset spray parameters are similar to those of SAGA2. The interface to customize SAGADO parameters is shown in Figure 5. The meanings of these parameters are described in Rasheed (1998).

SASA

The above methods are all based on the genetic algorithm. Simulated annealing is also a widely used global stochastic optimization technique. In order to compare its performance with the genetic algorithm, we develop SASA (Spray Advisor using Simulated Annealing), which is based on simulated annealing. The basic algorithm can be described as follows: successively generate a random move by perturbing the values of the spray parameters in their neighborhood, accept the move if it leads to a solution with lower energy (better deposition results) than the current solution, otherwise accept it with probability $e^{-\delta/T}$, where δ is the change of energy and T is the current temperature. The energy of a solution is the opposite of its fitness. The temperature is initially high and gradually reduced. The cooling schedule adopted is geometric cooling (Aarts and Korst, 1989). The temperature is updated using the formula: $Ti+1 = \alpha Ti$ ($\alpha \in (0, 1)$), where α denotes the cooling factor. Typically α ranges from 0.8 to 0.95. Its default value in our implementation is set to 0.92.

The main interface of SASA is shown in Figure 6. The interface to preset spray parameters is similar to that of SAGA2. The interface to customize SASA parameters is integrated in the main interface.

Results

In order to test the performance of these methods, we run them on three practical spray parameter specifications provided by Forest Service managers (the details are in Table 3). The simulation numbers are all set to 5000. Crossover rate is 1, mutation rate is 0.1 and population size is 100, which are the same for SAGA, SAGA2, SAGA2NN and SAGADO. The other parameters of SAGA2 and SAGA2NN are a tournament size of 2 and a 0.2 replacement rate, respectiviely (see Figure 3). The other parameters of SAGADO are shown in Figure 5. The parameters of SASA are shown in Figure 6. VMDCenter is 100.

Table 3. Detail of Spray Parameter Settings

Parameter Settings	Variables Constraint
Parameter Setting I	None
Parameter Setting II	Aircraft ID = 6, Swath Width = 1.2 VMD = 100, Block Width = 400
Parameter Setting III	Aircraft ID = 106, Swath Width = 2.25

Due to the stochastic nature of the genetic algorithm, different runs can have different results. Therefore each method was run five times with different random seeds. The five random seeds were 1702803237, 1517566982, 1368775034, 1918061247, and 1648047133. The results are in Table 4, Table 5 and Table 6. The value in the cell is the maximum fitness of each run (the corresponding maximum fitness for SASA is the opposite of its minimum energy). The "mean value" row is the average value of the five maximum fitness values, which indicates the performance of each system.

From the tables, we can see that the maximum fitness values achieved for SAGA2, SAGA2NN (except for Random Seed 3 in Parameter Setting II) and SAGADO are better than SAGA in every parameter setting using any random seed. In Parameter Setting I and II, the performance of SAGADO is the best. SAGADO had the best individual result as well as the best average result for Setting I. For Setting II, SAGADO had the best average result while SAGANN had the best individual result. In Parameter Setting III, the performance of SAGA2NN is the best (both individual and average result). SAGA, SAGA2, SAGA2NN and SAGADO all easily outperform SASA.

Table 4. Results of Spray Parameter Setting I

Random Seed	SAGA	SAGA2	SAGA2NN	SAGADO	SASA
Random Seed 1	9926.20	9966.31	9927.87	9973.24	9693.17
Random Seed 2	9943.15	9967.71	9963.99	9951.65	9263.01
Random Seed 3	9951.56	9971.04	9969.70	9975.32	9785.40
Random Seed 4	9926.03	9963.79	9938.80	9973.34	9602.15
Random Seed 5	9925.44	9952.54	9970.73	9982.42	9721.73
Mean Value	9934.48	9964.28	9954.22	9971.20	9613.09

