P1.7 GENETIC ALGORITHMS IN GEOPHYSICAL FLUID DYNAMICS

Sue Ellen Haupt *
Utah State University, Logan, UT

1. INTRODUCTION

The genetic algorithm (GA) is an optimization tool that has shown great success at solving problems not amenable to easy solution via more traditional means (such as the traveling salesman problem, solved by Koza 1992). Since most CFD problems have not been traditionally posed in terms of optimization, GAs have not yet been widely used in this field. The goal of this work is to demonstrate the ability of the GA to provide interesting solutions to problems difficult to solve in more traditional ways. One need only to be a bit creative in posing the problem as one in optimization. For instance, boundary value problems can be seen as minimizing the discretized version of the magnitude of a partial differential equation (PDE). Similarly, inversion problems often involve finding the best fit parameters to an assumed version of a model. For nonlinear model forms, it is often difficult to analytically minimize the mean square difference between model and data. These are cases where the GA has potential to help in making strides forward.

2. CONTINUOUS PARAMETER GENETIC ALGORITHM

The flow chart in Figure 1 provides a “big picture" overview of a continuous genetic algorithm, that is, one in which the parameters are real numbers. The parameters are the genes which are strung together in a one-dimensional array known as a chromosome. The GA begins with a population of chromosomes which are fed to the cost function for evaluation. The fittest chromosomes survive while the highest cost ones die off. This process mimics natural selection in the natural world. The lowest cost survivors mate. The mating process combines information from the two parents to produce two offspring. Some of the population experiences mutations. The mating and mutation operators introduce new chromosomes which may have a lower cost than the prior generation. The process iterates until an acceptable solution is found. More specific description of the process follows.

Creating the Population

The first step of a continuous parameter genetic algorithm is creating the population of chromosomes. First, the real parameters are concatenated together into a chromosome as:

$$chromosome = \lfloor p_1 p_2 \cdots p_a \cdots p_{N_{par}} \rfloor \qquad (1)$$

where the p_i are the parameters and there are a total of N_{par} parameters. The parameters are simply floating point numbers. The encoding function needs only keep track of which digits represent which parameters and to make sure they are within given bounds. A population of such chromosomes is created using a random number generator so that the chromosome arrays are gathered together in a two dimensional matrix.

Once the chromosomes have been created, their cost or fitness must be evaluated. This is done by the cost or objective function, which is very problem specific. The lowest cost chromosomes (N_{keep}) remain in the population while the higher ones are deemed less fit and die off. The reduced population is then the portion of the population available for mating.

Choosing the Mates

There are a variety of methods to pair the chromosomes for mating. Some popular methods are reviewed by Haupt and Haupt (1998). Here, we choose to pair the chromosomes according to numerical rank. After the cost function evaluation, the

---

* Corresponding author address: Sue Ellen Haupt, Department of Mechanical and Aerospace Engineering, 4130, Utah State University, Logan, UT 84322-4130; e-mail: suehaupt@ece.usu.edu

chromosomes are sorted in order from lowest cost to highest. That is, the nth chromosome will have a probability of mating of:

$$P_n = \frac{N_{keep} - n + 1}{\sum_{k=1}^{N_{keep}} k} \qquad (2)$$

Then the cumulative probabilities are used for selecting which chromosomes mate.

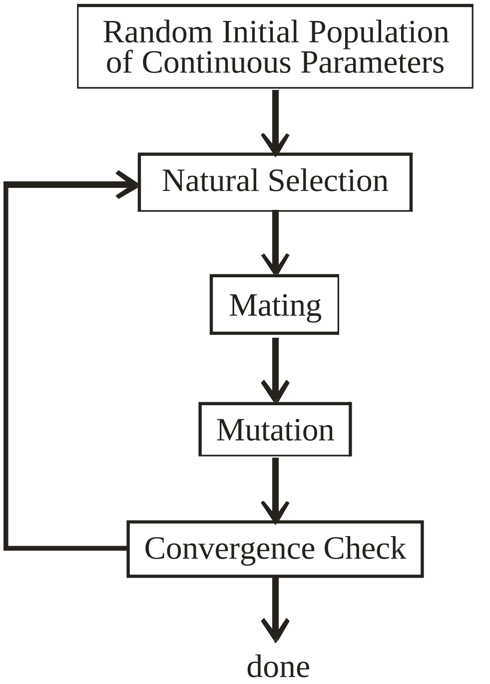

Figure 1. Flow chart of a continuous genetic algorithm.

