

AUTONOMOUS GENETIC ALGORITHM FOR FUNCTIONAL OPTIMIZATION

Z.-Q. Meng

Department of Electrical Engineering
Fukuoka University
8-19-1 Nanakuma, Jonan-ku, Fukuoka, 814-0180, Japan

Abstract—Genetic algorithm (GA) is effective for global optimizations, but needs the user to define several parameters. Unless these parameters are defined appropriately, search efficiency drops significantly. There are, however, no clear rules for the defining, and almost all users have considerable difficulty to use GA efficiently. A good algorithm must be use-friendly. It should not, if possible, need the user to define such parameters and can play high performance for any optimization problem. This paper proposes an autonomous GA addressing these problems.

1. INTRODUCTION

There are cases when optimization is required in the field of electromagnetic engineering, such as optimizing design parameters, resolving a nonlinear inverse scattering problem as an optimization problem, and so on. Usually, an objective function has multiple local solutions due to the function complexity or errors in approximately calculation, and a global applicable optimization method is necessary. Genetic algorithms (GA) have achieved considerable success [1–9] as a global optimization method with a wide range of applicability, requiring no prior information, being appropriate for diverse objective function, and offering the ability of recovery even after trapping on a local solution. GAs, however, have the following problems.

(i) Difficult to use

GAs require several parameters, including the probabilities for mutation and crossover of genes, and the population size (the number of individuals). Unless these parameters are set appropriately, search efficiency drops significantly. There are,

however, no clear rules for setting these parameters, and almost all users have considerable difficulty setting them without prior experience in parameter tuning.

(ii) Slow convergence

Global optimization methods do not utilize differential information, so convergence is usually slower than when using the descent category optimization method. GA also does not use this information, leading to a large quantity of GA improvements and a wide range of proposed methods [10–14]. If we blindly seek the convergence speed, however, the likelihood of trapping on a local solution also increases. A resolution to this problem is required to develop faster GAs.

This paper proposes an autonomous GA that does not need user to define the parameters and converges to the solution rapidly.

The details about GA general procedure are omitted here, because there are so many GA textbooks.

2. THE PROBABILITY FOR MUTATION

To observe the effect of mutation probability concretely, let us first consider the maximization of function

$$\Omega_1(x_1, x_2, \dots, x_n) = \prod_{i=1}^n \frac{\sin x_i}{x_i}, \quad -10 < x_i < 10. \quad (1)$$

Figure 1 shows the case when $n = 2$ for (1).

2.1. Performance vs. Mutation Probability

The performance of GA can be estimated with the quantity of performed fitness evaluations (PFE), which means how many times the objective function was calculated until the algorithm convergence.

The author applied a general GA (tournament type selection, one point type crossover) to maximizing (1) for cases $n = 2$ and $n = 5$, with an admissible error of 10.4. The PRE vs. Mutation Probability is shown in Fig. 2. In this diagram, the result is the average for 200 tests, taking the fact that the GA is a probabilistic algorithm into account.

The following points are evident:

- (i) GA's performance is susceptible to changes in mutation probability;
- (ii) the optimal values of mutation probability for the two cases are different and if one use 0.2, the optimal value for case $n = 2$,

to the optimizing of case $n = 5$, the convergence speed drops significantly.

The waveform for $n = 5$ cannot be shown as a figure, but imagine a form similar to Fig. 1 within 5D space. In spite of the fact that the objective functions are similar, the characteristics of the PFE count to mutation probability are widely different. This fact means that an appropriate mutation probability should be determined by tuning it with respect to the specific objective function, and the result of this tuning, however, may be not universally applicable to other objective functions. Thus it is impractical to find the optimal mutation probability by such a tuning. Another idea is needed.

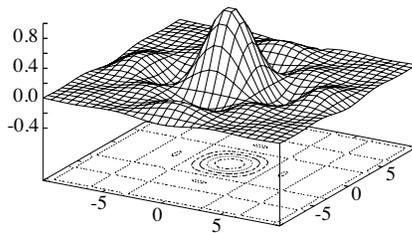


Figure 1. Function (1) surface plot for $n = 2$.