Table 5. Results of Spray Parameter Setting II

Random Seed	SAGA	SAGA2	SAGA2NN	SAGADO	SASA
Random Seed 1	9633.54	9655.79	9656.42	9656.15	9561.04
Random Seed 2	9619.27	9650.91	9658.32	9649.60	9427.91
Random Seed 3	9633.41	9650.61	9632.53	9651.33	9386.88
Random Seed 4	9612.26	9655.79	9655.79	9658.13	9412.83
Random Seed 5	9631.96	9632.07	9651.44	9654.29	9545.72
Mean Value	9626.09	9649.03	9650.90	9653.90	9466.88

Table 6. Results of Spray Parameter Setting III

Random Seed	SAGA	SAGA2	SAGA2NN	SAGADO	SASA
Random Seed 1	9469.19	9582.10	9583.19	9598.99	9367.72
Random Seed 2	9462.65	9588.06	9553.53	9608.46	9192.00
Random Seed 3	9468.48	9599.51	9575.86	9589.43	9342.84
Random Seed 4	9489.46	9585.76	9699.12	9574.29	9235.15
Random Seed 5	9427.02	9589.19	9622.69	9601.99	9344.23
Mean Value	9463.36	9588.93	9606.88	9594.63	9296.39

It may be difficult to see how much better a method is from the fitness value. The advantage of a method will be obvious from the three output values. For example, Table 7 shows the output values of these methods for Parameter Setting I with Random Seed 1. Although the fitness value of SAGA2 or SAGADO is not much higher than that of SAGA, drift fraction and COV of either method are greatly improved. For SAGA2, drift fraction is reduced by 59%, and COV is reduced by 46%; For SAGADO, the drift fraction is reduced by 81%, and the COV is reduced by 54%. As mentioned earlier, drift is a very important aspect of aerial spray application and is watched closely to determine actual off-target contamination. With respect to pest control, COV is very important since better COV values relate directly to an increased probability of pest control.

Table 7. Output values of Parameter Setting I with Random Seed 1

Output Values	SAGA	SAGA2	SAGA2NN	SAGADO	SASA
Fitness	9926.20	9966.31	9927.87	9973.24	9693.17
Drift Fraction	0.002672	0.001096	0.003268	0.000505	0.018312
COV	0.021023	0.011282	0.022096	0.009694	0.065420
VMD	100.31526	100.00046	99.97806	99.99997	97.93129

In order to see the evolution process, we recorded the maximum fitness every one hundred simulations. The evolution graphs for these three spray parameter settings are shown in Fig. 7, Fig. 8 and Fig. 9 (the x-coordinate is the simulation number divided by 100, and the y-coordinate is the average value of the maximum fitness in various runs). From the graphs, we can see that SAGA2NN obtains a much better maximum fitness value in the first few hundred simulations, however, its lead is offset later. In these five methods, the performance of SAGA2, SAGA2NN and SAGADO is equally good. They are a little better than SAGA in Parameter Setting I and II, and much better than SAGA in Parameter Setting III. The performance of SASA is the worst in all cases.

Conclusion

In this paper, we described several heuristics to deal with the aerial spray parameter optimization problem, and compared them with the original heuristic technique SAGA. SAGA2 and SAGADO outperform SAGA in all experiments. It proves that exquisite choice of the type of genetic algorithm, selection, crossover and mutation operator can boost GA performance. In this problem, the steady-state GA plus tournament selection is preferred over the generational GA plus roulette wheel selection. By applying a neural network to the genetic algorithm, we hoped SAGA2NN could achieve even better results than SAGA2 and SAGADO, but our experiments don’t show this. We think the reason is that the advantage of the neural network is counteracted by premature convergence of the genetic algorithm. However, SAGA2NN converges very fast, which is useful in real aerial spray applications because it can get the near-optimal result by far fewer simulations. SASA performs much more poorly than the methods based on the genetic algorithm. It shows that (for us) genetic algorithms perform better than simulated annealing in an extremely complicated domain such as aerial spray deposition management.

