Optimum Population Size and Mutation Rate for a Simple Real Genetic Algorithm that Optimizes Array Factors

Randy L. Haupt
Utah State University
Electrical and Computer Engineering
4120 Old Main Hill
Logan, UT 84322-4120
haupt@ieee.org

Introduction

There has been an explosion of papers describing applications of a genetic algorithm (GA) to electromagnetics problems [1]. Most of the work has followed traditional GA philosophy when choosing the population size and mutation rate of the genetic algorithm. This paper reports the results of experiments to determine the optimum population size and mutation rate for a simple real genetic algorithm. The choice of population size and mutation rate can cause the run time of the GA to vary by several orders of magnitude. The results of this investigation show that a small population size and relatively large mutation rate is far superior to the large population sizes and low mutation rates that is used by most of the papers presented in the electromagnetics community and by the GA community at large.

Parameter Selection for a Simple Genetic Algorithm

The GA used in this paper works with continuous parameters and uses a roulette wheel proportional weighting selection and the single point crossover using the method advocated in [2]. Determining the optimum population size and mutation rate must take into account the random components of the GA.

The first intensive study of genetic algorithm parameters was done by De Jong [3] and is nicely summarized in Goldberg [4]. De Jong found that a small population size improved initial performance while large population size improved long-term performance and a high mutation rate was good for off-line performance while low mutation rate was good for on-line performance. Grefenstette [5] used a meta GA. He found the best GA had a population size of 30 and mutation rate of 0.01. Schaffer found a population size =20 to 30 and a mutation rate = 0.005 to 0.01 are best. In [2] both binary and continuous parameter GAs, a small population size allowed to evolve for many generations produced the best results. Similar sensitivity studies with mutation rate suggested that mutation rates in the range of 0.05 to 0.35 found the best minima.

Problem Fromulation

The goal of the optimization is to find the weighting for a linear array that produces the minimum maximum sidelobe level. The objective function is

$$f = \max\ sidelobe\ of \left\{ \sum_{n=1}^{N} a_n e^{jp_n} e^{j(n-.5)kdu} \right\}$$

0-7803-6369-8/00/$10.00 ©2000 IEEE

where

N = number of array elements
a = vector of amplitude weights
p = vector of phase weights
$k = 2\pi$/wavelength
d = element spacing
u = angle variable

In the cases presented, only a or p are optimized but not both in the same GA run. The array factor is calculated from broadside to endfire, and a search is performed to find all the peaks. The highest sidelobe level is returned with the function call.

Results

The first example uses a GA to find the optimum amplitude taper for an 18 element uniformly spaced array ($d = 0.5\lambda$). The taper is symmetric about the center of the array and the two center elements have an amplitude of one. Whenever the minimum maximum sidelobe level is less than 25 dB below the peak of the main beam or the number of function calls exceeds 50,000, the algorithm stops. The GA was run 20 independent times and the results were averaged for population sizes of 4, 8, …, 64 and mutation rates of .01, .02,…, .4. Figure 1 displays a plot of the average number of function calls vs. population size and mutation rate when the results were averaged over 20 independent runs. This graph is very low when the population size and mutation rate is less than 20%, except for a subregion where the population size and mutation rate is small.

A strong region of performance in the plot in Figure 1 is the region between a population size of 4 to 16 and a mutation rate of 0.1 to 0.2. Figure 2 shows this region when the GA is averaged over 50 runs. The plot shows a population size of 8 or less and a mutation rate of 13% or less produce excellent results.

In order to become more confident with the results presented in the previous figures, the GA was averaged over 500 runs for several different mutation rates and population sizes as shown in Table 1. Results (in number of function calls) from running a GA 200 times to find the optimum amplitude taper for an 18 element array that minimizes the maximum sidelobe level. A single GA run stopped when the sidelobe level went below –25dB or the number of function calls exceeded 50,000. The minimum and maximum number of function calls over the 200 runs as well as the mean, and standard deviation of the number of function calls for the 200 runs are shown here. A population size of 4 with a mutation rate of 15% produced the best average result. Next best was the population size of 8 and mutation rate of 15% then mutation rate of 20%. These results are consistent with those in Figure 1.