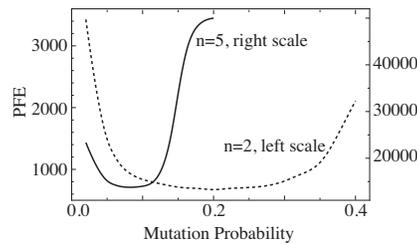


Figure 2. Performed fitness evaluations vs. mutation probability.

2.2. Homozygous Mutation

As mentioned in Section 1, there are no clear guidelines for adjusting the mutation probability, and it is difficult to develop fitness-based mutations. This paper focuses on the genes at the root of the GA. The search efficiency is dependent on the speed of evolution of the virtual individual. From the viewpoint of biological evolutionary theory, individual competition for survival and gene diversity are the two prerequisites for evolution. The higher the survival competition is, the higher the pressure on improving fitness, with superior genes inherited by offspring. But the gene diversity may be reduced because individuals with even slightly slower fitness are culled. A population lacking diversity ceases to evolve, making it crucial to maintain a balance between inheritance and diversity. Mutation is needed to keep the diversity. Excessive mutating, however, may disturb the evolution, because superior genes may also be destroyed.

Compared to the quantity of generations involved in species survival in the natural world, the quantity of mutations occurring in the gene is extremely low. This is because the species has sufficient

diversity. If the quantity of individuals is extremely small, inbreeding is unavoidable, and it is well known that this leads to an increase in the spontaneous mutation probability. This mechanism can be interpreted as a natural “emergency measure” to preserve diversity in the species.

The author introduced this mechanism into GAs, eliminating the usual spontaneous mutation processing and instead causing mutation only in same-type pairings (homozygous mutation). Concretely, the target genes of the parents are compared prior to crossover, and if they are identical to each other, then a same-type pairing is assumed, and spontaneous mutation occurs. As pairs are selected at random for crossover, matching genes means that the ratio of that gene is high within the population, probabilistically speaking. Mutation of a gene with a high incidence within a population will decrease uniformity and increase diversity. If there is no gene which is common within a population, on the other hand, there would be no spontaneous mutation, and therefore no interference with the course of natural evolution. As there is no point in crossing identical genes, the introduction of spontaneous mutation to inbreeding will improve search efficiency. Homozygous mutation does not require the control of sensitive parameters, unlike spontaneous mutation, is independent of the virtual environment (including methods of natural selection and crossover, and objective function), and naturally preserves diversity within the species. Further, taking into consideration that

- (i) inbreeding in nature does not always result in offspring with significantly different genes, and
- (ii) minor mutations centered on existing genes may be required in latter evolutionary stages,

the homozygous mutation plays a general uniform distribution mutation in one of the parents of the same-type pair, and a standard Gaussian distribution (centered at value of the current gene) mutation in the other.

The performance of GA with the homozygous mutation is shown in Section 4.

3. THE PROBABILITY FOR CROSSOVER

The effect of crossover probability on GA performance is not as strong as that of mutation probability. Since the hope of getting the solution is always placed on new chromosomes in the next generation, usually the probability for crossover is large but not one, because it also destroys superior genes.

The author implements crossover for all pairs due to the following reasons.

- (i) The most superior individual is unconditionally seeded into the next-generation population to ensure the inheritance (the royalty system).
- (ii) Along with the function for automatic adjustment of homozygous mutations, the crossover probability can be little larger than that for a general GA (without homozygous mutation).

4. THE PERFORMANCE OF HRGA

For purposes of simplification, the GA with the combination of homozygous mutation and the royalty system, with the elimination of the mutation and crossover probability, is referred to below as “HRGA”.

The author applied HRGA and a standard GA to maximize (1) to compare their performance. The standard GA uses tournament method for natural selection and two-point crossover, and its crossover and mutation probabilities are determined from tuning based on a large quantity of simulations for the objective function. For comparison, the population size was set to 30 for both HRGA and the standard GA. Figure 3 shows trends in fitness over generations when using $n = 2$ in (1). GA probability is taken into account by taking the average of 500 tests in Fig. 3 (and in Fig. 4). It is evident that HRGA only requires about half as many generations as the standard GA to finish the evolution.