References

E. Aarts, J. Korst (1989). Chapter 4, *Simulated Annealing and Boltzmann Machines*, John Wiley & Sons, New York.

A.J. Bilanin, 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 American Society of Agricultural Engineers,* 32(1): 327-334.

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

L. Eshelman and J. Schaffer (1993). Real-coded genetic algorithms and interval-schemata. *Foundations of Genetic Algorithms*, Vol. 2: 187-202, Morgan Kaufman, San Francisco, CA.

S. I. Gallant (1993). Chapter 11, *Neural Network Learning and Expert Systems*, the MIT Press, Cambridge, MA.

D.E. Goldberg (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning.* Addison-Wesley, Reading, MA.

D.E. Goldberg, B. Korb, and K. Deb (1989). Messy genetic algorithms: motivation, analysis, and first results. *Complex Systems*, vol.3: 493-530.

D.E. Goldberg, and K. Deb (1991). A comparative analysis of selection schemes used in genetic algorithms. *Foundations of Genetic Algorithms*: 69-93, Morgan Kaufman, San Mateo, CA.

J.J. Grefenstette (1986). Optimization of control parameters for genetic algorithms. *IEEE Transactions on Systems, Man, and Cybernetics*, 16:122-128.

J. Holland (1975). *Adaptation in Natual and Artificial Systems*. University of Michigan Press, Ann Arbor, MI.

Z. Michalewicz, T. Logan, and S. Swaminathan (1994). Evolutionary operators for continuous convex parameter spaces. *Proceedings of the 3rd Annual Conference on Evolutionary Programming*: 84-97, World Scientific Publishing, River Edge, NJ.

Z. Michalewicz (1996). *Genetic Algorithm + Data Structure = Evolution Progams*, 3rd Edition, Springer-Verlag, New York.

W.D. Potter, W. Bi, D. Twardus, H. Thistle, M.J. Twery, J. Ghent, M. Teske (2000). Aerial Spray Deposition Management Using the Genetic Algorithm, in *Proceedings of the 13th Int. Conf. On Industrial & Engineering Applications of Artificial Intelligence and Expert Systems*, IEA/AIE'2000, pp. 210-219, New Orleans, June 2000.

K. Rasheed (1998). GADO: *A Genetic Algorithm for Continuous Design Optimization*. Ph.D. Thesis, Department of Computer Science, Rutgers University, New Brunswick, NJ, 1998.

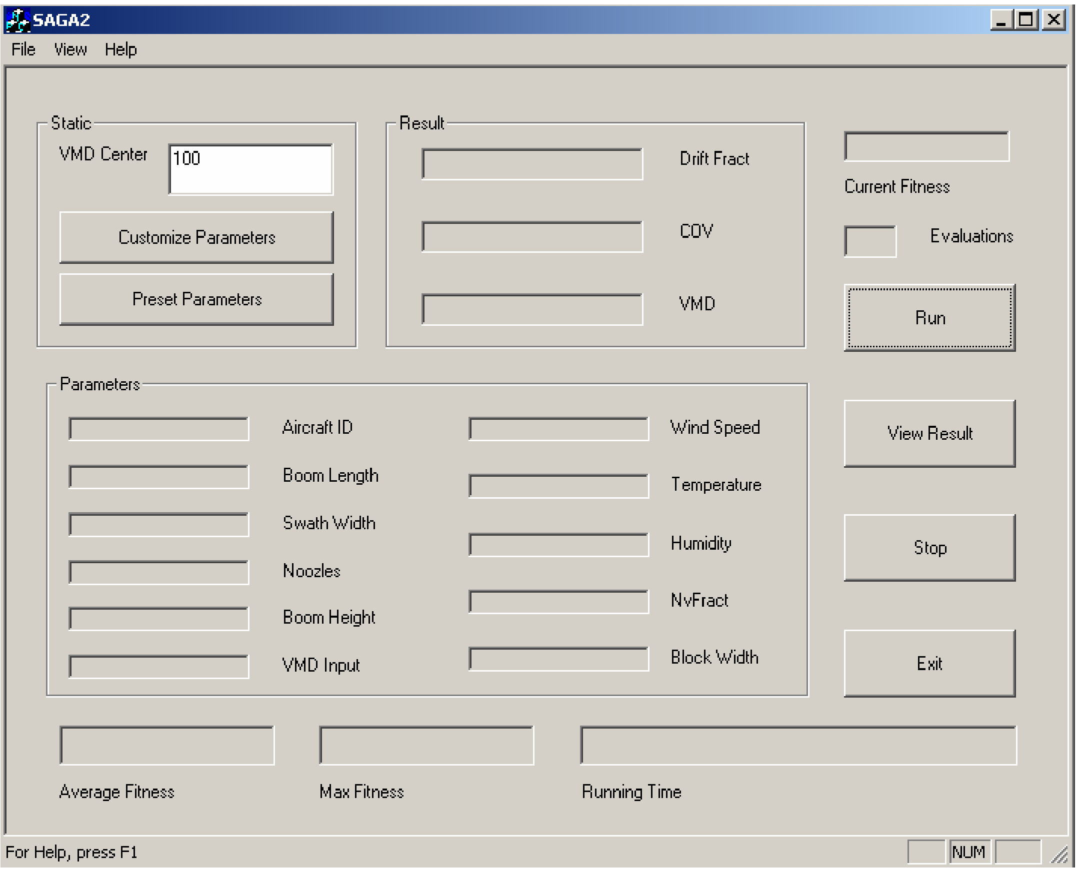
K. Rasheed, H. Hirsh (1999). Learning to be Selective in Genetic-Algorithm-Based Design Optimization. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, 13:157-169.

