AFRL-SN-HS-TR-2005-012

MILITARY ANTENNA DESIGN USING SIMPLE AND COMPETENT GENETIC ALGORITHMS

Edward E. Altshuler
David E. Goldberg
Robert Mailloux
Terry H. O'Donnell
Scott G. Santarelli
Hugh L. Southall
Tian-Li Yu

In-House Technical Report: June 2003 – June 2004

AIR FORCE RESEARCH LABORATORY
Sensors Directorate
Electromagnetics Technology Division
80 Scott Drive
Hanscom AFB MA 01731-2909

TECHNICAL REPORT

Military Antenna Design Using Simple and Competent Genetic Algorithms

Unlimited, Statement A

NOTICE

USING GOVERNMENT DRAWINGS, SPECIFICATIONS, OR OTHER DATA INCLUDED IN THIS DOCUMENT FOR ANY PURPOSE OTHER THAN GOVERNMENT PROCUREMENT DOES NOT IN ANY WAY OBLIGATE THE US GOVERNMENT. THE FACT THAT THE GOVERNMENT FORMULATED OR SUPPLIED THE DRAWINGS, SPECIFICATIONS, OR OTHER DATA DOES NOT LICENSE THE HOLDER OR ANY OTHER PERSON OR CORPORATION; OR CONVEY ANY RIGHTS OR PERMISSION TO MANUFACTURE, USE, OR SELL ANY PATENTED INVENTION THAT MAY RELATE TO THEM.

THIS TECHNICAL REPORT HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION.

//signature//
Scott G. Santarelli
Project Manager, Electromagnetics Scattering Branch

//signature//
Livio D. Poles
Chief, Electromagnetics Scattering Branch

//signature//
Michael N. Alexander
Technical Advisor
Electromagnetics Technology Division

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*		3. DATES COVERED *(From - To)*
01-02-2005	IN-HOUSE TECHNICAL REPORT	1 June 2003 – 1 June 2004

4. TITLE AND SUBTITLE
Military Antenna Design Using Simple and Competent Genetic Algorithms

5a. CONTRACT NUMBER
N/A

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER
2304

6. AUTHOR(S)

Scott G. Santarelli, Edward E. Altshuler, David E. Goldberg, Robert Mailloux, Terry H. O'Donnell, Hugh L. Southall, and Tian-Li Yu,

5d. PROJECT NUMBER
4916

5e. TASK NUMBER
HA

5f. WORK UNIT NUMBER
01

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

AFRL/SNHA
80 Scott Drive
Hanscom AFB, MA
01731

Illinois Genetic Algorithm Laboratory
104 South Mathews
Urbana, IL
61801

Arinc, Inc.
70 Westview St.
Lexington, MA
02173

8. PERFORMING ORGANIZATION REPORT

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Electromagnetics Technology Division
Sensors Directorate
Air Force Research Laboratory
80 Scott Drive
Hanscom AFB MA 01731-2909

10. SPONSOR/MONITOR'S ACRONYM(S)
AFRL/SNHA

11. SPONSOR/MONITOR'S REPORT NUMBER(S)
AFRL-SN-HS-TR-2005-012

12. DISTRIBUTION / AVAILABILITY STATEMENT
APPROVED FOR PUBLIC RELEASE, DISTRIBUTION UNLIMITED. (ESC-04-1144, NOVEMBER 15, 2004).

13. SUPPLEMENTARY NOTES

14. ABSTRACT

Over the past decade, the Air Force Research Laboratory (AFRL) Antenna Technology Branch at Hanscom AFB has employed the simple genetic algorithm (SGA) as an optimization tool for a wide variety of antenna applications. Over roughly the same period, researchers at the Illinois Genetic Algorithm Laboratory (IlliGAL) at the University of Illinois at Urbana Champaign have developed GA design theory and advanced GA techniques called *competent genetic algorithms*—GAs that solve hard problems quickly, reliably, and accurately. Recently, under the guidance and direction of the Air Force Office of Scientific Research (AFOSR), the two laboratories have formed a collaboration, the common goal of which is to apply simple, competent, and hybrid GA techniques to challenging antenna problems.

This paper is composed of two parts. The first part of this paper summarizes previous research conducted by AFRL at Hanscom for which SGAs were implemented to obtain acceptable solutions to several antenna problems. This research covers diverse areas of interest, including array pattern synthesis; antenna test-bed design; gain enhancement; electrically small, single, bent wire elements; and wideband antenna elements.

The second part of this paper starts by briefly reviewing the design theory and design principles necessary for the invention and implementation of fast, scalable genetic algorithms. A particular procedure, the hierarchical Bayesian optimization algorithm (hBOA) is then briefly outlined, and the remainder of the paper describes collaborative efforts of AFRL and IlliGAL to solve more difficult antenna problems. In particular, recent results of using hBOA to optimize a novel, wideband overlapped subarray system to achieve -30-dB sidelobes over a 20% bandwidth. The problem was sufficiently difficult that acceptable solutions were not obtained using SGAs. The case study demonstrates the utility of using more advanced GA techniques to obtain acceptable solution quality as problem difficulty increases.

15. SUBJECT TERMS
antennas, phased arrays, optimization algorithm, genetic algorithm, overlapped subarrays

16. SECURITY CLASSIFICATION OF:			17.LIMITATION OF ABSTRACT	18.NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON Dr. Scott G. Santarelli
a. REPORT	b. ABSTRACT	c.THIS PAGE			19b. TELEPHONE NUMBER *(include area code)*
Unclassified	Unclassified	Unclassified	UU	91	*781-377-6854*