Crossover

Once two parents are chosen, some method must be devised to produce offspring which are some combination of these parents. Many different approaches have been tried for crossing over in continuous parameter genetic algorithms. Adewuya (1996) reviews some of the current methods thoroughly. Several interesting methods are demonstrated by Michalewicz (1992). Haupt and Haupt (1998) gives an overview of some popular methods.

The method used here is a combination of an extrapolation method with a crossover method. We wanted to find a way to closely mimic the advantages of the binary genetic algorithm mating scheme. It begins by randomly selecting a parameter in the first pair of parents to be the crossover point.

$$a = roundup\{random \times N_{par}\} \qquad (3)$$

We'll let

$$\begin{aligned} parent_1 &= \lfloor p_{m1} p_{m2} \cdots p_{ma} \cdots p_{mN_{par}} \rfloor \\ parent_2 &= \left[p_{d1} p_{d2} \cdots p_{da} \cdots p_{dN_{par}} \right] \end{aligned} \qquad (4)$$

where the *m* and *d* subscripts discriminate between the *mom* and the *dad* parent. Then the selected parameters are combined to form new parameters that will appear in the children:

$$\begin{aligned} p_{new1} &= p_{ma} - b\left[p_{ma} - p_{da}\right] \\ p_{new2} &= p_{da} - b\left[p_{ma} - p_{da}\right] \end{aligned} \qquad (5)$$

where b is also a random value between 0 and 1. The final step is to complete the crossover with the rest of the chromosome as before:

$$\begin{aligned} offspring_1 &= \lfloor p_{m1} p_{m2} \cdots p_{new1} \cdots p_{dN_{par}} \rfloor \\ offspring_2 &= \left[p_{d1} p_{d2} \cdots p_{new2} \cdots p_{mN_{par}} \right] \end{aligned} \qquad (6)$$

If the first parameter of the chromosomes is selected, then only the parameters to right of the selected parameter are swapped. If the last parameter of the chromosomes is selected, then only the parameters to the left of the selected parameter are swapped. This method does not allow offspring parameters outside the bounds set by the parent unless b is greater than one. In this way, information from the two parent chromosomes is combined a way that mimics the crossover process during meiosis.

Mutations

If care is not taken, the genetic algorithm converges too quickly into one region of the cost surface. If this area is in the region of the global minimum, that is good. However, some functions have many local minima and the algorithm could get stuck in a local well. If we do nothing to solve this tendency to converge quickly, we could end up in a local rather than a global minimum. To avoid this problem of overly fast convergence, we force the routine to explore other areas of the cost surface by randomly introducing changes, or mutations, in some of the parameters. A mutated parameter is replaced by a new random parameter.

3. The Super Korteweg de Vries Equation

Above, we described the continuous parameter genetic algorithm. The piece yet missing is the cost function or objective function which determines the fitness of the chromosome. The cost function is very problem dependent. Here, we will look at solutions to nonlinear boundary value problems in fluid dynamics. The key is to formulate the objective, or cost function that the GA uses to determine the state of the optimization as the solution of the partial differential equation. The procedure consists of first expanding the independent variables in terms of orthogonal basis functions, performing a Galerkin projection, and then requiring that the result be minimized over a given set of points.

When the equation is nonlinear, finding a solution suddenly becomes extremely difficult. Although a few nonlinear PDEs can be solved analytically, we often must rely on numerical methods to determine an approximate solution. One class of PDEs of particular interest to engineers and scientists is the solitary wave. Solitary waves, or solitons, are permanent form waves for which the nonlinearity balances the dispersion to produce a coherent structure. They appear as models of many coherent phenomena ranging from propagation patterns in optical cables to the Great Red Spot of Jupiter. Here, we are interested in demonstrating that a genetic algorithm is a useful technique for solving a highly nonlinear differential equation that is formally nonintegrable.