K. Rasheed, H. Hirsh (2000). Informed operators: Speeding up genetic-algorithm-based design optimization using reduced models. *Genetic and Evolutionary Computation Conference 2000.*

G. Syswerda (1989). Uniform crossover in genetic algorithms. *Proceedings of the 3rd International Conference on Genetic Algorithms*: 2-9, Morgan Kaufman, San Francisco, CA.

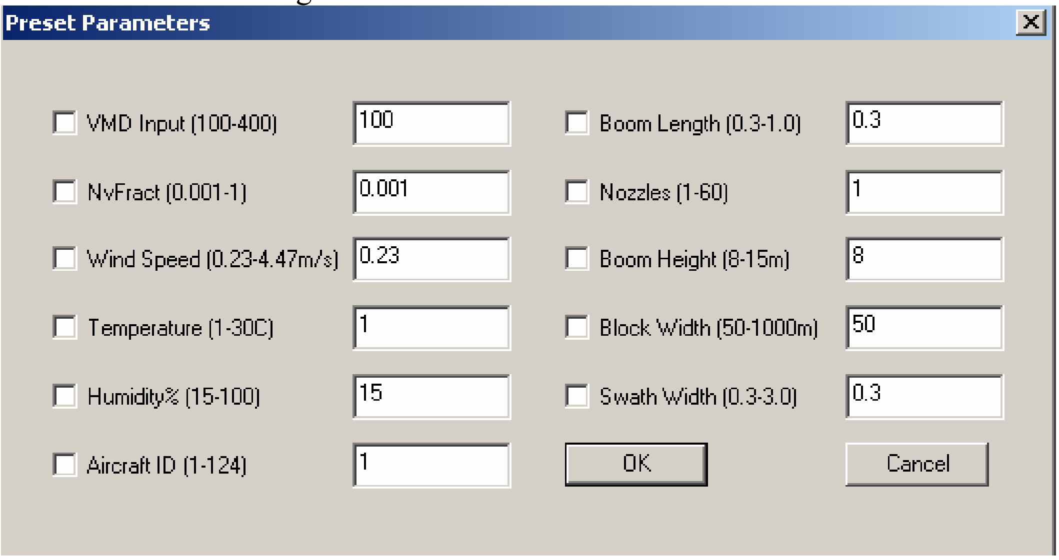
M.E. Teske, T.B. Curbishley (1989). *Forest Service Aerial Spray Computer Model 4.0 User Manual*. C.D.I. Report No. 90-06.

