Introduction to Genetic Algorithms in Electromagnetics

Randy L. Haupt
Utah State University
Electrical and Computer Engineering
4120 Old Main Hill
Logan, UT 84322-4120
Haupt@ieee.org
435-797-2841

This special issue of the ACES Journal is devoted to new developments in Genetic Algorithm (GA) applications in computational electromagnetics. Genetic Algorithms have become extremely popular in the computational electromagnetics literature. The papers included in this special issue are very arcane, so I decided to include an unreviewed tutorial overview at the last minute as an introduction for those of you who are at a more basic level. GAs model natural selection and genetics on a computer to optimize a wide range of problems. Some of the advantages of a genetic algorithm include that it

- Optimizes with continuous or discrete parameters,
- Doesn't require derivative information,
- Simultaneously searches from a wide sampling of the cost surface,
- Deals with a large number of parameters,
- Is well suited for parallel computers,
- Optimizes parameters with extremely complex cost surfaces,
- Provides a list of semi-optimum parameters, not just a single solution,
- May encode the parameters so that the optimization is done with the encoded parameters, and
- Works with numerically generated data, experimental data, or analytical functions.

These advantages have inspired many people working in computational electromagnetics. For a nice historical development of applications of genetic algorithms in electromagnetics, see [1].

A genetic algorithm is relatively simple compared to many of the local optimizers used. As an example, consider the very simple MATLAB code presented in [2]:

```
% This is a simple binary GA

N=8;      % # bits in a chromosome
M=16;     % # chromosomes
last=20;  % # generations
M2=M/2;

% creates initial population
chromo=round(rand(M,N));

for ib=1:last

   ************************
   * insert subroutine to calculate
   * objective function output
   *   cost=function(chromo)
   *   cost is a Nx1 array
   ************************

   % ranks results and chromosomes
```

```
    [cost,ind]=sort(cost);
    chromo=chromo(ind(1:M2),:);

    %mate
    cr=ceil((N-1)*rand(M2,1));

    % pairs chromosomes
    % performs crossover
    for ic=1:2:M2
chromo(M2+ic,1:cr)=chromo(ic,1:cr);
chromo(M2+ic,cr+1:N)=chromo(ic+1,cr+1:N);
chromo(M2+ic+1,1:cr)=chromo(ic+1,1:cr);
chromo(M2+ic+1,cr+1:N)=chromo(ic,cr+1:N);
    end

    %mutate
    ix=ceil(M*rand);
    iy=ceil(N*rand);
    chromo(ix,iy)=1-chromo(ix,iy);

end %last
```

This small code has inspired many people to try genetic algorithms and is given to students taking a computational electromagnetics course at Utah State University. If you have never tried a GA, then this one is a good starter program.

Figure 1 shows the components of a GA. Compare this approach to a typical line search approach shown in Figure 2. The GA usually loses to a local optimizing line search in a race to the bottom of a bowl. On the other hand, the GA has the ability to jump out of a bowl into another bowl within the search area whereas a line search is much more constrained. Often times a local optimizer is worth using after a GA finds the bowl containing the desired minimum.

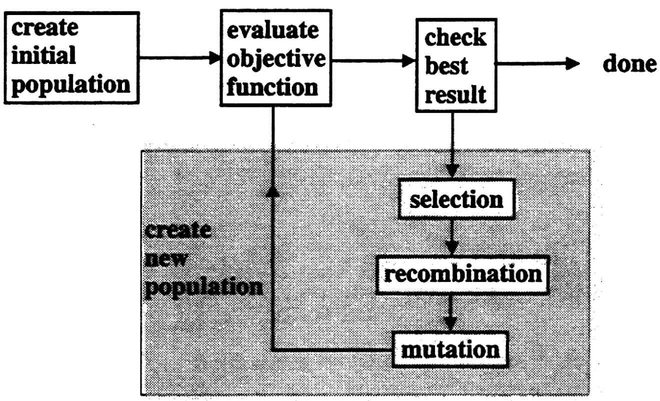

Figure 1. Flow chart of a genetic algorithm. Numerical simulation of genetics and evolution occurs in the gray box.

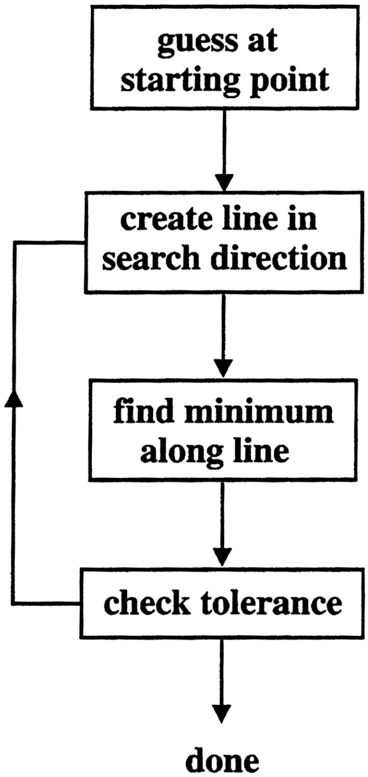

Figure 2. Flow chart of a typical line search optimizer.

A GA can work with either continuous parameters or binary encodings of the continuous parameters. In some cases, the parameters are naturally binary. In either case, the GA begins by creating a random set of parameters called a population. Each member of the population is a chromosome and contains all the information necessary as an input to an objective function that creates an output of interest. This first part is a random search. Next, the algorithm enters the gray box in Figure 2. Here, parents are selected to generate offspring by taking part(s) of one chromosome parent selected and combining with part(s) of one or more other parents. Natural selection occurs by weighting the probability of a chromosome being selected as a parent in proportion to its fitness. Also, inferior solutions or chromosomes with low fitness values are usually discarded from the population. Finally, random mutations are introduced to the population by randomly changing parameter values or bits in the binary encoding.

For the reader interested in pursuing introductory material on genetic and evolutionary programming, see the nice articles by Fogel [3] and Holland [4]. Goldberg has been a leader in the field and his book [5] is an excellent overview. For a practical introduction with a more tutorial, handholding approach to writing and using GAs see [6].

[1] D.S. Weile and E. Michielssen, "Genetic algorithm optimization applied to electromagnetics: a review," IEEE AP-S Trans., Vol. 45, No. 3, Mar 97, pp. 343-353.
[2] R.L Haupt, "An introduction to genetic algorithms for electromagnetics," IEEE Antennas and Propagation Magazine, Vol. 37, No. 2, Apr 95.
[3] D.B. Fogel, "Evolutionary computing," IEEE Spectrum, Vol. 37, No. 2, Feb 00, pp. 26-32.
[4] J.H. Holland, "Genetic algorithms," Sci. Am., Jul 92, pp. 66-72.
[5] D.E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Reading, MA: Addison-Wesley, 1989.
[6] R.L. Haupt and S.E. Haupt, *Practical Genetic Algorithms*, New York: John Wiley & Sons, 1998.