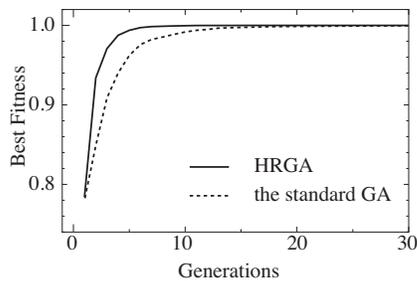


Figure 3. Comparison of HRGA and the standard GA ($n = 2$ case), where the fitness value is equivalent to the value of Ω_1 .

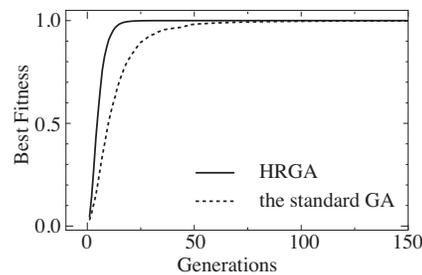


Figure 4. As Fig. 3, but $n = 5$.

Next the author used $n = 5$ in (1). No adjustment was used for HRGA, but the crossover and mutation probabilities for the standard GA were reset. Fitness trends (Fig. 4) show that

the standard GA required about four times as many generations as HRGA. One of the reasons the search efficiency of HRGA without any tuning was superior to that of the standard GA with tuning is thought to be that the mutation probability for the standard GA was fixed. Tuning was implemented experimentally on the final results, but remained unchanged throughout evolution once set. In natural evolution, however, the quantity of species is constantly changing. When the quantity of species is high there is no need for mutation, and forced mutation can result in the loss of superior genes, having an adverse effect on evolution. When the quantity of species is too low (uniformity), mutation must be increased. HRGA resolves this problem rationally and simply.

5. THE POPULATION SIZE

Population size is the only parameter that must be set for using HRGA in Section 4. In this section, an algorithm that can automatically adjust its population size is discussed.

There is no generally effective value for population size because it varies with the specific optimization goal (objective function). From the viewpoint of probability, it is evident that the probability of discovering a solution is proportional to the quantity of sample points, which is another way of saying the population size.

The efficiency of GA is not dependent on random search probability, the point is evolution. As long as the diversity of the population is not sacrificed, an increase in population size will not make a major contribution in evolution, but will increase the cost of evaluating individuals from each generation. A comparison of two strategies – using a large population and reducing the quantity of generations, or using a small population and driving evolution over many generations – indicates that the latter approach offers better search efficiency as long as the population size is not excessively small.

Experiment results for the application of HRGA in maximizing (1) using dimension values $n = 2, 3, 4, 5$ are shown in Fig. 5. With a population of 10, the quantity of evaluations of the objective function needed to find the solution is much larger, indicating that the population size is too small. When the population is increased to 15, 20 or 30, it becomes possible to preserve species diversity and evolution proceeds smoothly, lowering the quantity of evaluations. An increase in population above 40 tends to again increase the quantity of evaluations, indicating that an excessively large quantity of individuals will degrade search efficiency.

Consider the optimal population size. In addition to the

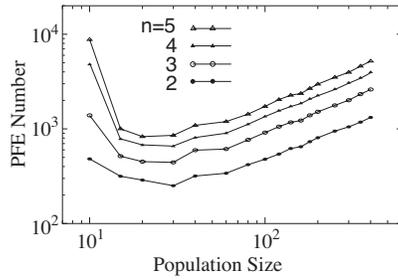


Figure 5. Performed fitness evaluation number (PFE) vs. population size.

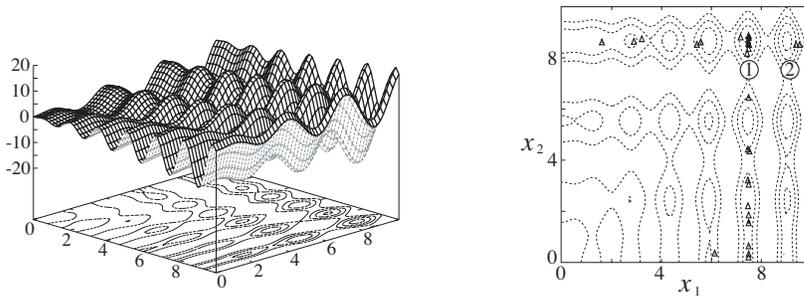


Figure 6. Surface plot of Ω_2 in (2).