L. Zadeh (1963). Optimality and non-scalar-valued performance criteria. *IEEE Transactions on Automatic Control*, vol. 8, no 59.



VMD Center 100

Figure 1. The Main Interface of SAGA2



☐ VMD Input (100-400) 100

☐ Boom Length (0.3-1.0) 0.3

☐ NvFract (0.001-1) 0.001

☐ Nozzles (1-60) 1

☐ Wind Speed (0.23-4.47m/s) 0.23

☐ Boom Height (8-15m) 8

☐ Temperature (1-30C) 1

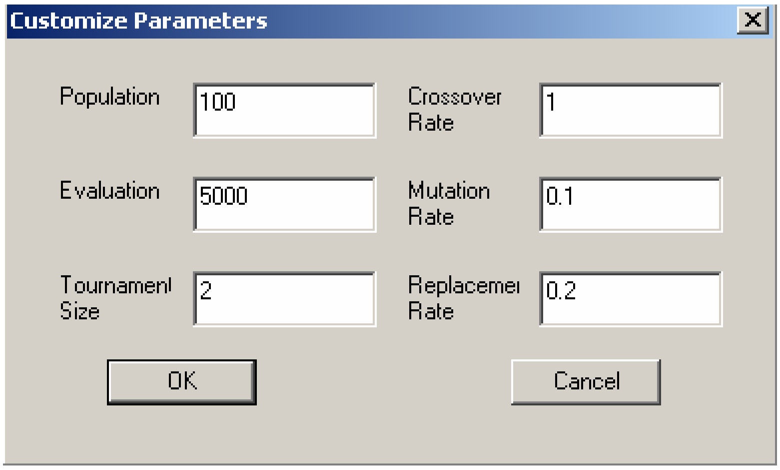
☐ Block Width (50-1000m) 50

☐ Humidity% (15-100) 15

