Genetic Algorithms in Electromagnetics

Randy L. Haupt
Applied Research Laboratory
Pennsylvania State University

Douglas H. Werner
Department of Electrical Engineering
Applied Research Laboratory
Materials Research Institute
Pennsylvania State University

IEEE PRESS

BICENTENNIAL 1807 WILEY 2007 BICENTENNIAL

WILEY-INTERSCIENCE
A John Wiley & Sons, Inc., Publication

Genetic Algorithms in Electromagnetics

BICENTENNIAL 1807 WILEY 2007 BICENTENNIAL

THE WILEY BICENTENNIAL–KNOWLEDGE FOR GENERATIONS

Each generation has its unique needs and aspirations. When Charles Wiley first opened his small printing shop in lower Manhattan in 1807, it was a generation of boundless potential searching for an identity. And we were there, helping to define a new American literary tradition. Over half a century later, in the midst of the Second Industrial Revolution, it was a generation focused on building the future. Once again, we were there, supplying the critical scientific, technical, and engineering knowledge that helped frame the world. Throughout the 20th Century, and into the new millennium, nations began to reach out beyond their own borders and a new international community was born. Wiley was there, expanding its operations around the world to enable a global exchange of ideas, opinions, and know-how.

For 200 years, Wiley has been an integral part of each generation's journey, enabling the flow of information and understanding necessary to meet their needs and fulfill their aspirations. Today, bold new technologies are changing the way we live and learn. Wiley will be there, providing you the must-have knowledge you need to imagine new worlds, new possibilities, and new opportunities.

Generations come and go, but you can always count on Wiley to provide you the knowledge you need, when and where you need it!

William J. Pesce
PRESIDENT AND CHIEF EXECUTIVE OFFICER

Peter Booth Wiley
CHAIRMAN OF THE BOARD

Genetic Algorithms in Electromagnetics

Randy L. Haupt
Applied Research Laboratory
Pennsylvania State University

Douglas H. Werner
Department of Electrical Engineering
Applied Research Laboratory
Materials Research Institute
Pennsylvania State University

IEEE PRESS

BICENTENNIAL 1807 WILEY 2007 BICENTENNIAL

WILEY-INTERSCIENCE
A John Wiley & Sons, Inc., Publication

Published by John Wiley & Sons, Inc., Hoboken, New Jersey.
Published simultaneously in Canada.

Wiley Bicentennial Logo: Richard J. Pacifico

Library of Congress Cataloging-in-Publication Data:

Haupt, Randy L.
Genetic algorithms in electromagnetics / by Randy L. Haupt and Douglas H. Werner.
p. cm.
A Wiley-Interscience publication.
Includes bibliographical references and index.
ISBN: 978-0-471-48889-7
1. Antenna arrays—Design. 2. Electromagnetism—Mathematical models. 3. Genetic algorithms. I. Werner, Douglas H., 1960– II. Title.
TK7871.6.H37 2007
621.30285'631—dc22

2006046738

Printed in the United States of America.

10 9 8 7 6 5 4 3 2 1

We dedicate this book to our wives,
Sue Ellen Haupt and Pingjuan L. Werner.

Contents

Preface

Most books on electromagnetics describe how to solve particular problems using classical analysis techniques and/or numerical methods. These books help in formulating the objective function that is used in this book. The objective function is the computer algorithm, analytical model, or experimental result that describes the performance of an electromagnetic system. This book focuses primarily on the optimization of these objective functions. Genetic algorithms (GAs) have proved to be tenacious in finding optimal results where traditional techniques fail. This book is an introduction to the use of GAs to optimizing electromagnetic systems.

This book begins with an introduction to optimization and some of the commonly used numerical optimization routines. Chapter 1 provides the motivation for the need for a more powerful "global" optimization algorithm in contrast to the many "local" optimizers that are prevalent. The next chapter introduces the GA to the reader in both binary and continuous variable forms. MATLAB® commands are given as examples. Chapter 3 provides two step-by-step examples of optimizing antenna arrays. This chapter serves as an excellent introduction to the following chapter, on optimizing antenna arrays. GAs have been applied to the optimization of antenna arrays more than has any other electromagnetics topic. Chapter 5 somewhat follows Chapter 4, because it reports the use of a GA as an adaptive algorithm. Adaptive and smart arrays are the primary focal points, but adaptive reflectors and crossed dipoles are also presented. Chapter 6 explains the optimization of several different wire antennas, starting with the famous "crooked monopole." Chapter 7 is a review of the results for horn, reflector, and microstip patch antennas. Optimization of these antennas entails computing power significantly greater than that required for wire antennas. Chapter 8 diverges from antennas to present results on GA optimization of scattering. Results include scattering from frequency-selective surfaces and electromagnetic bandgap materials. Finally, chapter 9 presents ideas on operator and parameter selection for a GA. In addition, particle swarm optimization and multiple objective optimization are explained in detail. The Appendix contains some MATLAB® code for those who want to try it, followed by a chronological list of publications grouped by topic.