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. Z39.18

TABLE OF CONTENTS

Table of Contents iii

List of Figures v

List of Tables viii

1.0 Introduction 1

2.0 Application of Simple Genetic Algorithms (SGAs) to Antenna Design and Array Optimization 2

2.1 Introduction 2

2.2 Simple Genetic Algorithm (SGA) Tutorial 3

2.3 Survey of Projects and Experiments 5

2.3.1 A Circularly Polarized Antenna with Hemispherical Coverage 5

2.3.2 An Electrically Small Antenna 6

2.3.3 An Ultra Wideband Antenna 6

2.3.4 Genetic Yagi Antennas 7

2.3.5 Simple Genetic Algorithm Array Pattern Synthesis 8

2.3.6 Antenna Pattern Control of Phased Array Antennas 11

2.3.7 Advances in Electrically Small Antennas 14

2.3.7.1 Coordinate Chromosome Representation 14

2.3.7.2 Angular Chromosome Representations 14

2.3.7.3 Comparison of Chromosome Limitations 15

2.3.7.4 Results for Electrically Small, Bent-Wire Chromosomes 16

2.3.8 Hybrid DISS Transmit Antenna 16

2.3.8.1 Genetic DISS Hybrid Antenna 17

2.3.8.2 Genetic Algorithm and Optimization Goals 18

2.3.8.3 Results 18

2.3.9 Technology Transfer to Other Applications 19

2.3.9.1 Reflector Dish Array Pattern Synthesis for RCS Measurements 19

2.4 Conclusions 19

3.0 Optimization of a Constrained Feed Network for a Linear Array: SGA vs. Competent GA 20

3.1 Introduction 20

3.2 Introduction to hBOA 20

3.2.1 GA Design Theory and Competent GAs 21

3.2.2 The Bayesian Optimization Algorithm (BOA) 22

3.2.3 Learning Bayesian Networks 23

3.2.4 Hierarchical Decomposition – From BOA to hBOA 23

3.3 Optimization of a Constrained Feed Network for a Linear Array 24

3.3.1 Problem Statement 24

3.3.2 Approach 26

3.3.2.1 Implementation of the Simple Genetic Algorithm (SGA) 26

3.3.2.2 Implementation of the Hierarchical Bayesian Optimization Algorithm (hBOA) 27

3.3.2.3 Objective Function 28

3.3.3 Results: SGA vs. hBOA 30

3.4 Summary/Conclusions 32

4.0 Summary/Conclusions 32

References 79

LIST OF FIGURES

Figure 1: SGA parameters and operators 34

Figure 2: Sketch of GPS/Iridium Antenna 35

Figure 3: (Fig. 2) VSWR of GPS/Iridium Antenna 36

Figure 4: Radiation Pattern of GPS/Iridium Antenna 37

Figure 5: Electrically Small Genetic Antenna 38

Figure 6: Q of Electrically Small Genetic Antenna 39

Figure 7: Ultra-Wideband, Impedance-Loaded Genetic Antenna 40

Figure 8: VSWR of Ultra Wideband Antenna 41

Figure 9: Conventional Yagi Antenna 42

Figure 10: Genetic Yagi 43

Figure 11: Gain of Yagi Antennas 44

Figure 12: Experimental Configuration 45

Figure 13: Single Element Transfer-Function Characteristics 46

Figure 14: Experimental Radiation Pattern 47

Figure 15: Chromosomal Representation 48

Figure 16: Example Gray-Coding Scheme 49

Figure 17: The adaptive array model with the SGA [12] 50

Figure 18: Flow chart of the adaptive SGA [12] 51

Figure 19: A simple, bit-level example of the genetic algorithm for a two-element array [12] 52

Figure 20: The antenna pattern resulting from weights (phase-shifter settings) produced by the SGA with an interfering source at 45° [12] 53

Figure 21: Measured (solid) and theoretical (dashed) array beam patterns for the 20° null 54

55

Figure 22: Measured (solid) and theoretical (dashed) array beam patterns for the 15° scanned beam

Figure 23: Typical subarray pattern illustrating the shaded areas used to calculate the cost function for the genetic algorithm optimization 56

Figure 24: Skobelev (N = 2 case) subarray pattern simulated using quadrature hybrid coupler coupling coefficients obtained from the SGA 57

Figure 25: The real-valued, Cartesian coordinate-based genetic antenna chromosome and a typical resulting antenna 58

Figure 26: The absolute angle genetic antenna chromosome and a typical resulting antenna 59

Figure 27: The relative angle genetic antenna chromosome and a typical resulting antenna 60

Figure 28: Comparison of VSWR performance for the Cartesian coordinate, absolute-angle, and relative-angle GAs and a normal mode helix 61

Figure 29: The 0.05λ relative-angle, electrically small, bent-wire antenna, whose measured VSWR is depicted in Figure 28 62

Figure 30: The original DISS TX antenna – an off-the-shelf TCI 613 communications antenna 63

Figure 31: A schematic of the genetic DISS hybrid antenna design and the chromosome used to encode it into the SGA 64

Figure 32: The genetic DISS hybrid TX antenna 65

Figure 33: A schematic of the 3-dish reflector array used to illuminate the target on the pylon for RCS (radar cross section) measurements on the RATSCAT ground-bounce range 66

Figure 34: The power taper on the pylon resulting from the SGA configuration of the reflector dish array 67

Figure 35: An example of a Bayesian network structure 68

Figure 36: Single section of antenna system, including front-end array, Rotman lens, and Butler matrix 69

Figure 37: Far-field radiation patterns as a function of frequency for the optimized ideal system 70

Figure 38: Inverse transfer function for a single Rotman lens (phase only) 71

72

Figure 39: Far-field radiation patterns as a function of frequency when the experimental Rotman lens data is incorporated into the system model and the ideal system weights from [40] are applied

Figure 40: Chromosomal encoding scheme for SGA 73

Figure 41: Objective Function for Case 1 74

Figure 42: Objective Function for Cases 2 and 3 75

Figure 43: SGA vs. hBOA performance, CASE 1, Run 1 76

Figure 44: SGA vs. hBOA performance, CASE 2, Run 1 77

Figure 45: SGA vs. hBOA performance, CASE 3, Run 1 78

LIST OF TABLES

Table 1: The results of the SGA and hBOA for CASE 1 31

Table 2: The results of the SGA and hBOA for CASE 2 31

Table 3: The results of the SGA and hBOA for CASE 3 32

1.0 Introduction

Over the past decade, the Air Force Research Laboratory (AFRL) Antenna Technology Branch at Hanscom AFB has employed the simple genetic algorithm (SGA) [1] as an optimization tool for a wide variety of antenna applications. Over roughly the same period, researchers at the Illinois Genetic Algorithm Laboratory (IlliGAL) at the University of Illinois at Urbana Champaign have developed GA design theory and advanced GA techniques called *competent genetic algorithms*—GAs that solve hard problems quickly, reliably, and accurately [2]. Recently, under the guidance and direction of the Air Force Office of Scientific Research (AFOSR), the two laboratories have formed a collaboration, the common goal of which is to apply simple, competent, and hybrid GA techniques to challenging antenna problems.