☐ Swath Width (0.3-3.0) 0.3

☐ Aircraft ID (1-124) 1

Figure 2. The Interface to Preset Spray Parameters


Customize Parameters

Population 100

Crossover Rate 1

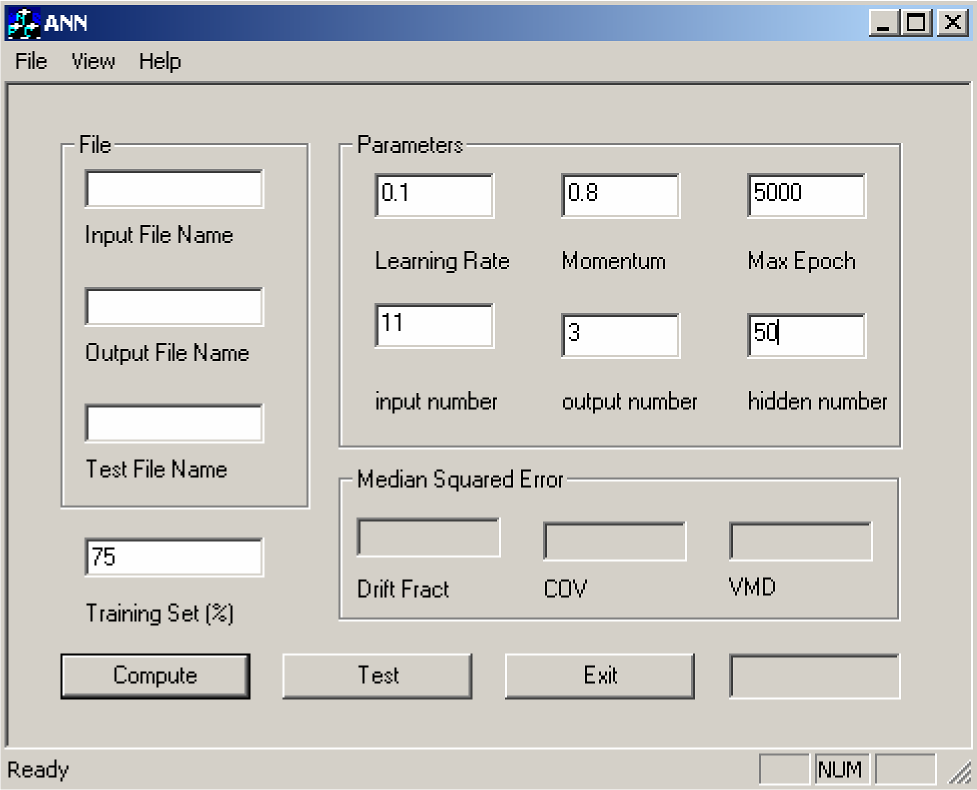
Evaluation 5000

Mutation Rate 0.1

Tournament Size 2

Replaceme Rate 0.2

Figure 3. The Interface to Customize SAGA2 Parameters


ANN

File

Input File Name

Output File Name

Test File Name

75 Training Set (%)

Parameters

0.1 Learning Rate

0.8 Momentum

5000 Max Epoch