State College, Pennsylvania

RANDY L. HAUPT
DOUGLAS H. WERNER

Acknowledgments

A special thanks goes to Lana Yezhova for assisting in the preparation and editing of the manuscript for this book, and to Rodney A. Martin for the cover art. The authors would also like to express their sincere appreciation to the following individuals, each of whom provided a valuable contribution to the material presented in this book:

Zikri Bayraktar
Daniel W. Boeringer
Jeremy A. Bossard
Matthew G. Bray
Jennifer Carvajal
James W. Culver
Steven D. Eason
K. Prakash Hari
Douglas J. Kern
Ling Li
Derek S. Linden
Kenneth A. O'Connor
Joshua S. Petko
Shawn Rogers
Thomas G. Spence
Pingjuan L. Werner
Michael J. Wilhelm

1

Introduction to Optimization in Electromagnetics

As in other areas of engineering and science, research efforts in electromagnetics have concentrated on finding a solution to an integral or differential equation with boundary conditions. An example is calculating the radar cross section (RCS) of an airplane. First, the problem is formulated for the size, shape, and material properties associated with the airplane. Next, an appropriate mathematical description that exactly or approximately models the airplane and electromagnetic waves is applied. Finally, numerical methods are used for the solution. One problem has one solution. Finding such a solution has proved quite difficult, even with powerful computers.

Designing the aircraft with the lowest RCS over a given frequency range is an example of an optimization problem. Rather than finding a single solution, optimization implies finding many solutions then selecting the best one. Optimization is an inherently slow, difficult procedure, but it is extremely useful when well done. The difficult problem of optimizing an electromagnetics design has only recently received extensive attention.

This book concentrates on the genetic algorithm (GA) approach to optimization that has proved very successful in applications in electromagnetics. We do not think that the GA is the best optimization algorithm for all problems. It has proved quite successful, though, when many other algorithms have failed. In order to appreciate the power of the GA, a background on the most common numerical optimization algorithms is given in this chapter to familiarize the reader with several optimization algorithms that can be applied to

Genetic Algorithms in Electromagnetics, by Randy L. Haupt and Douglas H. Werner

electromagnetics problems. The antenna array has historically been one of the most popular optimization targets in electromagnetics, so we continue that tradition as well.

The first optimum antenna array distribution is the binomial distribution proposed by Stone [1]. As is now well known, the amplitude weights of the elements in the array correspond to the binomial coefficients, and the resulting array factor has no sidelobes. In a later paper, Dolph mapped the Chebyshev polynomial onto the array factor polynomial to get all the sidelobes at an equal level [2]. The resulting array factor polynomial coefficients represent the Dolph–Chebyshev amplitude distribution. This amplitude taper is optimum in that specifying the maximum sidelobe level results in the smallest beamwidth, or specifying the beamwidth results in the lowest possible maximum sidelobe level. Taylor developed a method to optimize the sidelobe levels and beamwidth of a line source [3]. Elliot extended Taylor's work to new horizons, including Taylor-based tapers with asymmetric sidelobe levels, arbitrary sidelobe level designs, and null-free patterns [4]. It should be noted that Elliot's methods result in complex array weights, requiring both an amplitude and phase variation across the array aperture. Since the Taylor taper optimized continuous line sources, Villeneuve extended the technique to discrete arrays [5]. Bayliss used a method similar to Taylor's amplitude taper but applied to a monopulse difference pattern [6]. The first optimized phase taper was developed for the endfire array. Hansen and Woodyard showed that the array directivity is increased through a simple formula for phase shifts [7].

Iterative numerical methods became popular for finding optimum array tapers beginning in the 1970s. Analytical methods for linear array synthesis were well developed. Numerical methods were used to iteratively shape the mainbeam while constraining sidelobe levels for planar arrays [8–10]. The Fletcher–Powell method [11] was applied to optimizing the footprint pattern of a satellite planar array antenna. An iterative method has been proposed to optimize the directivity of an array via phase tapering [12] and a steepest-descent algorithm used to optimize array sidelobe levels [13]. Considerable interest in the design of nonuniformly spaced arrays began in the late 1950s and early 1960s. Numerical optimization attracted attention because analytical synthesis methods could not be found. A spotty sampling of some of the techniques employed include linear programming [14], dynamic programming [15], and steepest descent [16]. Many statistical methods have been used as well [17].

1.1 OPTIMIZING A FUNCTION OF ONE VARIABLE

Most practical optimization problems have many variables. It's usually best to learn to walk before learning to run, so this section starts with optimizing one variable; then the next section covers multiple variable optimization. After describing a couple of single-variable functions to be optimized, several single variable optimization routines are introduced. Many of the multidimensional optimization routines rely on some version of the one-dimensional optimization algorithms described here.