Figure 7. A population distribution.

experiment shown in Fig. 5, the author also investigated the applicability of GA to multiple optimization problems with single-peak objective functions (continuous functions with only one minimum or maximum), with similar results. For single-peak and other simple functions, 20 to 30 individuals appear to offer good performance.

When objective functions are complex and there are many local solutions, it is possible to avoid the problem of having the entire population trap on local minima by increasing the population size, but the problem of how large an increase is appropriate remains unresolved.

The waveform for 2D function

$$\Omega_2(x_1, x_2) = x_1 \sin 4x_1 + x_2 \sin 2x_2, \quad 0 \leq x_i \leq 10 \quad (2)$$

is shown in Fig. 6, while Fig. 7 is the contour diagram for $\Omega_2 < 0$. There are multiple local minimum, and the minimum is in region ② of Fig. 7.

The distribution of sample points of 10th-generation for the

application of the standard GA with a population size of 60 in minimizing Ω_2 is shown in Fig. 7 with \triangle . Two-thirds of the individuals (40) are concentrated in local minimum ①, and the remainder have low fitness (note the contour lines) and are unlikely to survive in the next generation in a high-competition environment. This is only one example, and in fact most of the minimizing experiments run resulted in success, but it does point out that even with a large population it is possible that the population traps on a local minimum.

Because 20 to 30 individuals are sufficient for single-peak functions, to prevent the entire population from trapping on a local minimum it is of course best to disperse some of the individuals groups at ① to other regions.

To do this, the information about local minima basins (LMBs) of attraction of the objective function is needed. GAs handle large amounts of information related to the objective function, but conventional algorithms apply this information only to evaluation of individuals. If this objective function information can be utilized more effectively it becomes possible to grasp the LMBs.

5.1. An Algorithm for Grasping LMBs

The author proposes a technique called “multi-class” capable of grasping LMBs of a continuous objective function by using the information about the function obtained in each generation of a GA. The term “class” here can be understood as a LMB from the viewpoint of search, but within the virtual world of GAs it is closer to a community. Many virtual individuals belong to several communities, evolving through survival competition within that community.

The multi-class GA flow is as outlined below.

- (i) Grid the search-space (n dimensional space).
In principle the cell width can be user-defined, but it should be fine-grained enough to ensure that adjacent local solutions are not identical or belonging to the adjacent cell. For example, the width can be a multiple of the allowable margin of solution error defined by the user. The maximum fitness of individuals belonging to a cell is defined as the cell fitness, with an initial value of zero.
- (ii) Generate an initial population, calculate individual fitness, and check for a solution, in the same manner as for standard GAs.
- (iii) Update the cell fitness.
If the maximum fitness of individuals in a cell is higher than the cell fitness, let the fitness be the cell fitness.
- (iv) Divide the search-space among the classes (a class denotes a LMB) in accordance with the classification procedure, the following

algorithm:

- (a) Cells with fitness of the average or below are classed to a special class named *vagrant*. Others are classified as below.
- (b) Set class number $n_c = 1$.
- (c) For cells not yet assigned to classes, assign the one with the highest fitness to the n_c^{th} class. The cell is called *center cell* of the class.
- (d) If there is a newly assigned cell C_{new} within the n_c^{th} class proceed to step (e), otherwise to step (g).
- (e) The adjacent cells of C_{new} (Fig. 8) satisfying the following conditions will also be assigned to that class:
 - do not belong to any other class, and
 - fitness is lower than the cell C_{new} .
- (f) Return to step (d).
- (g) If there are any unassigned cells, set $n_c = n_c + 1$ and return to step (c) to assign the cells to the new class, otherwise end processing.

The distance between two class's center cells is called *class distance*. The individuals existing in a cell also belong to the class to which the cell is assigned.

- (v) For each class, construct a sub-population with its individuals, and generate the next-generation sub-population in the same manner as for standard GAs. However,
 - if the quantity of individuals s belonging to a class is below a preset value (in this paper, $s < 10$), the class is disqualified to a community, and the individuals are instead assigned to class *vagrant*;
 - the maximum sub-population size is 20.
If $s > 20$, create the sub-population to include only the 20 individuals with the highest fitness values, excluding class *vagrant*.
- (vi) Return to step (ii).