The equation to be solved here the Super Korteweg de Vries (SKDV) Equation:

$$u_t = auu_x + mu_{xxx} - uu_{xxxxx} = 0 \qquad (7)$$

where the functional form is denoted by u, time derivative by the t subscript, spatial derivative by the x subscript, and a , m , and u are parameters of the problem. It is a model for several physical phenomena, including shallow water waves near a critical value of surface tension, magneto-acoustic waves propagating at an angle to an external magnetic field, and waves in a nonlinear LC circuit with mutual inductance between neighboring inductors Yoshimura and Watanabe (1982). We wish to solve for waves that are steadily translating so we write the t variation using a Galilean transformation, $X=x-ct$ where c is the phase speed of the wave. Thus, our SKDV becomes a fifth order, nonlinear ordinary differential equation:

$$(au-c)u_X + muu_{XXX} - uu_{XXXXX} = 0 \qquad (8)$$

Boyd (1986) extensively studied methods of solving this equation. He expanded the solution in terms of Fourier series to find periodic cnoidal wave solutions (solitons that are repeated periodically). Among the methods used are the analytical Stokes' expansion, which intrinsically assumes small amplitude waves, and the numerical Newton-Kantorovich iterative method, which can go beyond the small amplitude regime if care is taken to provide a very good first guess. This method is based on the Newton iteration and requires an analytical expansion of the equation into a mean plus a perturbation. The equation is solved for the perturbation and then discretized. At each step of the iteration, the perturbation is found and added to the prior solution. This method assumes that the perturbation is small, thus requiring a good "first guess" to the solution.

Haupt and Boyd (1988) were able to extend these methods to deal with resonance conditions. These methods were generalized to two dimensions to find double cnoidal waves (two waves of differing wave number on each period) for the integrable Korteweg de Vries equation (Haupt and Boyd 1988) and the nonintegrable Regularized Long Wave Equation (Haupt 1988). However, these methods require careful analytics and programming that is very problem-specific. Here, we use a genetic algorithm to obtain a similar result with considerably less effort.

Normally, we don't think of PDEs as minimization problems. However, if we want to find values that force a differential equation to zero (a form in which we can always cast the system), we can look for the minimum of its absolute value. Koza (1992) demonstrated that a genetic algorithm could solve a simple differential equation by minimizing the value of the solution at 200 points. To do this, he numerically differentiated at each point and fit the appropriate solution. For this fifth order equation, this is not so easy. Therefore we combine the genetic algorithm with a spectral expansion which eases the differentiation.

To find the solution of (8), we expand the function u in terms of a Fourier cosine series:

$$u(X) \cong u_K(X) = \sum_{k=1}^{K} a_k \cos(kx) \qquad (9)$$

Without loss of generality, we have assumed that the function is symmetric about the *x*-axis by not including sine functions. In addition, we use the "cnoidal convention'" by assuming that the constant is 0. Now, we can easily take derivatives as powers of the wave numbers to write the cost that we wish to minimize as:

$$\mathrm{cost}(u_K) = \sum_{k=1}^{K} \left[-k(au - c) + k^3 m + k^5 u \right] a_k \sin(kx) \qquad (10)$$

Equation (10) is reasonably easy to put into the cost function of a continuous genetic algorithm. We wish to find the coefficients of the series, a_k. The only minor complication is computing *u* in the nonlinear term to insert into the cost function (10). However, this is merely one extra line of code. We just use the spectral expansion of the current chromosome.