Optimization implies finding either the minimum or maximum of an objective function, the mathematical function that is to be optimized. A variable is passed to the objective function and a value returned. The goal of optimization is to find the combination of variables that causes the objective function to return the highest or lowest possible value.

Consider the example of minimizing the output of a four-element array when the signal is incident at an angle ϕ. The array has equally spaced elements ($d = \lambda/2$) along the x axis (Fig. 1.1). If the end elements have the same variable amplitude (a), then the objective function is written as

$$AF_1(a) = 0.25\left|a + e^{j\Psi} + e^{j2\Psi} + ae^{j3\Psi}\right| \tag{1.1}$$

where $\Psi = kdu$
$k = 2\pi/\lambda$
$\lambda =$ wavelength
$u = \cos\phi$

A graph of AF_1 for all values of u when $a = 1$ is shown in Figure 1.2. If $u = 0.8$ is the point to be minimized, then the plot of the objective function as a function of a is shown in Figure 1.3. There is only one minimum at $a = 0.382$.

Another objective function is a similar four-element array with uniform amplitude but conjugate phases at the end elements

$$AF_2(\delta) = 0.25\left|e^{j\delta} + e^{j\Psi} + e^{j2\Psi} + e^{-j\delta}e^{j3\Psi}\right| \tag{1.2}$$

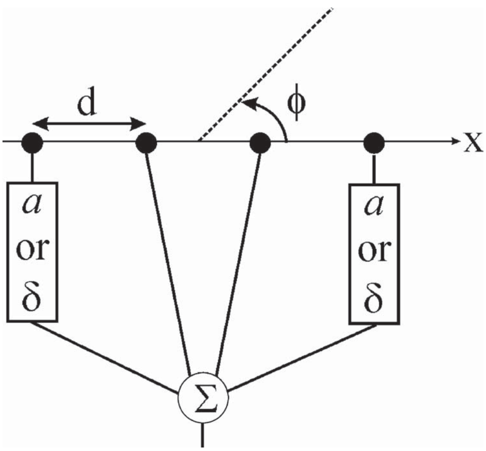

Figure 1.1. *Four-element array with two weights.*

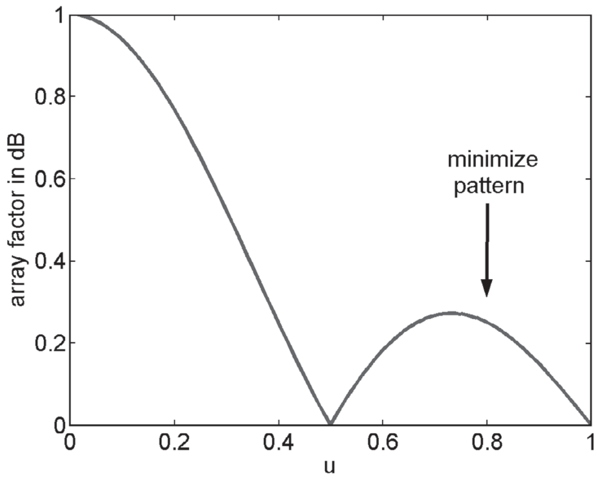

Figure 1.2. *Array factor of a four-element array.*

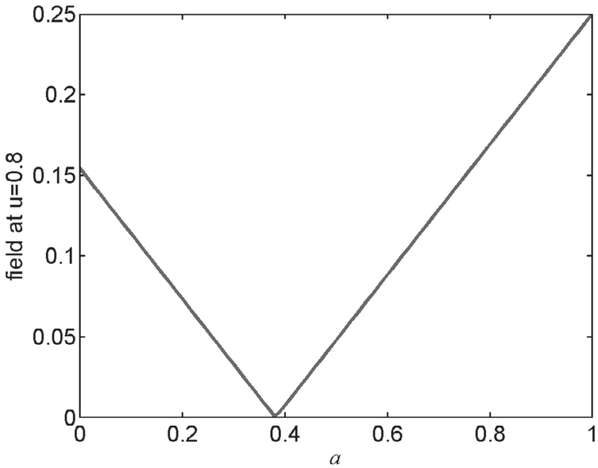

Figure 1.3. *Objective function with input* a *and output the field amplitude at* u = *0.8.*

where the phase range is given by $0 \leq \delta \leq \pi$. If $u = 0.8$ is the point to be minimized, then the plot of the objective function as a function of δ is as shown in Figure 1.4. This function is more complex in that it has two minima. The global or lowest minimum is at $\delta = 1.88$ radians while a local minimum is at $\delta = 0$.