The next example finds a low sidelobe taper for a linear array. Table 2 shows the results (in number of function calls) from running a GA 100 times to find the optimum phase taper that minimizes the maximum sidelobe level of a 20 element array. A single GA run stopped when the sidelobe level went below −14dB or the number of function calls exceeded 50,000. The minimum and maximum number of function calls over the 100 runs as well as the mean, and standard deviation of the number of function calls for the 100 runs are shown here.

Table 3 shows the results (in dB) from running a GA for 100,000 function evaluations in order to find the optimum amplitude taper for a 20 element array that minimizes the maximum sidelobe level. The minimum and maximum results as well as the mean, and standard deviation of the best sidelobe levels for the 100 runs are shown. Population sizes of 4 and 8 with 15% mutation rate outperformed the GA's with population sizes of 64 and 128 with a mutation rate of 2%.

Conclusions

The results of the numerical experiments presented in this paper suggest that the best mutation rate for GA's lies between 5 and 20% while the population size should be less than 16.

Bibliography

[1] Y. Rahmat-Samii and E. Michielssen, Electromagnetic Optimization by Genetic Algorithms, New York: John Wiley & Sons, 1999..

[2] R.L. Haupt and S.E. Haupt, *Practical Genetic Algorithms*, New York: John Wiley & Sons, 1998.

[3] K. A. De Jong, "Analysis of the Behavior of a Class of Genetic Adaptive Systems," Ph.D. Dissertation, The University of Michigan, Ann Arbor, MI, 1975.

[4] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Reading, MA: Addison-Wesley, 1989.

[5] J. J. Grefenstette, "Optimization of control parameters for genetic algorithms," IEEE Trans. on Systems, Man, and Cybernetics 16, Jan/Feb 86, p. 128.

[6] J. D. Schaffer, et. al., "A study of control parameters affecting online performance of genetic algorithms for function optimization," in *Proceedings of the Third International Conference on Genetic Algorithms*, ed. J. D. Schaffer, Los Altos, CA: Morgan Kaufmann, June 4-7, 1989, pp. 51-60.

Table 1. Results (in number of function calls) from running a GA 200 times to find the optimum amplitude taper for an 18 element array that minimizes the maximum sidelobe level.

Run	Mutation rate	Population size	min	max	mean	std dev
1	0.15	4	26	3114	398	455
2	0.20	4	110	50002	7479	12798
3	0.15	8	60	2457	461	332
4	0.20	8	49	2624	654	466
5	0.01	64	300	50031	1158	3498
6	0.02	64	277	11818	1028	911
7	0.01	128	393	2535	1410	365
8	0.02	128	1215	50071	10208	16077

Table 2. Results (in number of function calls) from running a GA 100 times.

Run	Mutation rate	Population size	min	max	mean	std dev
1	0.15	4	134	50002	2973	5856
2	0.20	4	163	50000	5232	9744
3	0.15	8	168	8223	1827	1510
4	0.20	8	124	21307	3220	3604
5	0.01	64	614	50024	7914	15040
6	0.02	64	546	50036	6624	13130
7	0.01	128	955	50043	4791	9708
8	0.02	128	933	50033	3942	7636

Table 3. Results (in dB) from running a GA for 100,000 function evaluations in order to find the optimum amplitude taper for a 20 element array that minimizes the maximum sidelobe level.

Run	Mutation rate	Population size	min	max	mean	std dev
1	0.15	4	-57.5	-28.4	-36.1	4.6
3	0.15	8	-46.0	-29.5	-36.5	3.3
6	0.02	64	-42.5	-27.1	-32.5	3.3
8	0.02	128	-41.2	-28.0	-32.5	2.5

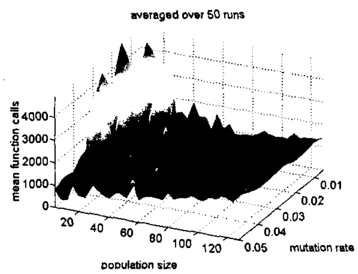

Figure 1. The GA performed best when the population size was small and the mutation rate around 10%.

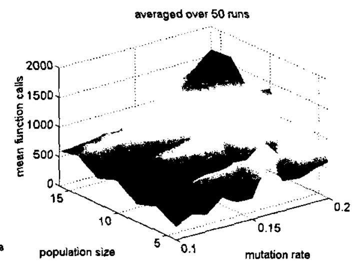

Figure 2. This graph shows that the small population size and relatively large mutation rate causes the GA to find an answer in the fewest number of function evaluations.