This paper is organized as follows. Section 2 is essentially a survey of the SGA-related research projects and experiments conducted at AFRL Hanscom over the past decade. This research covers diverse areas of interest, including array pattern synthesis; antenna test-bed design; gain enhancement; electrically small, single, bent-wire elements; and wideband elements. A review of fundamental SGA concepts, including a description of common parameters and operators, is provided as a brief tutorial. At the end of the section, we summarize our accomplishments and highlight some of the more interesting insights we've gained (as a direct result of our research) concerning SGA implementation and application.

Section 3 describes the optimization of a constrained feed network for a linear array. Both a simple and competent GA were applied to this problem in an effort to find a solution that meets the system requirements/specifications. The section starts by briefly reviewing the design theory and design principles necessary for the invention and implementation of fast, scalable genetic algorithms. A particular procedure, the hierarchical Bayesian optimization algorithm (hBOA) [3-6] is then briefly outlined. This section also describes three variations of the objective function that were implemented in this experiment. Following this, we present the results, demonstrating that the problem was sufficiently difficult that acceptable solutions were not obtainable using the SGA (regardless of the objective function used). Computationally, hBOA finds solutions that meet those of the SGA in all three cases. Electromagnetically, the first two cases did not exhibit solutions with sufficient sidelobe rejection. In case 3, min-max solutions using hBOA were able to give sidelobe rejection of -27dB, whereas the SGA got stuck in far inferior local optima. We analyze the performance of the SGA and hBOA for all three cases in order to gain insight into the inherent strengths and weaknesses of these approaches. This case study demonstrates the utility of using more advanced GA techniques to obtain acceptable solution quality as problem difficulty increases.

Finally, in Section 4, we summarize the paper and discuss future collaborative efforts between the two laboratories.

2.0 Application of Simple Genetic Algorithms (SGAs) to Antenna Design and Array Optimization

2.1 Introduction

Although the existing literature has many examples of simple genetic algorithms (SGAs) applied to antenna design [7], this section focuses solely on research conducted at the Air Force Research Laboratory (AFRL) Antenna Technology Branch. The Antenna Technology Branch is part of the AFRL Sensors Directorate Electromagnetics Technology Division, which is located at Hanscom AFB, MA. Over the past decade, the AFRL Antenna Technology Branch has employed several variants of the SGA as optimization tools for a wide variety of antenna applications. This work covers diverse areas of interest, including array pattern synthesis; antenna test-bed design; gain enhancement; electrically small, bent-wire, single element design; and wideband element design.

Before we begin, it is worth clarifying two terms that we use throughout Section 2 in discussing our application of SGAs to antenna design and optimization. In antenna ***optimization***, we start with a basic antenna-problem formulation resulting from conventional antenna theory and wisdom. Here, the SGA is applied to an existing solution and used to fine-tune the outcome. For example, we may initialize a structure as a type of Yagi antenna and use the SGA to find the best element spacing and lengths for a desired gain and overall boom length.

In contrast, genetic antenna ***design*** implies that we start from scratch with only some basic materials and problem constraints; the shape of the solution itself is not pre-determined. As the SGA explores the solution space, the actual structure and shape of the antenna emerges, sometimes resulting in a well-know antenna solution, such as a top-loaded structure. Other times, novel structures are created, which may yield insight into the basic physics of the problem space itself.

The remainder of this section is outlined as follows. Section 2.2 is provided as a brief tutorial for the reader. Here, we review fundamental SGA concepts, including a description of common parameters and operators. Section 2.3 is essentially a survey of the SGA-related research projects and experiments conducted at the AFRL Antenna Technology Branch over the past decade. Section 2.4 summarizes our accomplishments and highlights some of the more interesting insights we've gained (as a direct result of our research) concerning SGA implementation and application.

2.2 Simple Genetic Algorithm (SGA) Tutorial

A genetic algorithm (GA) is an optimization technique modeled after the genetic process used by living organisms. In nature, the instructions for the design of each living organism are contained within the chromosomes of that organism. These instructions are coded into a sequence of genes on long strands of DNA. A number of these strands make up the entire instruction set from which an organism is designed and maintained throughout its lifetime. Chromosomes and their resulting organisms go through a number of important processes: birth, survival-of-the-fittest, mating and the production of children, and death. Thus, a species of organism adapts itself to its environment.

In a similar manner, designs in just about any engineering discipline can be reduced to a number of instructions or specifications: lengths, diameters, materials, or other characteristics. A GA encodes these instructions/specifications along an artificial chromosome. Basically, a procedure designed to mimic the life cycle of an organic organism is used to optimize the chromosome to a specific objective or set of objectives (*i.e.*, desired cost of final system $\leq$ \$5,000, desired engine efficiency $\geq$ 96%, *etc.*). For the purpose of this paper, genetic algorithms can be divided into two broad categories: *simple genetic algorithms* (SGAs) and *competent genetic algorithms*. The term "competent GAs" has been used to describe GAs that solve a large class of optimization problems in scalable manner and will be addressed in Section 3. A brief tutorial on SGAs is presented below (for a much more detailed description, one should consult [1].

First, a *chromosome* is designed to encode potential solutions to the problem, often as a string of binary or real values. In Figure 1, we show an example of a chromosome used to encode the Cartesian coordinates that represent the nodes of a bent-wire antenna. The GA starts with a large number of randomly generated chromosomes. These chromosomes are essentially a pool of potential design configurations, where each chromosome represents a different design. The range of possible designs is determined by the constraints of the problem, as well as the manner in which the design parameters are encoded along the length of the chromosome.

The *cost* or *objective function* compares each potential solution to a set of objectives, assigning each chromosome a *fitness value*. This provides a means of ranking chromosomes from best to worst. In antenna design, for example, the individual chromosomes are first decoded to extract the values of the design parameters. These parameter values are then fed into an electromagnetic simulator to determine the antenna performance. The fitness value is computed by directly comparing the simulated performance to the desired performance. It should be noted that some properties might be weighted more heavily than others in the calculation of the fitness value (*i.e.*, the designer may feel that a low voltage-standing-wave

ratio (VSWR) is more important than minimizing the amount of material used to build the antenna).