The parameters that we used here are $u = 1$, $m = 0$, $a = 1$, and a phase speed of *c*=14.683 to match with a known highly nonlinear solution. Note that the phase speed and amplitude of solitary type waves are interdependent. We could instead have specified the amplitude and solved for the phase speed. It is equivalent. We computed the coefficients, a_k, to find the best cnoidal wave solution for *K=6*. We used $N_{ipop} = 500$, $N_{pop} = 100$, mutation rate of 0.2, and 70 iterations. We evaluated the cost function at merely 2 points for this run and summed their absolute value. The results appear in Figure 2. The solid line is the "exact" solution reported by Boyd (1986) and the dashed line is the genetic algorithm's approximation to it. They are barely distinguishable. For interest's sake we show a genetic algorithm solution that converged to a double cnoidal wave as Figure 3. This solution was found for a total of 6 basis functions, variable phase speed (computed c=-9.6436), 100 iterations, population size of 32, and mutation rate of 0.25. These double cnoidal wave solutions would require a two-dimensional expansion involving two different phase speeds in the Newton-Kantorovich method. Here, we were able to find them using the same basic cost function as for the single cnoidal wave.

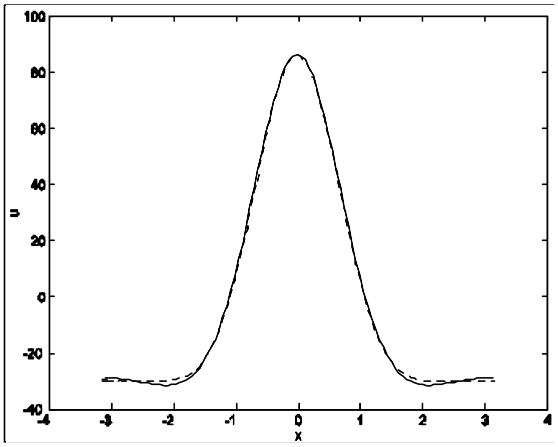

Figure 2. A cnoidal wave of the SKDV with phase speed of *c*=14.683. The solid line is an "exact" solution due to Boyd (1986) and the dashed line is the GA solution which approximates it.

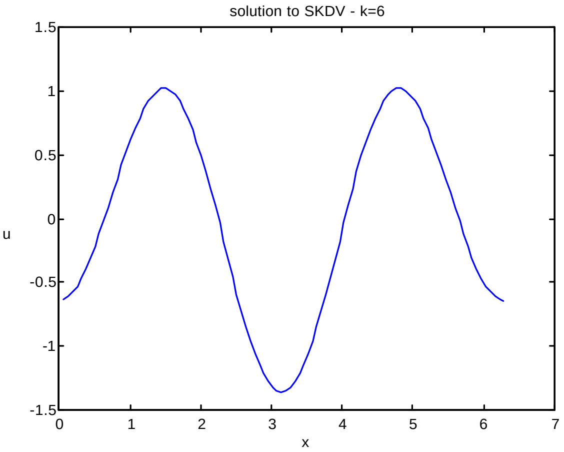

Figure 3. Double cnoidal wave of the SKDV as found by the genetic algorithm.

4. EMPIRICAL MODELING WITH A GENETIC ALGORITHM

A second use of GAs is to solve for solutions of nonlinear inversion problems. For example, empirical models have gained popularity in recent years as an alternative to the more traditional dynamical models (for example see Hasselman 1988, Penland 1989, Branstator and Haupt 1998). Linear empirical models are easy to produce from data using standard least squares inversion techniques. However, nonlinear models are more difficult to devise due to the introduction of high order tensors to the problems. GAs can resolve this issue through redefining the problem in terms of optimization and directly searching for the propagator matrix given the data. The use of the binary GA for a simple inverse problem is demonstrated in the companion paper, Haupt and Haupt (2003). Here, this technique is demonstrated for matching low order nonlinear systems, such as the Lorenz equations (Lorenz 1963) which can be written:

$$\begin{aligned} \dot{x} &= -sx + sy \\ \dot{y} &= rx - y - xz \\ \dot{z} &= -bz + xy \end{aligned} \qquad (11)$$

where *x, y, z* are the lowest order coefficients of a truncated series of atmospheric flow and we use parameters: $s = 10, b = 8/3, r = 28$ These parameters produce a chaotic regime that results in a strange attractor. The equations (11) were integrated using a fourth order Runge-Kutta method to produce the data in Figure 4.