Finding the minimum of (1.1) [Eq. (1.1), above] is straightforward—head downhill from any starting point on the surface. Finding the minimum of (1.2) is a little more tricky. Heading downhill from any point where $\delta < 0.63$ radian (rad) leads to the local minimum or the wrong answer. A different strategy is needed for the successful minimization of (1.2).

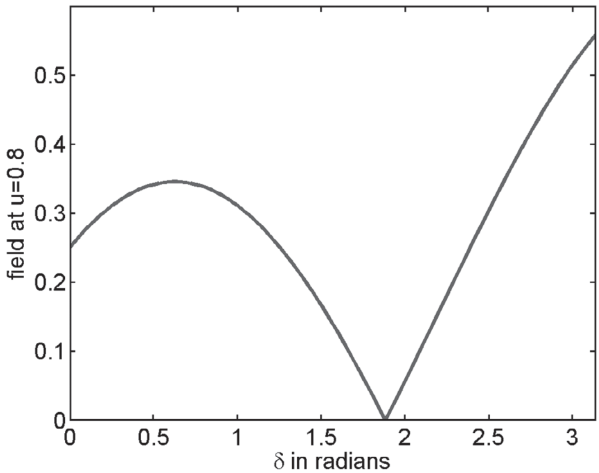

Figure 1.4. *Objective function with input δ and output the field amplitude at* u = *0.8.*

1.1.1 Exhaustive Search

One way to feel comfortable about finding a minimum is to check all possible combinations of input variables. This approach is possible for a small finite number of points. Probably the best example of an exhaustive search is graphing a function and finding the minimum on the graph. When the graph is smooth enough and contains all the important features of the function in sufficient detail, then the exhaustive search is done. Figures 1.3 and 1.4 are good examples of exhaustive search.

1.1.2 Random Search

Checking every possible point for a minimum is time-consuming. Randomly picking points over the interval of interest may find the minimum or at least come reasonably close. Figure 1.5 is a plot of AF_1 with 10 randomly selected points. Two of the points ended up close to the minimum. Figure 1.6 is a plot of AF_2 with 10 randomly selected points. In this case, six of the points have lower values than the local minimum at $\delta = 0$. The random search process can be refined by narrowing the region of guessing around the best few function evaluations found so far and guessing again in the new region. The odds of all 10 points appearing at $\delta < 0.63$ for AF_2 is $(0.63/\pi)^{10} = 1.02 \times 10^{-7}$, so it is unlikely that the random search would get stuck in this local minimum with 10 guesses. A quick random search could also prove worthwhile before starting a downhill search algorithm.

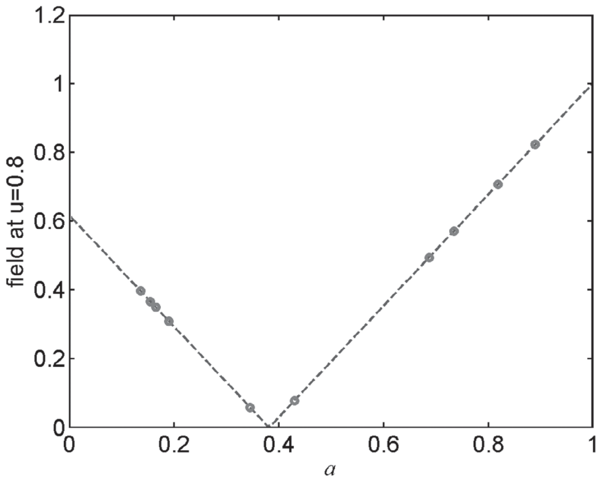

Figure 1.5. *Ten random guesses (circles) superimposed on a plot of AF_1.*

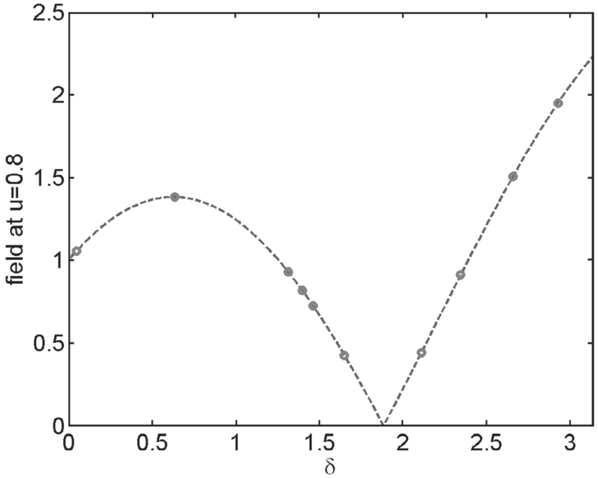

Figure 1.6. *Ten random guesses (circles) superimposed on a plot of AF_2.*

1.1.3 Golden Search

Assume that a minimum lies between two points a and b. Three points are needed to detect a minimum in the interval: two to bound the interval and one in between that is lower than the bounds. The goal is to shrink the interval by picking a point (c) in between the two endpoints (a and b) at a distance Δ_1 from a (see Fig. 1.7). Now, the interval is divided into one large interval and one small interval. Next, another point (d) is selected in the larger of the two subintervals. This new point is placed at a distance Δ_2 from c. If the new point