The SGA begins the evolutionary process with an initial (usually random) *population* of chromosomes, as shown in Figure 1. This population represents a small subset of the set of all possible solutions to the problem. Next, the objective function ranks the chromosomes from best to worst. Then, chromosomes are selected from the population in a pseudo-random manner, such that high-ranking chromosomes have a higher probability of being chosen. These chromosomes, the *parents* of the next generation, undergo a *mating* or *recombination* process to produce *children*. Offspring can be created through many different procedures, each of which is essentially a method of combining information from two or more parent chromosomes to form a child that has the potential to outperform its parents.

The children are passed to a *mutation* operator, which is essentially a mechanism by which information encoded along the chromosome is randomly altered. For binary-encoded chromosomes, this is usually a bit-flip, as shown in Figure 1. For real- valued encoded chromosomes, this may be accomplished by adding Gaussian noise or other types of offsets to the current value of the parameter in question. The purpose of mutation is to recover lost genetic information that may not be present in the initial population and is not obtainable by recombination alone. Following mutation, the next population is evaluated and ranked, and the process continues as shown in Figure 1. With succeeding generations, the quality of the strings continues to improve until an acceptable solution is found or convergence occurs (*i.e.*, solutions do not continue to improve).

In essence, selection, recombination, and mutation work together to combine small pieces of salient information (called *building blocks*) from different chromosomes to form "good" solutions. When SGAs work well, chromosomes containing good building blocks will, on average, outperform chromosomes containing inferior building blocks. Thus, each successive generation is populated with chromosomes containing more and more good building blocks (*i.e.*, better quality solutions). The goal of the SGA is to ultimately find a chromosome that is comprised of the best building blocks (*i.e.*, an optimal solution to the problem). When a SGA converges to a non-optimal solution, it is either because crossover does not exchange the correct material or mutation does not explore the proper material. It has been demonstrated that premature convergence due to mining failure is primarily due to a poor match between encoding and genetic operators [1,2]. It will be shown in the second half of this paper that competent GAs are able to learn a good match between operators and encoding during the process of evolution, which effectively eliminates many of the difficulties and pitfalls of SGAs. That being said, it should be noted that for many problems, a SGA proves to be an effective search mechanism that is capable of obtaining solutions that meet or exceed the objectives set by the user.

2.3 Survey of Projects and Experiments

This section is essentially a survey of the SGA-related research projects and experiments conducted at the AFRL Antenna Technology Branch over the past decade. This work covers diverse areas of interest, including array pattern synthesis; antenna test-bed design; gain enhancement; electrically small, bent-wire, single element design; and wideband element design. The common thread that ties these projects together is the implementation of a SGA to either design or optimize an antenna or array of antenna elements; however, these projects represent a wide range of problem difficulty. Thus, we will see that some of these experiments were able to achieve acceptable solutions from the initial SGA parameter choices, whereas other projects required an in depth analysis of the SGA mechanics in order to obtain high quality solutions. Projects that fall into the latter category reveal interesting insights into SGA implementation/application.

2.3.1 A Circularly Polarized Antenna with Hemispherical Coverage

In this section, we describe a vehicular wire antenna that may be used for both the GPS and Iridium systems [8]. At the time of this experiment, the top priorities in GPS antenna design were dual-frequency operation at 1225 MHz (*i.e.*, L2 precision code) and 1575 MHz (*i.e.*, L1 coarse acquisition), the gain of the radiation pattern near the horizon, and the use of circular polarization. The Iridium antenna had similar requirements for the frequency band from 1610 to 1625 MHz. Thus, an antenna was designed (using a SGA) to operate over the band from 1225 to 1625 MHz. For comparison, four previously designed GPS receiving antennas were examined: a circularly polarized patch antenna; an array of square, helical antennas; a quadrifilar helical antenna; and a conical, spiral antenna [8]. The conical, spiral antenna provided the most promising results; however, this antenna had to be placed on lossy absorbing material and was approximately 2 wavelengths in height.

The antenna configuration was simulated using the Numerical Electromagnetics Code (NEC) [9]. A SGA worked in conjunction with NEC to optimize the antenna parameters. The optimized design was then fabricated and tested. The antenna, which consists of 5 copper-tubing segments connected in series, is shown in Figure 2. It fits in a volume approximately 10 cm × 10 cm × 15 cm. The input VSWR and the circular-polarization radiation patterns were computed and measured and are shown in Figures 3 and 4. The VSWR was under 2.2 at the design frequencies of 1225, 1575 and 1625 MHz. The gain varied by less than 12 dB for a 170° sector (it generally fell off near the horizon, so the variation was less for 150° and 160° sectors). Thus, the crooked-wire antenna developed by the SGA has characteristics that are comparable to those of the conical spiral at both GPS L1 and L2 bands and also operates over the Iridium band of 1610-1626.5 MHz. In addition, it is

much smaller than the conical spiral and does not need absorbing material. It is a very inexpensive antenna, since it is fed directly from a coaxial cable and does not require multiple inputs or a phasing network to achieve circular polarization.

2.3.2 An Electrically Small Antenna

One of the major limitations of electrically small antennas is the following: as the size of the antenna decreases, its radiation resistance approaches zero, and its reactance approaches plus or minus infinity. Therefore, most small antennas are inefficient and non-resonant and thus require matching networks. In this investigation, we searched for resonant wire shapes that best utilize the volume within which the antenna is confined [10]. The parameter that best characterizes the performance of a small, resonant antenna is the quality factor, Q. In general, the lower the Q, the more broadband the antenna. A SGA was used to optimize several bent-wire, antenna configurations (over a ground plane), each having from 2 to 10 wire segments. The antennas were optimized for a resonant frequency near 400 MHz and then built and tested. In FIGURE 5, we show a ten-segment antenna that fits in a cube with a side length of 0.037λ. As the cube size deceased from a side length of 0.096λ to 0.026λ, the computed value of Q increased from 15.8 to 590. The measured Q increased from 16.0 to 134 for cubes with side lengths of 0.093λ to 0.037λ. A plot of the Q as a function of antenna size is shown in FIGURE 6.

This process for designing small antennas produced novel, self-resonant antenna configurations. Using intuition, it would seem that the wires should be arranged so that they are orthogonal wherever possible; also, nearly parallel wires that are too close together should be avoided, thus minimizing the transmission-line currents that increase the antenna Q. Upon examining the resultant antenna designs, it seems as though the SGA converged to configurations that incorporate these principles. The preliminary results were very encouraging. These antennas are very inexpensive and can be easily fed from a coaxial line. Since they are self-resonant, the only matching that is required is an impedance transformer.