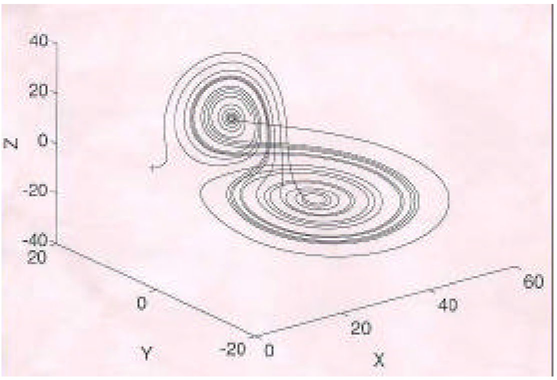

Figure 4. A Lorenz attractor computed by integrating equations (11) in time for 2000 steps.

We wish to create an empirical model of these data. Since the model is highly nonlinear, we choose to fit a nonlinear model of the form:

$$\dot{x} = Bx + Cx^T x + x \qquad (12)$$

where the variables have been incorporated into the vector x, *B* is an *N x N* matrix that serves as the linear propagator, *C* is an *N x N x N* third order tensor that gives the coefficients of the quadratic interactions, and x is the constant noise vector. We wish to compute the matrices *B* and *C* so that the least square error of (12),

$$E = \left\langle \left(\dot{x} - Bx - Cx^T x \right)^2 \right\rangle \qquad (13)$$

is minimized. The angle brackets denote a time average. Minimizing *E* with respect to *B* and *C* gives the system of equations:

$$\begin{aligned} T_3 + CT_4 &= 0 \\ B &= \langle x\dot{x} \rangle - C \langle xxx \rangle \langle xx \rangle^{-1} \end{aligned} \qquad (14)$$

where:

$$\begin{aligned} T_3 &= \langle xx\dot{x} \rangle - \langle x\dot{x} \rangle \langle xx \rangle^{-1} \langle xxx \rangle \\ T_4 &= \langle xxx \rangle \langle xx \rangle^{-1} \langle xxx \rangle - \langle xxxx \rangle \end{aligned} \qquad (15)$$

Although this is a closed form solution, to compute the third order tensor *C* requires inverting the fourth order tensor T_4. Such an inversion is not trivial. Therefore, we choose to instead compute C in equation (14) by doing a best fit with a genetic algorithm.

The parameters of the GA are an initial population of 500, working population of 100, mutation rate of 0.3 for a total of 200 generations. Taking into account symmetries for this problem results in 18 unique parameters to find. For this highly nonlinear regime, we were able to find a solution, which when propagated via equation (12), produces the time evolution depicted in Figure 5. Although the match is not perfect, we have replicated the general shape of the strange attractor and the size of the domain is approximately correct.

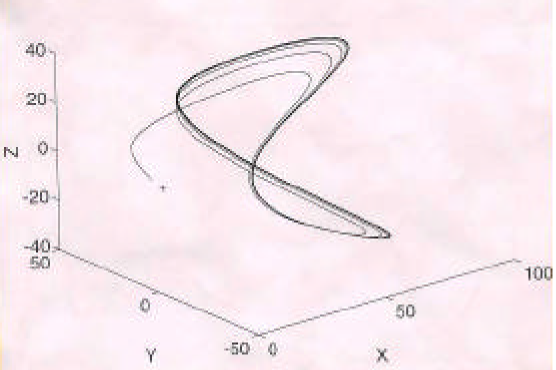

Figure 5. Nonlinear model of Lorenz attractor (equation 12) as computed with a GA.

For comparison, the solution is compared to a linear model fit, which is merely the linear part of (12). For this portion, there is a simple closed form solution that is easily computed. The linear match is shown in Figure 6. We see that the linear model is not able to capture the shape of the attractor, but instead shows a decaying spiral behavior.