11 input number

3 output number

50 hidden number

Median Squared Error

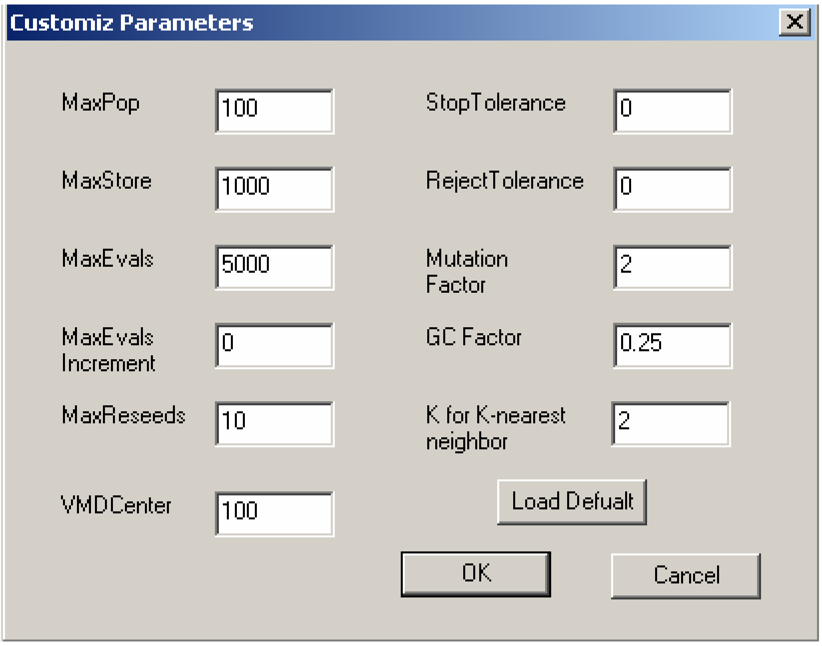
Drift Fract

COV

VMD

Ready

Figure 4. The Interface to Train the Neural Network



MaxPop 100

MaxStore 1000

MaxEvals 5000

MaxEvals Increment 0

MaxReseeds 10

VMDCenter 100

StopTolerance 0

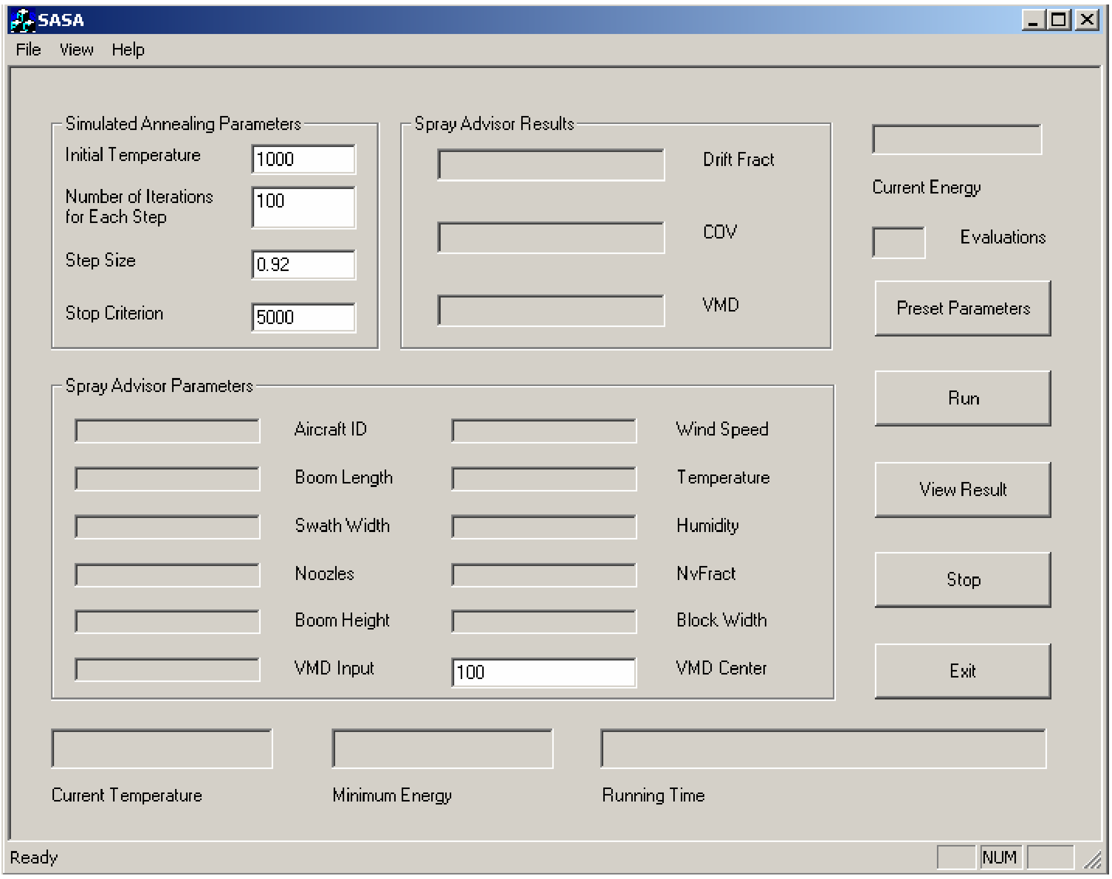
RejectTolerance 0

Mutation Factor 2

GC Factor 0.25

K for K-nearest neighbor 2

Figure 5. The Interface to Customize SAGADO Parameters