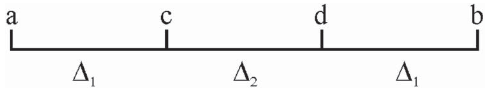

Figure 1.7. *Golden search interval.*

on the reduced interval ($\Delta_1 + \Delta_2$) is always placed at the same proportional distance from the left endpoint, then

$$\frac{\Delta_1}{\Delta_1 + \Delta_2 + \Delta_1} = \frac{\Delta_2}{\Delta_1 + \Delta_2} \tag{1.3}$$

If the interval is normalized, the length of the interval is

$$\Delta_1 + \Delta_2 + \Delta_1 = 1 \tag{1.4}$$

Combining (1.3) and (1.4) yields the equation

$$\Delta_1^2 - 3\Delta_1 + 1 = 0 \tag{1.5}$$

which has the root

$$\Delta_1 = \frac{\sqrt{5} - 1}{2} = 0.38197\ldots \tag{1.6}$$

This value is known as the "golden mean" [18].

The procedure above described is easy to put into an algorithm to find the minimum of AF_2. As stated, the algorithm begins with four points (labeled 1–4 in Fig. 1.8). Each iteration adds another point. After six iterations, point 8 is reached, which is getting very close to the minimum. In this case the golden search did not get stuck in the local minimum. If the algorithm started with points 1 and 4 as the bound, then the algorithm would have converged on the local minimum rather than the global minimum.

1.1.4 Newton's Method

Newton's method is a downhill sliding technique that is derived from the Taylor's series expansion for the derivative of a function of one variable. The derivative of a function evaluated at a point x_{n+1} can be written in terms of the function derivatives at a point x_n

$$f'(x_{n+1}) = f'(x_n) + f''(x_n)(x_{n+1} - x_n) + \frac{f'''(x_n)}{2!}(x_{n+1} - x_n)^2 + \cdots \tag{1.7}$$

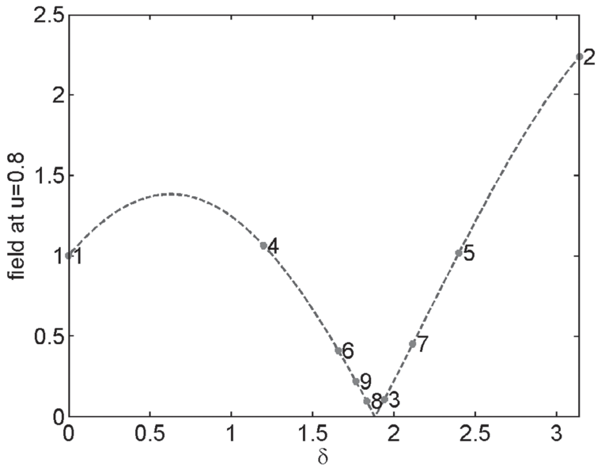

Figure 1.8. *The first eight function evaluations (circles) of the golden search algorithm when minimizing AF_2.*

Keeping only the first and second derivatives and assuming that the next step reaches the minimum or maximum, then (1.7) equals zero, so

$$f'(x_n) + f''(x_n)(x_{n+1} - x_n) = 0 \tag{1.8}$$

Solving for x_{n+1} yields

$$x_{n+1} = x_n - \frac{f'(x_n)}{f''(x_n)} \tag{1.9}$$

If no analytical derivatives are available, then the derivatives in (1.9) are approximated by a finite-difference formula

$$x_{n+1} = x_n - \frac{\Delta[f(x_{n+1}) - f(x_{n-1})]}{2[f(x_{n+1}) - 2f(x_n) + f(x_{n-1})]} \tag{1.10}$$

where

$$\Delta = |x_{n+1} - x_n| = |x_n - x_{n-1}| \tag{1.11}$$

This approximation slows the method down but is often the only practical implementation.