2.3.3 An Ultra Wideband Antenna

In this experiment, we designed an antenna that operates over very wide bandwidths [11]. A compact antenna consisting of a set of wires connected in series, and with impedance loads inserted in the wires, was designed and then measured. The shape of the antenna and the location of the loads and their impedances were optimized using a SGA. The resultant antenna, shown in Figure 7, was mounted over a ground plane. This antenna is elliptically polarized and demonstrates near hemispherical coverage. It has a VSWR that is under ~4.5

over the 50 to 1 band from 300 to 15000 MHz, as is seen in Figure 8. The VSWR, radiation patterns, and antenna efficiency were simulated, and the VSWR was also measured. It was found that the main limitation on ultra wide bandwidths occurs at the lower frequencies. There does not appear to be a limitation on the VSWR at the higher frequencies. We covered a 50 to 1 band, and we could certainly have extended that band higher. The radiation patterns do, however, become multi-lobed as the frequency is increased, and this is usually not desirable. The antenna efficiency is lowest at the low frequencies but becomes higher with increasing frequency. Future plans are to incorporate the radiation pattern into the objective function of the SGA (in addition to the VSWR).

2.3.4 Genetic Yagi Antennas

The Yagi antenna (shown in Figure 9) evolved as a special configuration of an endfire array. It is a traveling-wave antenna with a surface wave that propagates along the array with a phase velocity slightly less than that of free space. Prof. H.Yagi and his student, S. Uda, first proposed this concept in the late 1920's. The configuration consists of a single, driven element and a number of parasitic elements (consisting of a reflector and a set of directors). The Yagi has been exhaustively investigated, both theoretically and experimentally, for many years. The Yagi has not been amenable to theoretical analysis, since it is an array of elements of different lengths with non-uniform spacing, and thus cannot be treated using conventional array theory. Most analyses have been restricted to relatively short arrays. Progress throughout the years for longer arrays has been slow and has been achieved mostly experimentally and computationally. It is believed that maximum gain is achieved by controlling the phase velocity of the surface wave. The Yagi structure must be designed so that the surface wave is properly retarded. This has been accomplished with some success by logarithmically tapering the elements – the director spacing is gradually increased while the lengths are gradually decreased until they approach constant values at a distance of about 3 or 4 wavelengths from the driven element. Minor changes in the antenna configuration have produced only a small improvement in performance.

To illustrate the concept of a *genetic Yagi* (*e.g.*, the design of an improved Yagi configuration using a SGA), we present two applications. For the first, we chose to maximize the gain of four Yagi antennas with boom lengths ranging from 3.6 to 6.1λ at a frequency of 432 MHz. VSWR was of secondary importance, and back and sidelobes were not included in the optimization [12]. A sketch of the approach that was used to design a Yagi is shown in Figure 10. We compared these designs with those that were obtained using the best design techniques currently available. The gains, radiation patterns, and VSWRs of both conventional and genetic Yagi antennas were computed using the Numerical Electromagnetics Code (NEC). Some of these designs, namely those antenna configurations

having a boom length of 5.16λ, were fabricated and tested. The SGA produced configurations that were quite different from typical Yagis. The conventional Yagi has directors that start out with lengths that gradually decrease and spacing that gradually increases along the array. The genetic Yagis had elements with lengths and spacing that did not show any systematic change along the antenna. Yet, the genetic Yagis had computed gains that ranged from 0.4 to 1.1 dB higher than those of the conventional Yagis at the design frequency of 432 MHz. The measured gain of the genetic Yagi (with a boom length of 5.16λ) was 0.8 dB higher than the corresponding conventional Yagi (as shown in Figure 11).

For the second application, the most important design goal was that the antenna operate over the band from 219 to 251 MHz and have sidelobes and backlobes at least 25 dB down in the azimuth plane for $70° < \phi < 290°$. Of lesser importance was that the E- and H-plane beamwidths be approximately 50°. The desired VSWR was under 3.0, and the desired gain was specified to be consistent with the beamwidth. The feed was mounted over a 1.17-meter square ground plane with a side length of 0.92λ at center frequency. The SGA produced a configuration that was quite different from one that would have been obtained using conventional methods. Typical Yagi designs have directors that are about 0.4λ in length and 0.35λ in spacing; the lengths become slightly shorter and the spacings become slightly larger the further the distance from the driven element. The genetic Yagi had 13 elements (plus the ground plane) with a boom length of only 1.11λ. The directors varied in length from about 0.25λ to 0.4λ with an average spacing of less than 0.1λ. A conventional 14 element Yagi has a boom length about 3 times as long.

2.3.5 Simple Genetic Algorithm Array Pattern Synthesis

In this section, we describe the application of a SGA to an antenna-array, *pattern synthesis* problem. First, we give a brief overview of the problem. Next, we present the experiment and results. Then, we focus our attention on the implementation of the SGA itself, highlighting some of the more interesting parameter choices. Finally, we summarize our findings and conclusions.

Typical operation of an antenna array requires the user to specify a set of complex weights that are applied to the individual elements of the array. The signals emitted from the elements combine constructively and destructively to produce a far-field radiation pattern, which varies as a function of angle. There is a one-to-one mapping between the element excitations and the far-field radiation pattern. For many array geometries, closed-form solutions are available to allow the user to compute the pattern resulting from a particular set of excitations. For many antenna applications, however, we wish to solve the inverse problem. *Pattern synthesis* is the process by which the element excitations are computed

from a given/desired radiation pattern. For this experiment, we used an array composed of elements possessing highly non-linear properties. Our objective was to demonstrate that pattern synthesis could be successfully applied to such nonlinear arrays, thus paving the way for lower cost array systems (*i.e.,* if manufacturers can relax the tolerance specifications for antenna elements and array systems, this will translate to reduced cost).