In fact, the cell with the highest fitness is taken as the center, and adjacent cells with lower fitness values are assigned to its class, and extending the class. If no extension is possible then the unassigned cell with the higher fitness is taken as the center for a new class, and the process repeated until all cells are assigned.

Figure 9 shows an example for the classification procedure. (a) The search-space is gridded and the fitness of each cell is set to zero. (b) Estimate the individuals. (c) Update cell fitness. If there is more than one individual in a cell, such as cell (2,4), the cell fitness is set with the best individual fitness. The average fitness of cells is 6.65. (d)

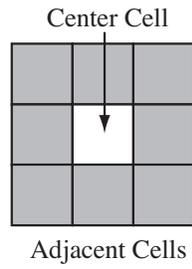


Figure 8. Center and adjacent cells.

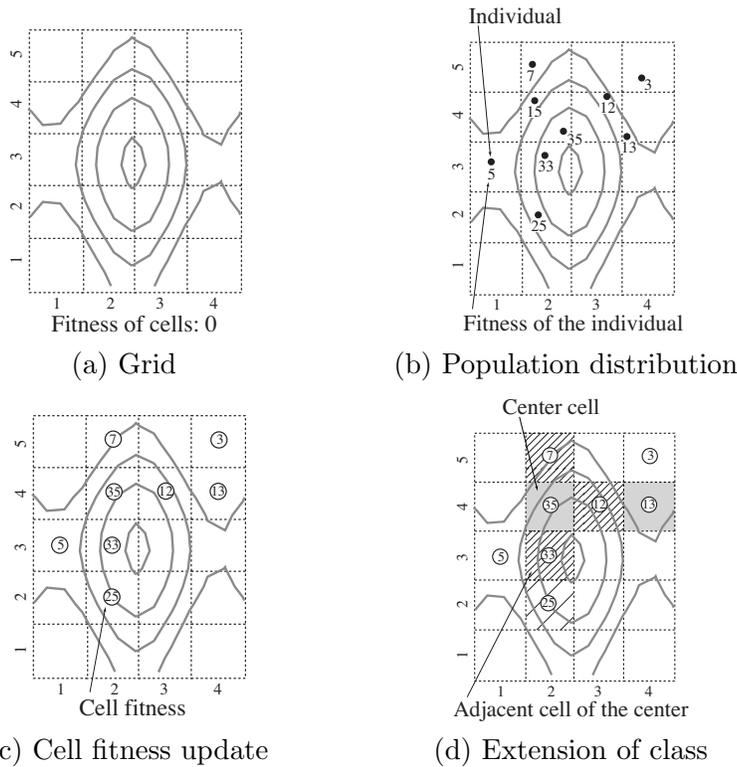


Figure 9. An example for the classification procedure.

The best cell is assigned to class (cell (2,4), the center cell of the class), and then the adjacent cell of cell (2,4) is checked if it can be assigned to the class. Cell (2,3), (3,4) and (2,5) are assigned to the class because their fitness is larger than the average fitness and smaller than that of

cell (2,4), but cell (1,3) is not. Similarly, cell (2,2) is also assigned to this class as adjacent cell of cell (2,3). Cell (4,4) is a adjacent cell of cell (3,4), but it is not assigned to this class because its fitness is larger than cell (3,4). It will be selected as a center cell of next class.

It is evident that there are two errors of the class region in Fig. 9(d). One is about the center cell. Cell (2,3) or (3,3) should be the center cell, not (2,4). The other is cell (4,4) should also be assigned to this class, not other class. The errors have those roots in the imperfection of information on objective function. Because the generation is done for each class and the individuals in center cell have high fitness, the information on center cell and its neighborhood will be closely investigated, and the errors will be corrected after several generations. In the early stage of evolution, LMBs may not be expressed by the classes correctly (the center cell and region may change generation by generation), but as evolution proceeds and more and more information is accumulated, the class region comes close to the LMB, in order of LMB depth. After the class distance becomes fixed, the cell width can be adjusted. Double the width when the shortest class distance is large enough (for example, longer than the width of 8 cells). For 4 or fewer cells, halve the width.