Let's try finding the minimum of the two test functions. Since it's not easy to take the derivatives of AF_1 and AF_2, finite-difference approximations will be used instead. Newton's method converges on the minimum of AF_1 for every starting point in the interval. The second function is more interesting,

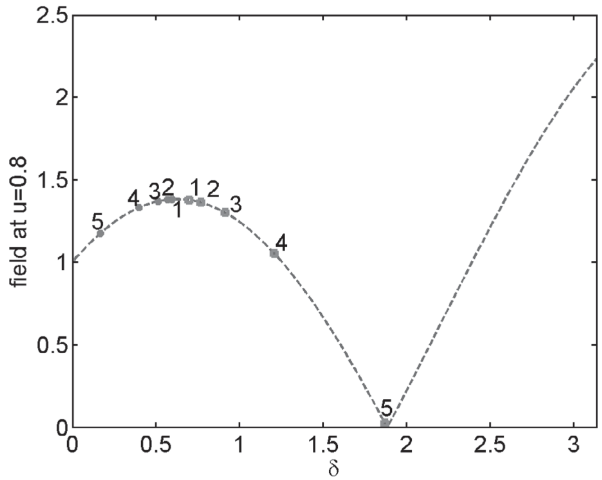

Figure 1.9. *The convergence of Newton's algorithm when starting at two different points.*

though. Figure 1.9 shows the first five points calculated by the algorithm from two different starting points. A starting point at $\delta = 0.6$ radians results in the series of points that heads toward the local minimum on the left. When the starting point is $\delta = 0.7$ rad, then the algorithm converges toward the global minimum. Thus, Newton's method is known as a *local search algorithm*, because it heads toward the bottom of the closest minimum. It is also a nonlinear algorithm, because the outcome can be very sensitive to the initial starting point.

1.1.5 Quadratic Interpolation

The techniques derived from Taylor's series assumed that the function is quadratic near the minimum. If this assumption is valid, then we should be able to approximate the function by a quadratic polynomial near the minimum and find the minimum of that quadratic polynomial interpolation [19]. Given three points on an interval (x_0, x_1, x_2), the extremum of the quadratic interpolating polynomial appears at

$$x_3 = \frac{f(x_0)\left(x_1^2 - x_2^2\right) + f(x_1)(x_2^2 - x_0^2) + f(x_2)(x_0^2 - x_1^2)}{2f(x_0)(x_1 - x_2) + 2f(x_1)(x_2 - x_0) + 2f(x_2)(x_0 - x_1)} \tag{1.12}$$

When the three points are along the same line, the denominator is zero and the interpolation fails. Also, this formula can't differentiate between a minimum and a maximum, so some caution is necessary to insure that it pursues a minimum.

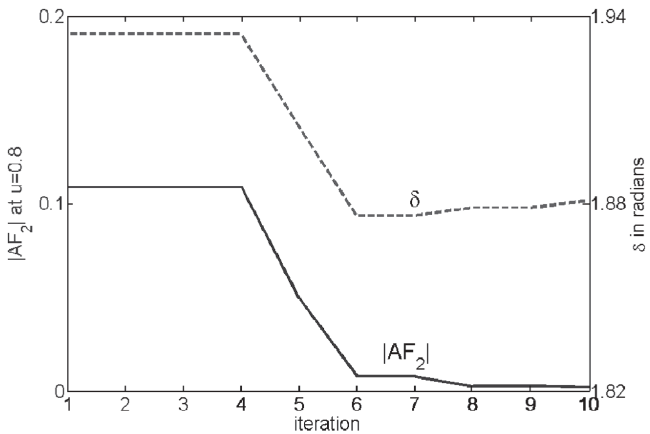

Figure 1.10. *Convergence of the MATLAB quadratic interpolation routine when minimizing* AF_2.

MATLAB uses a combination of golden search and quadratic interpolation in its function *fminbnd.m*. Figure 1.10 shows the convergence curves for the field value on the left-hand vertical axis and the phase in radians on the right-hand vertical axis. This approach converged in just 10 iterations.

1.2 OPTIMIZING A FUNCTION OF MULTIPLE VARIABLES

Usually, arrays have many elements; hence many variables need to be adjusted in order to optimize some aspect of the antenna pattern. To demonstrate the complexity of dealing with multiple dimensions, the objective functions in (1.1) and (1.2) are extended to two variables and three angle evaluations of the array factor.

$$\mathrm{AF}_3(a_1, a_2) = \frac{1}{6}\sum_{m=1}^{3}\left|a_2 + a_1 e^{j\Psi_m} + e^{j2\Psi_m} + e^{j3\Psi_m} + a_1 e^{j4\Psi_m} + a_2 e^{j5\Psi_m}\right| \tag{1.13}$$

$$\mathrm{AF}_4(\delta_1, \delta_2) = \frac{1}{6}\sum_{m=1}^{3}\left|e^{j\delta_2} + e^{j\delta_1} e^{j\Psi_m} + e^{j2\Psi_m} + e^{j3\Psi_m} + e^{j\delta_1} e^{j4\Psi_m} + e^{j\delta_2} e^{j5\Psi_m}\right| \tag{1.14}$$