The experimental configuration, shown in Figure 12, consists of an 8-element, linear array illuminated by a stationary, far-field source. The radiation pattern is measured by rotating the array along θ from –90° to 90°. Each element is an open-ended waveguide connected to a single, 8-bit digital phase shifter. The objective is to find a set of phase-shifter settings that will steer the main beam of the radiation pattern to a specified angle. Figure 13 shows the transfer characteristics for a typical element, including the phase-shifter response curves (both amplitude and phase), as well as the single-element radiation pattern. The dotted curve in each figure represents ideal performance. These highly nonlinear transfer characteristics were our motivation to apply a SGA to this problem. We knew that a simple gradient-descent optimization approach could easily fall victim to local maxima in such a treacherous solution space. We saw the SGA as a robust search mechanism capable of leading us to an acceptable solution. Figure 14 shows experimental results for a case in which it was desired to steer the main beam of the radiation pattern to -10°. This SGA-solution not only steers the main beam to the appropriate angle, but it maintains an 8.5-dB difference between the peak of the main beam and the highest sidelobe level. Although it is possible to achieve –13-dB sidelobes for a linear, 8–element, *ideal* array, we were quite pleased with this result, considering the highly non-linear properties of the experimental elements.

In the following paragraphs, we discuss the details of the SGA implementation, highlighting the more interesting aspects. The chromosome consisted of 64 binary-encoded bits (*i.e.,* 8 phase shifters × 8 bits/phase shifter as shown in Figure 15). Note that we arbitrarily chose to encode the phase shifters sequentially from 1 to 8 along the length of the chromosome. In other words, we could have encoded the phase shifters randomly (*i.e.,* {2 4 7 8 1 3 5 6}), or we could have let the first half of the chromosome consist of the 4 most significant bits from each phase shifter and the latter half consist of the 4 least significant bits. In theory, we could have picked any one of 2^{64} chromosome representations.

We used a somewhat unconventional mating scheme. The initial population consisted of 10 chromosomes. Each parent chromosome mated with *every other* parent to conceive 90 children for a total population of 100. The 10 best chromosomes from each population were then carried over to the next generation where they became the new parents. We realize now that this "forced" evolution may have prevented us from exploring valuable regions of the

solution space. By totally excluding less fit individuals from the mating ritual, we didn't take advantage of all the available genetic information that each population had to offer.

Initially, our recombination operator consisted of 1-point crossover; however, this method imposes a bias on bit position such that bits near the ends of the string have much less of a probability of being traded than bits near the center. By switching to 2-point crossover, the biases on the string ends were eliminated, resulting in an overall decrease in convergence time.

It is worth noting that we also employed a *gray coding* scheme as part of our objective function in an attempt to find higher quality solutions and decrease the convergence time of the algorithm. The term "Hamming distance" is used to describe how many bits are different between two binary representations. For example, the Hamming distance between '0100' and '0101' is '1' – only one bit is different, the last one. The Hamming distance between the binary representations of 7 (seven) '0111' and 8 (eight) '1000' is '4' – all four bits are different. The trouble with most binary representations is that, for example, although the numbers 7 and 8 have consecutive integer representations, they are vastly different in binary representation (*i.e.*, all four bits are flipped). This can cause major problems for a binary-encoded SGA. If a "good" chromosome has a schema or sub-piece with a value of 7, but a chromosome representing the ideal solution needs a value of 8, it's going to require 4 mutations to produce an 8 in that field.

A bit representation is needed in which the Hamming distance between consecutive values is always '1'. In this manner, small amounts of mutation can change parameter values by small amounts, and the genetic algorithm can slowly fine-tune its solution towards the ideal without requiring massive amounts of bit-changes. This type of bit representation, in which the Hamming distance between consecutive representations is always '1' is called a *gray code*. In the case of this particular experiment, there is a further condition required by our binary encoding scheme. Phase represents a cyclic quantity (*i.e.*, it wraps around so that a phase of 360° is the same as 0°). Our gray code representation for phase must be cyclic, in that the Hamming distance between the representations of 360° and 0° is also '1', and therefore transitions across the 0°/360° line do not require massive bit changes. An example of cyclic gray code for 5-bit encoding is shown in Figure 16 (for this experiment we actually used an 8-bit cyclic gray code, but it is easier to demonstrate this concept with fewer bits). Notice that throughout the code, the Hamming distance between consecutive entries is always '1', as is the distance from the very first and last entry.

In summary, this research has demonstrated the successful application of a SGA to an array pattern synthesis problem when the individual elements possess highly nonlinear properties. Despite choosing an arbitrary chromosomal encoding and using an unconventional mating scheme, we were still able to obtain an acceptable solution – no doubt a testament to the robust nature of the SGA as an efficient optimization tool. Although

this experiment was not aimed at analyzing SGA performance quantitatively, we did observe that our switch from 1 to 2-point crossover, as well as our gray encoding of the phase shifters, resulted in a decrease in the overall convergence time of the algorithm.

2.3.6 Antenna Pattern Control of Phased Array Antennas

In this section, we describe two SGA applications related to antenna pattern control of phased-array antennas. The first application is experimental and incorporates measured array output power into the SGA objective function. Two very different types of array antennas are used in these experiments. The second application is a theoretical antenna pattern synthesis problem that incorporates the mean square error between the desired pattern and a predicted (simulated) pattern into the cost function. The first application addresses interference rejection, *i.e.*, the formation of a null (or deep minimum) in the antenna pattern to suppress undesired signals in the antenna sidelobe region [13, 14] and also addresses array beam calibration and beam steering [14]. In both cases, the SGA adaptively adjusts phase shifters to obtain results. For the theoretical problem, a SGA was used to determine the coupling coefficients of quadrature hybrid couplers (used in the constrained feed network of a single subarray of a phased-array antenna) needed to produce a suitable subarray pattern [15, 16].

In the first application, a desired signal enters the main beam, and interfering signals enter the sidelobes. The SGA is implemented on a PC that controls an eight-element cylindrical array [13]. An adaptive, phase-only SGA used in the theoretical simulation of a linear array has been reported previously [17]. We extend that research to an experimental cylindrical array developed by the Air Force Research Laboratory (AFRL) Antenna Technology Branch [13]. Eight active elements of the cylindrical array are connected via a corporate feed to a power combiner. The SGA only has access to the received power out of the power combiner, not the element signals themselves. As shown in Figure 17, each element is connected to a single channel containing an eight-bit phase shifter. There is also an eight-bit attenuator in each channel. Since the attenuator adjustment was not used for nulling, we limit the rest of the discussion to phase control. The attenuators were used to establish a 25-dB Taylor quiescent pattern (without nulling). The four highest order bits of each phase shifter were set to steer the beam to broadside and to quiescently compensate for array curvature and feed-path length difference, *i.e.*, beam collimation (see the paragraph after the next for an explanation of why the four highest order bits were used).