5.2. A Test of Multi-Class GA

Figure 10 shows an example of class transition when the multi-class GA is applied to minimize Ω_2 in (2). In this experiment the number of LMBs was an unknown, and initially 10 local solutions were tried with 200 individuals. The allowable margin of solution error was $\varepsilon = 10^{-2}$ (search space size = $10^3 \times 10^3$), the initial cell width 10 (total cell quantity 10,000) and fitness values saved for top 300 cells.

Figure 10(a) indicates classes assigned in an early generation with “□”. Cells containing individuals with high fitness values were assigned as class centers, but as there was no generational change or fitness information for cells adjacent to the center, each class holds only one cell.

Figure 10(b) shows the state in the fifth generation. Classes at ② and ③ are expanding, incorporating adjacent cells. Several classes exist for a single LMB, but if expansion continues they will reduce to a single class. New classes have appeared at ①, ④ and ⑤ and are expanding. The low-fitness classes evident in Fig. 10(a) have disappeared.

Figure 10(c) shows the class state after 30 generations. The LMBs in the upper right region each have classes, and they are quite large. The class center cells are stable, with ample spacing between classes, so cell width is also expanding.

Figure 10(d) shows the class state after 200 generations. The cells are quite large, and the class regions perfectly match the top ten LMBs, and ten local solutions are precisely obtained.

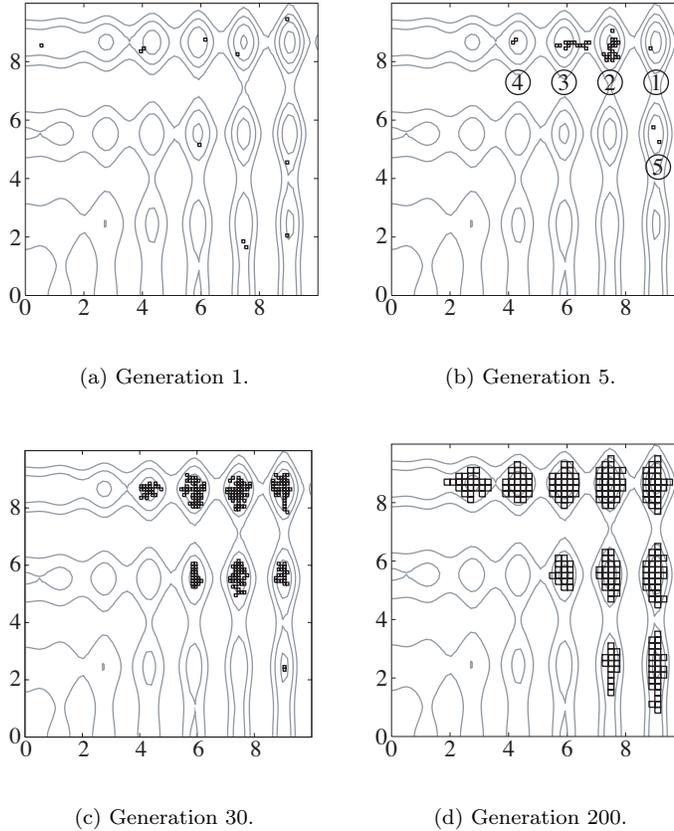


Figure 10. Transition in class regions.

5.3. Uses of Multi-Class GA

The LMB information obtained by multi-class GA, in turn, can be used for the following:

- limiting the sub-population size for each LMB. The objective function is a single-peak function, so about 20 individuals per LMB is sufficient. In addition, individuals assigned to a particular LMB can search for local solutions, as shown in subsection 5.2.

- developing a GA that can automatically adjust population size. For example, it starts with only 20~30 individuals, and automatically increases the quantity when multiple LMBs are identified.
- getting multiple local solutions in a single experiment, and determining the width and depth of each LMB. When used in design problems, for example, it becomes possible to select designs with low cost and high stability.

6. AN APPLICATION EXAMPLE

With the homozygous mutation and multi-class technique, one can construct an autonomous GA that does not need the user to define the probabilities for mutation and crossover of genes, and the population size. Here, as a simple application example, a design of a three layers radome shown in Fig. 11 is considered.