Simulated Annealing Parameters

Initial Temperature 1000

Number of Iterations for Each Step 100

Step Size 0.92

Stop Criterion 5000

VMD Center 100

Figure 6. The Main Interface of SASA

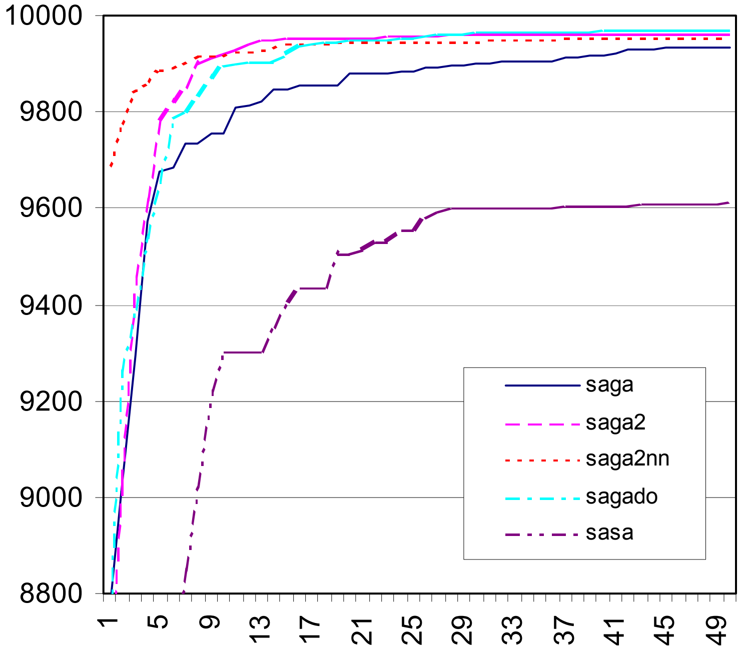


Fig. 7 Evolution Process of Parameter Setting I

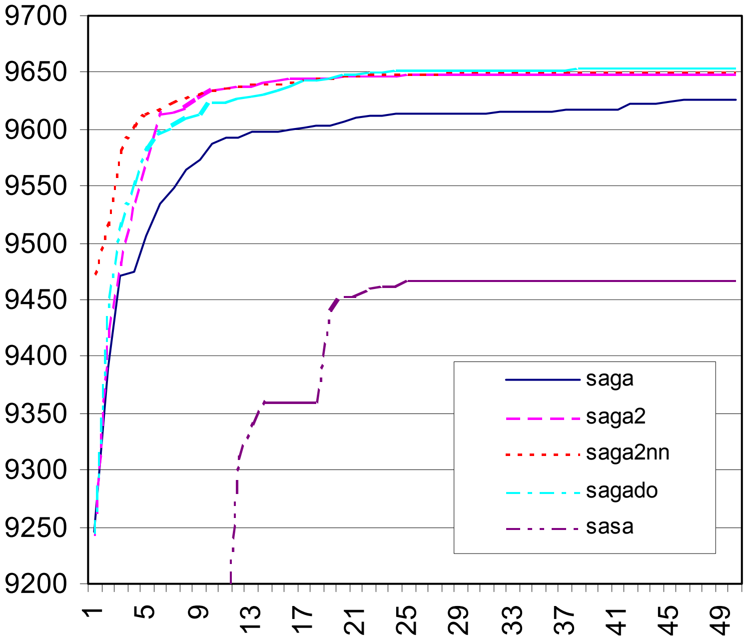


Fig. 8 Evolution Process of Parameter Setting II

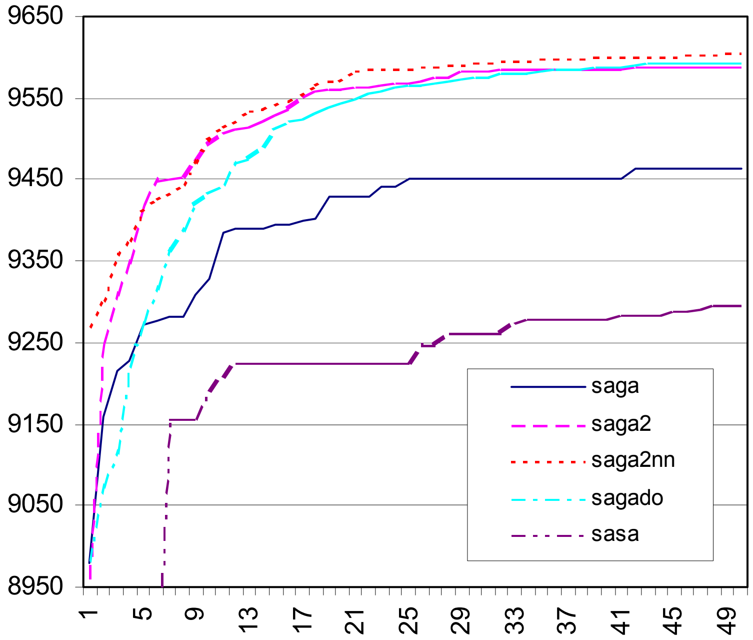


Fig. 9 Evolution Process of Parameter Setting III