The SGA is coded such that the phase-shifter settings evolve over time until the antenna pattern has a null in the jammer direction. A genetic algorithm was chosen for this problem because it is an efficient method for searching large numbers of phase settings [18]. An adaptive array has 2^{NP} possible phase-shifter settings, where N is the number of elements,

and P is the number of phase-shifter bits used for nulling. Many of these settings correspond to local minima in the output power. For the cylindrical array, $N = 8$ and $P = 4$ (see following paragraph), therefore, there are 2^{32}, or 4.29×10^9, different settings for the nulling bits. Such a large number of potential settings renders random search and gradient-based algorithms impractical.

An adaptive algorithm (in our case a SGA) modifies quantized phase weights based on the measured output power of the array. The goal of the adaptive array is to minimize output power. With no interference, the algorithm would minimize the desired signal. We mitigated this effect by using a limited number of digital phase-shifter bits. Using the four lowest order bits for nulling allows formation of nulls in the sidelobe region without significant impact on the main beam (desired signal). The SGA operates only on the nulling bits, while the higher order bits are left unchanged. The higher order bits perform beam steering and beam collimation functions.

A flow chart of the SGA is shown in Figure 18, where an initial population matrix is filled with random ones and zeros (*i.e.,* parents of first generation). Each row of the matrix (a chromosome) consists of P nulling bits (genes) per element placed side-by-side N times; therefore, there are NP columns and M rows. The output power corresponding to each chromosome is measured and placed in a vector. The process for minimizing measured output power is illustrated in detail (at the bit level) in Figure 19 where, for simplicity, we show a two-element array using $P = 2$ nulling bits and $M = 4$ chromosomes. In the experiment, a relatively small population (*i.e.,* M varied from 12 to 20) performed well. As shown in Figure 19(a), the nulling bits are sent to each element phase shifter, and the array output power corresponding to each chromosome is measured. The chromosomes are ranked from lowest to highest output power in Figure 19(b), and the bottom 50% are discarded. The remaining "best" 50% of the chromosomes are mated pair wise (*i.e.*, one to two, three to four and so forth) to produce the same number of children as parents.

Through mating, we replenish the discarded bottom 50%. In the example of Figure 19(c), mating produces $M/2 = 2$ children (chromosomes). We use 1-point crossover, such that nulling bits to the right and left of a break point are swapped leaving each child with a set of bits (genes) from each parent. For simplicity, in Figure 19(c), we show the breakpoint at the mid-point of each parent chromosome; however, in practice the break point is selected randomly. The two new chromosomes replace the two chromosomes that were discarded. When enough new chromosomes are created to replace the discarded ones, the output power corresponding to each new child chromosome is measured, and all chromosomes are again ranked by power. A small number (less than one percent) of the nulling bits are mutated – toggled randomly from a one to zero or zero to one. Mutation allows the algorithm to try new areas of the search space while still converging to a solution. Note that the "best" phase

setting (corresponding to minimum power) is not altered. This overall process is repeated until an appropriate convergence criterion is met.

The SGA was used to suppress a continuous wave (CW) interfering signal located 45° from the broadside (0°) main beam of the array. The quiescent broadside pattern is shown superimposed on the adapted pattern in Figure 20. The null at 45° is below the receiver noise floor, and the SGA cannot improve it further. We used a total population of $M = 16$ chromosomes, and the algorithm converged quickly (*i.e.*, within only a few iterations). The adapted pattern has increased sidelobes (over quiescent) at –45°, which is characteristic of phase-only nulling.

An identical SGA technique was used to produce a null in the array pattern of a second experimental phased-array antenna. The antenna was designed for a space experiment on a satellite and consisted of a linear array of four subarrays, each controlled by a six-bit phase shifter [14]. An interfering RF source was aligned at 20° off broadside, and a SGA was used to minimize the measured output power of the array. An initial population of 16 chromosomes was used, and the eight poorest performers (highest output powers) were discarded every iteration. GA-generated phase-shifter settings were used to produce the array pattern in Figure 21 with a null 20° off broadside.

Our SGA technique was also used to scan the array main beam. In this case, we *maximized* measured output power. Our objective was to scan the main beam to 15°. When we used phase-shifter settings, which were determined from the theoretical phase gradient across the array, in conjunction with a look-up table of phase shift versus digital control word, we obtained a poor quality pattern with very high sidelobes. As an alternative, we simply rotated the array off broadside by 15° and maximized output power. The GA phase-shifter settings resulted in the improved array pattern shown in Figure 22.

The second SGA application involved the design of a constrained feed network. The feed network determines the antenna pattern of a subarray, which is part of a larger phased-array antenna [15, 16]. The SGA cost function is illustrated in Figure 23 and is the sum of the shaded areas. The SGA seeks to minimize these differences. The synthesized (simulated) pattern is a function of the four coupling coefficients (there are eight coupling coefficients; however, only four are independent) in a set of hybrid couplers used in the network. Each coupling coefficient is parameterized as a digital word, and the four digital words (one for each coupling coefficient) form a chromosome. The SGA was used to obtain coupling coefficients that resulted in an acceptable subarray antenna pattern as shown in Figure 24.

2.3.7 Advances in Electrically Small Antennas

After we had compared some of our 1st-generation electrically small antennas (see Section 2.3.2) [19–21] to the conventional electrically small spiral helix, it became evident that we had only succeeded up to a certain point. The goal of the genetic, electrically-small, bent-wire antenna effort had been to create a single, bent-wire antenna structure having the lowest possible VSWR when matched to 50Ω for a given cube size. We had also imposed the following additional constraints: the antenna could consist of only a single wire, and pieces of the antenna within the cube could not touch the ground plane or each other.

At cube sizes of 0.04λ and smaller, we discovered that certain conventional antennas, such as the spiral helix, provided a better (lower) VSWR than our best *genetic antennas* designed up to this point. Visually, there was also a striking difference – the helixes were smooth and curvy, while our 1st-generation genetic antennas were straight-lined and angular. While one might argue that simply adding more wire segments to the structural representation of the antenna would allow for the automatic generation of more-curvy geometries, this was not the case. Altshuler and Linden reported in [19,21] that the addition of more wires did not improve the solution and, in fact, degraded it.

It became clear that we had not reached the limit of electrically small antenna design; rather, our SGA had become limited in some manner. This prompted us to investigate different chromosomal encodings/representations in an attempt to obtain better solutions to this design problem.