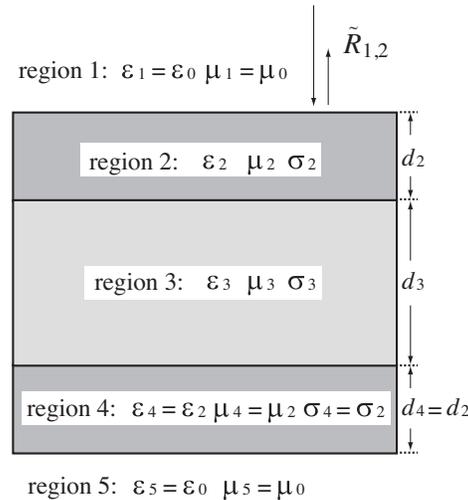


Figure 11. A three layers radome.

The reflection coefficient of a TE plane wave (vertical incidence) is given by the following recursive relations for the generalized reflection coefficients [15]:

$$\tilde{R}_{n,n+1} = \frac{R_{n,n+1} + \tilde{R}_{n+1,n+2} \exp(2jk_{n+1}d_{n+1})}{1 + R_{n,n+1}\tilde{R}_{n+1,n+2} \exp(2jk_{n+1}d_{n+1})}, \quad (3)$$

$$\tilde{R}_{4,5} = R_{4,5}, \quad (4)$$

where d_{n+1} ($n = 1, 2, 3$) is the layer thickness, $R_{n,n+1}$ is the Fresnel reflection coefficient of a plane wave at the interface between the n^{th} and $(n + 1)^{\text{th}}$ regions:

$$R_{n,n+1} = \frac{\mu_{n+1}k_{n,z} - \mu_n k_{n+1,z}}{\mu_{n+1}k_{n,z} + \mu_n k_{n+1,z}}, \tag{5}$$

and k_{n+1} is the wave number in region $(n + 1)$. For purposes of simplification, the layer 3 in region 4 is the same as layer 1 in region 2, and the other parameters are assumed as following:

$$\epsilon_2 = 3.65\epsilon_0, \quad \epsilon_3 = 6.32\epsilon_0, \tag{6}$$

$$\mu_2 = \mu_3 = \mu_0, \tag{7}$$

$$\sigma_2 = 0.1168, \quad \sigma_3 = 0.02338, \tag{8}$$

$$0.1\text{mm} < d_2, d_3 < 8\text{mm}. \tag{9}$$

What we want to do is

$$\min_{d_2, d_3} \tilde{R}_{1,2}.$$

The autonomous GA is applied to the optimization, where the allowable margin of solution error is assumed to 10^{-4}m for both d_2 and d_3 . As a result, four proposals shown in Table 1 and the class regions shown in Fig. 12 are obtained simultaneously.

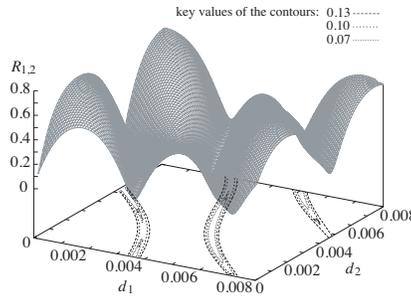
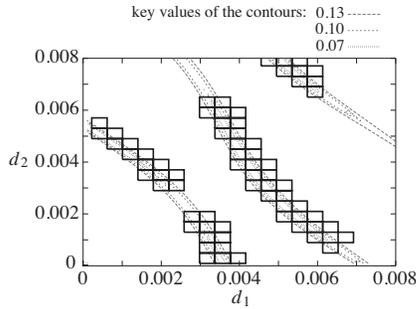


Figure 12. The class regions.

Figure 13. Waveform of $\tilde{R}_{1,2}$

Compared the class regions with the contours and the waveform of $\tilde{R}_{1,2}$ in Fig. 13, it is evident that the regions perfectly match the LMBs. One can choose a proposal from Table 1 for processing easily or cost-performance, and try to adjust it by making use of the LMB information if necessary.

Table 1. Design proposals of the radome obtained by the autonomous GA.