Figure 1.11 is a diagram of the six-element array with two independent adjustable weights. The objective function returns the sum of the magnitude of the array factor at three angles: ϕ_m = 120°, 69.5°, and 31.8°. The array factor for a uniform six-element array is shown in Figure 1.12. Plots of the objective function for all possible combinations of the amplitude and phase weights appear in Figures 1.13 and 1.14. The amplitude weight objective func-

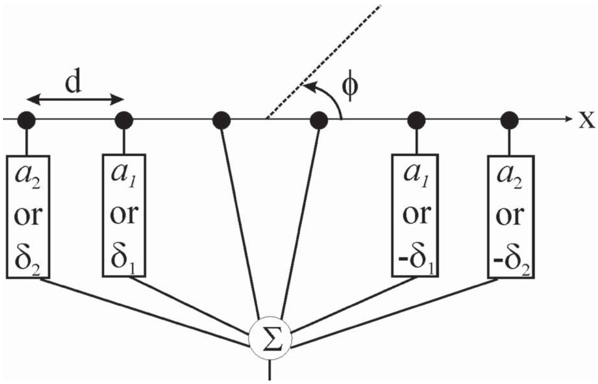

Figure 1.11. *A six-element array with two independent, adjustable weights.*

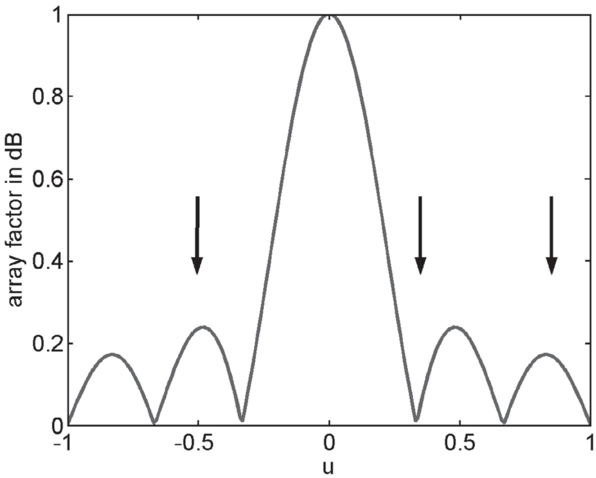

Figure 1.12. *The array factor for a six-element uniform array.*

tion has a single minimum, while the phase weight objective function has several minima.

1.2.1 Random Search

Humans are intrigued by guessing. Most people love to gamble, at least occasionally. Beating the odds is fun. Guessing at the location of the minimum sometimes works. It's at least a very easy-to-understand method for minimization—no Hessians, gradients, simplexes, and so on. It takes only a couple of lines of MATLAB code to get a working program. It's not very elegant, though, and many people have ignored the power of random search in the

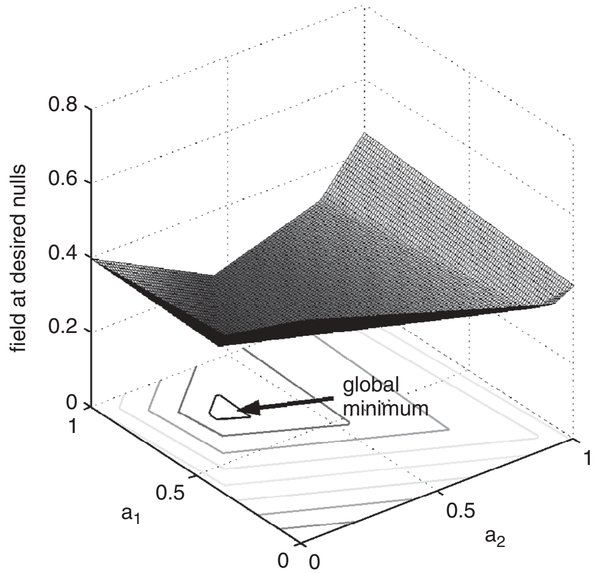

Figure 1.13. *Plot of the objective function for AF_3.*

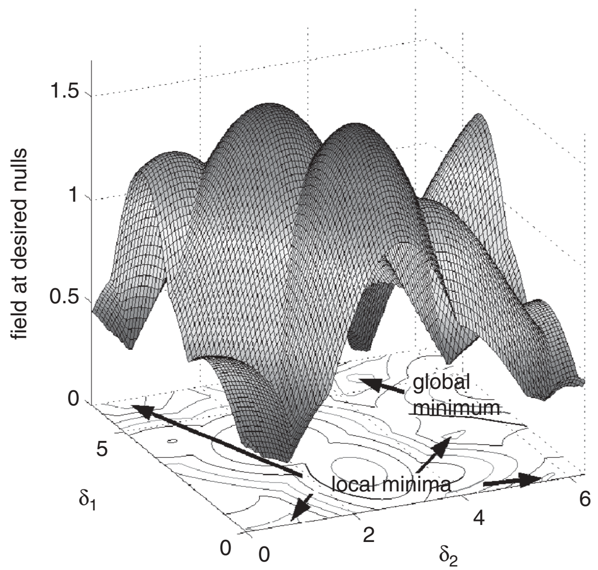

Figure 1.14. *Plot of the objective function for AF_4.*

development of sophisticated minimization algorithms. We often model processes in nature as random events, because we don't understand all the complexities involved. A complex cost function more closely approximates nature's ways, so the more complex the cost function, the more likely that random