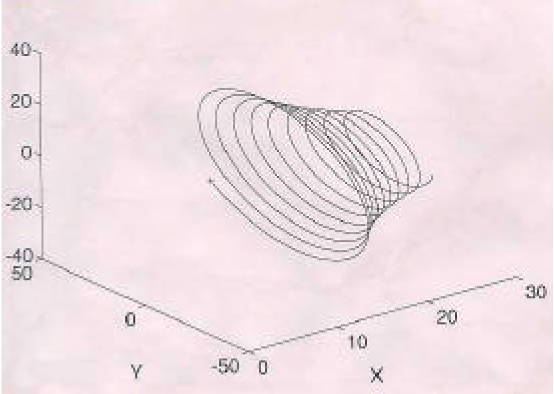

Figure 5. Linear model of the Lorenz equations.

5. CONCLUSIONS

Genetic algorithms show promise for optimization problems in complex systems. Here we have given two examples of problems where they may prove useful in the future. In the first problem, the genetic algorithm was used to find the coefficients of a Fourier series that best models a nonlinear wave equation. The equation was expanded in Fourier series and the phase speed and amplitudes were fit. The GA was quite good at reproducing an “exact” solution to the cnoidal wave solution. In addition, it was able to find a double cnoidal wave solution that would have involved major effort using most other techniques.

The GA can also fit empirical models to nonlinear systems much better than can the usual linear least squares fit. The GA reproduced the shape of the attractor to the Lorenz equations reasonably well given the utter failure of the linearized method.

Genetic algorithms are not perfect. One cannot predict their convergence properties well in advance. However, they show great ability at not only finding the best minimum solution, but also find multiple minima where other methods fail.

The utility of the GA is only beginning to be tapped. Its usefulness is only limited by the creativity of the scientists who use them.

REFERENCES

Adewuya, A.A., 1996. New Methods in Genetic Search with Real-Valued Chromosomes, Master’s Thesis, Cambridge, Massachusetts Institute of Technology.

Boyd, J.B., 1986: Solitons from Sine Waves: Analytical and Numerical Methods for Nonintegrable Solitary and Cnoidal Waves, *Physica* **21D**, pp. 227-246.

Branstator, G. and S.E. Haupt, 1998: An empirical model of forced barotropic atmospheric anomalies, *J. Climate*, **11**, 2645-2667.

Hasselman, K., 1988: PIPs and POPs: The reduction of complex dynamical systems using principal interaction and oscillation patterns. *J. Geophys. Res.*, **93**, 11015-11021.

Haupt, R.L. and S.E. Haupt, 1998: *Practical Genetic Algorithms*, New York: John Wiley & Sons, 177 pp.

Haupt, S.E. and J.P.Boyd, 1988: Modeling Nonlinear Resonance: A Modification to the Stokes' Perturbation Expansion, *Wave Motion*, **10**, 1988, pp. 83-98

Haupt, S.E. and J.P. Boyd, 1988: “Double Cnoidal Waves of the Korteweg De Vries Equation: Solution by the Spectral Boundary Value Approach,” *Physica D*, **50**, 117-134.

Haupt, S.E., 1988: Solving Nonlinear Wave Problems with Spectral Boundary Value Techniques, Ph.D. Dissertation, University of Michigan, 157pp.

Haupt, S.E. and R.L. Haupt, 2003: Genetic Algorithms in the Environmental Sciences, AMS Conference on Artificial Intelligence, Paper 1.1A.

Koza, J.R., 1992: Chapter 10: The Genetic Programming Paradigm: Genetically Breeding Populations of Computer Programs to Solve Problems, in *Dynamic, Genetic, and Chaotic Programming*, The Sixth Generation, B. Soucek, ed., New York: John Wiley & Sons, Inc., pp.203-321.

Lorenz, E.N., 1963: Deterministic Nonperiodic Flow, *J. Atmos. Sciences*, **20**, 130-141.

Penland, C., 1989: Random forcing and forecasting using principal oscillation pattern analysis. *Mon. Weather Rev*., **117**, 2165-2185.

Yoshimura, K. and S. Watanabe, 1982: Chaotic behavior of nonlinear evolution equation with fifth order dispersion, *J. Phys. Soc. Japan*, **51**, 1982, pp. 3028-3035.