No.	1	2	3	4
d_2 [mm]	0.1	4.7	5.0	3.0
d_3 [mm]	5.4	3.0	8.0	1.5
$\bar{R}_{1,2}$	0.017	0.031	0.039	0.068

7. CONCLUSIONS

This paper proposed an autonomous genetic algorithm (GA) by introducing the homozygous mutation and multi-class technique into a general GA. The homozygous mutation is spontaneous mutation based on the genetic theory that inbreeding trigger mutation, eliminating the need for mutation and crossover probabilities, preserving population diversity naturally and making it easier to utilize GAs while improving search efficiency. The multi-class technique utilizes the information of objective function obtained by the GA in estimating applicability, for survey of the local minima basins (LMBs) of attraction. With the information of LMBs, the algorithm can automatically adjust the population size to match the complexity of the objective function to prevent the system from trapping on a local minimum, and efficiently obtain multiple local solutions simultaneously in one play. Moreover, it is possible to develop the algorithm to automatically switch over to the descent category optimization method once the LMBs are approximated.

REFERENCES

1. Chiu, C.-C. and P.-T. Liu, "Image reconstruction of a perfectly conducting cylinder by the genetic algorithm," *Proc. Inst. Elect. Eng. Microwaves, Antennas Propagat.*, Vol. 143, 249–253, 1996.
2. Weile, D. S. and E. Michielssen, "Genetic algorithm optimization applied to electromagnetics: A review," *IEEE Trans. Antennas Propagat.*, Vol. AP-45, 343–353, 1997.
3. Takenaka, T., Z.-Q. Meng, T. Tanaka, and W. C. Chew, "Local shape function combined with genetic algorithm applied to inverse scattering for strips," *Microwave and Optical Technology Letters*, Vol. 16, No. 6, 337–341, 1997.

4. Haupt, R. L. and S. E. Haupt, *Practical Genetic Algorithms*, A Wiley-Interscience Pub., John Wiley & Sons, INC., New York, 1998.
5. Meng, Z.-Q., T. Takenaka, and T. Tanaka, "Image reconstruction of two-dimensional impenetrable objects using genetic algorithm," *Journal of Electromagnetic Waves and Applications*, Vol. 13, No. 1, 95–118, 1999.
6. Sijher, T. S. and A. A. Kishk, "Antenna modeling by infinitesimal dipoles using genetic algorithms," *Progress In Electromagnetics Research*, PIER 52, 225–254, 2005.
7. Tu, T.-C. and C.-C. Chiu, "Path loss reduction in an urban area by genetic algorithm," *Journal of Electromagnetic Waves and Applications*, Vol. 20, No. 3, 319–330, 2006.
8. Lu, Y.-Q. and J.-Y. Li, "Optimization of broadband top-load antenna using micro-genetic algorithm," *Journal of Electromagnetic Waves and Applications*, Vol. 20, No. 6, 793–801, 2006.
9. Chen, X., D. Liang, and K. Huang, "Microwave imaging 3-D buried objects using parallel genetic algorithm combined with FDTD technique," *Journal of Electromagnetic Waves and Applications*, Vol. 20, No. 13, 1761–1774, 2006.
10. Rawlins, G. J. E., L. D. Whitley, M. D. Vose, R. K. Belew, W. Banzhaf, C. Reeves, W. N. Martin, and W. M. Spears, *Foundations of Genetic Algorithms*, (1-6), 1991–2001, Morgan Kaufmann Publishers.
11. Johnson, J. M. and Y. Rahmat-Samii, "Genetic algorithms in engineering electromagnetics," *IEEE Antennas and Propagation Magazine*, Vol. 39, No. 4, 7–25, 1997.
12. Rahmat-Samii, Y., "Genetic algorithms in modern electromagnetics: concept, implementation and applications," *Tutorial Workshop, 2000 International Symposium on Antennas and Propagation*, 2000.
13. Meng, Z.-Q., T. Takenaka, and S. He, "A genetic algorithm with an adaptive chromosome structure for reconstruction of radome parameters using a Gaussian beam," *Microwave and Optical Technology Letters*, Vol. 25, No. 5, 323–327, 2000.
14. Meng, Z.-Q. and H. Misaka, "Fast genetic algorithm for optimization of inverse scattering problem," *IEEJ Trans.*, Vol. 122, No. 12, 1005–1010, 2002 (in Japanese).
15. Chew, W. C., *Waves and Fields in Inhomogeneous Media*, Van Nostrand Reinhold, New York, 1990.