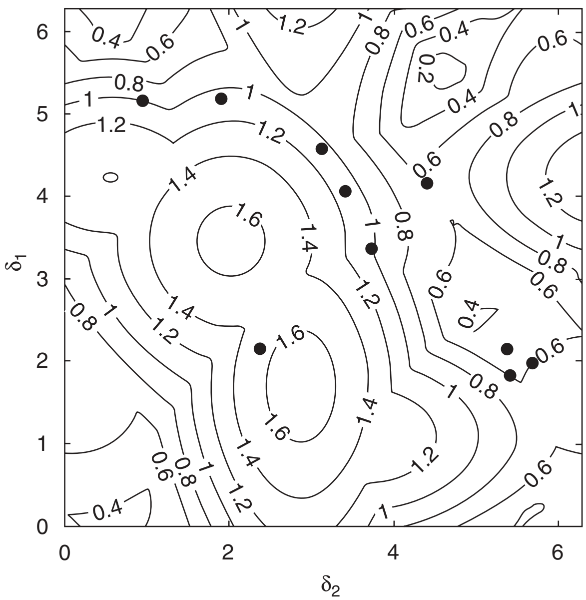

Figure 1.15. *Ten random guesses for AF$_4$.*

guessing plays an important part in finding the minimum. Even a local optimizer makes use of random starting points. Local optimizers are made more “global” by making repeated starts from several different, usually random, points on the cost surface.

Figure 1.15 shows 10 random guesses on a contour plot of AF$_4$. This first attempt clearly misses some of the regions with minima. The plot in Figure 1.16 results from adding 20 more guesses. Even after 30 guesses, the lowest value found is not in the basin of the global minimum. Granted, a new set of random guesses could easily land a value near the global minimum. The problem, though, is that the odds decrease as the number of variables increases.

1.2.2 Line Search

A line search begins with an arbitrary starting point on the cost surface. A vector or line is chosen that cuts across the cost surface. Steps are taken along this line until a minimum is reached. Next, another vector is found and the process repeated. A flowchart of the algorithm appears in Figure 1.17. Selecting the vector and the step size has been an area of avid research in numerical optimization. Line search methods work well for finding a minimum of a quadratic function. They tend to fail miserably when searching a cost surface with many minima, because the vectors can totally miss the area where the global minimum exists.

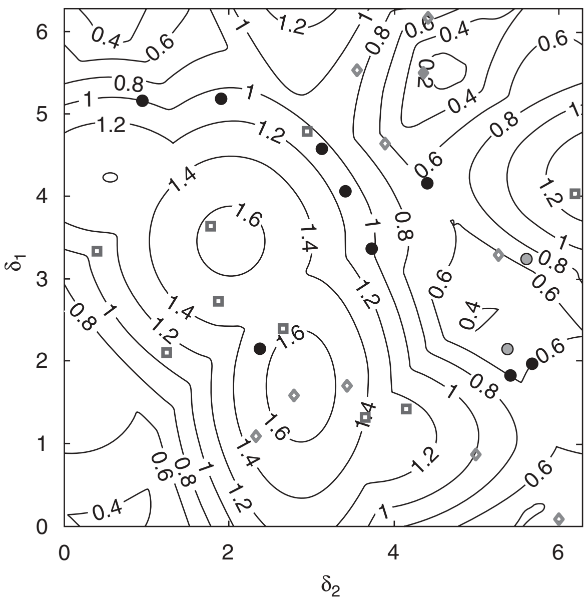

Figure 1.16. Thirty random guesses for AF_4.

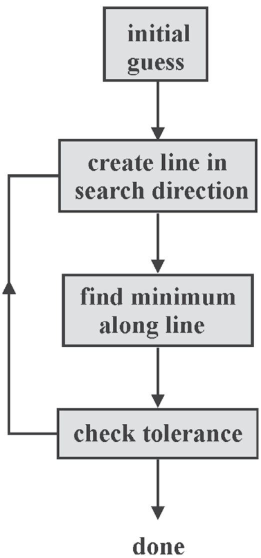

Figure 1.17. Flowchart of a line search minimization algorithm.

The easiest line search imaginable is the coordinate search method. If the function has two variables, then the algorithm begins at a random point, holds one variable constant, and searches along the other variable. Once it reaches a minimum, it holds the second variable constant and searches along the first