2.3.7.1 Coordinate Chromosome Representation

Our original method (Section 2.3.2) of encoding the antenna design was such that the antenna structure was represented as a fixed number of straight wire segments connected in series at their endpoints (called "nodes"). The x, y, and z Cartesian coordinates for each node were encoded along the chromosome, as shown in Figure 25. This encoding scheme originally used 5-bits for each coordinate. However, in subsequent research, a real-valued SGA was utilized, and the coordinates were represented by positive real values [19].

2.3.7.2 Angular Chromosome Representations

As an alternative way of encoding this problem, we modeled the antenna as a single piece of wire, subdivided into many fixed-length, straight segments. The angular orientation of each segment was encoded into the chromosome. There were two obvious ways to represent these angular orientations: in an ***absolute*** (*i.e.*, global) fixed spherical coordinate

system (Figure 26) or ***relative*** (*i.e.*, local) to a spherical coordinate system centered on the vector orientation of the previous wire segment, with the z-axis lying along the segment, as shown in Figure 27[1].

There are intuitive pros and cons to each of these sub-models. One might consider that there exists a strong relationship between the electrical properties of a wire segment and its orientation to the ground plane (and hence its mirror image). This would favor the absolute angle representation. However, during mating (or *recombination),* when a piece of one chromosome (or antenna) is merged with pieces of another in order to create children for the next generation, the building block might be better preserved in the relative angle representation, rather than in absolute terms.

Rather than implementing a real-valued chromosome, we used a cyclic, gray-code, binary representation. This had the benefit of allowing standard, binary, recombination operators, while eliminating the Hamming cliffs that occur in traditional binary representations. An additional benefit of a cyclic gray-code is the seamless cross-over in angular representation from the highest to the lowest values, *i.e.,* the branch cut at 0°/360° (See section 2.3.5 for a detailed description of cyclic, binary gray code).

2.3.7.3 Comparison of Chromosome Limitations

While a full discussion of the pros and cons of each approach may be found in [22], we provide a short summary here to illustrate the importance of a good chromosomal encoding scheme.

First, not all of these representations are capable of representing all possible bent-wire antennas. For both Cartesian and spherical representations (absolute and relative angle), the number of straight wire pieces needs to be determined *a priori.* While one may argue that if a sufficiently large number of small pieces were used, a general solution would be possible, our experimentation showed that only the relative angle representation benefited from using many small pieces. The performance of both the Cartesian and absolute angle representations deteriorated when the number of nodes or segments exceeded a relatively small number (like 7-12 pieces). The relative angle representation, however, was successful with pieces as small as we were able to model; hence, this representation was best able to model antennas closest to a general solution.

Also notable is the fact that the angular representations require the antenna pieces to be of equal-length and the total antenna length to be determined *a priori,* while the Cartesian representation does not fix the length of the wire pieces or the total length of the antenna.

[1] For both sub-models, the first azimuth angle was removed from the chromosome and fixed to be zero, to avoid competing identical solutions rotated around the z-axis. Also, for the relative angle chromosome, the first segment was represented relative to the z-axis of an initial reference frame.

These limitations could both be mitigated by expanding the angular chromosomes to include segment length information (which may be pursued in the future). However, note that equal-length segments are only a very minor limitation when many very-small wire pieces are used: short pieces of the antenna structure may be represented by only one or two of the short segments, while longer pieces are created by aligning the vectors of many short segments. Aside from the above discussion, other comparisons and limitations concerning these three representations can be made and are presented more completely in [23].

2.3.7.4 Results for Electrically-Small, Bent-Wire Chromosomes

In Figure 28, we compare the performance of the three chromosomal encodings. We see that the absolute angle and Cartesian representations performed similarly, but that the relative angle representation, able to use many small pieces effectively, created curvy antenna structures, which better met our low VSWR criterion (for a given cube-size). When compared to the electrically-small, normal-mode helix, it is clear that the relative angle representation, as presently modeled with a fixed wire length and fixed segment size, is still not optimal, but is performing much better than previously. In fact, when antennas created by this method are "smoothed" by hand (*i.e.*, creating non-equal segment sizes), even better performance is obtained.

Although these very small antennas are difficult to create by hand, the largest one was built and measured (shown in Figure 29) to verify our simulations. As shown in Figure 28, it performed as expected.

2.3.8 Hybrid DISS Transmit Antenna

We recently had another opportunity to transfer our research expertise in genetic antenna design from the laboratory to the operational Air Force when we learned that the Battlespace Environment Division of the AFRL Space Vehicles Directorate was looking at a redesign of a transmit antenna used for ionospheric measurements. The Digital Ionospheric Sounding System (DISS) network is operated by the US Air Force Weather Agency (AFWA) and the Air Force Research Laboratory (AFRL). Its purpose is to observe the global ionosphere in real time. Eighteen digital ionosondes are currently deployed worldwide by the Air Force to provide data for many products, including specification and forecasts of primary and secondary HF radio propagation characteristics; ionospheric electron density, and total electron content; ionospheric scintillation; environmental conditions for spacecraft anomalies; and sunspot number.

DISS was originally built using an off-the-shelf TCI model 613F communications antenna (shown in Figure 30). This antenna transmits radio signals of different frequencies

across a specified sweep (2-30 MHz) in a vertical direction; these are then reflected, absorbed, or distorted by the ionosphere. Co-located receive antennas intercept the returning signals for algorithmic processing. The current transmit antenna does not exhibit a consistent gain in the vertical direction for all desired frequencies (Figure 30), although the VSWR matched at 450Ω is excellent across the entire frequency band. Our goal was to determine if a simple hybrid augmentation to the antenna could increase gain across the bandwidth without prohibitive costs.

2.3.8.1 Genetic DISS Hybrid Antenna

Conventional antenna wisdom suggested the addition of a six-element, log-periodic dipole array (LPDA) as a complement to the existing TCI antenna. Although this hybrid looked promising in simulation, it proved cost prohibitive and mechanically difficult to create and assemble. We therefore decided to augment the TCI with six new pairs of wire elements, with one end of each wire connected to the feed line running up the tower, and the other end open and angled towards one of two new ground stakes. To reduce complexity, we used only four additional stakes, two per side, leading to the hybrid configuration in Figure 31. Following the active wire length, we transitioned to an insulator to complete the mechanical connection. The three highest new wires (L2, L3, and L4) are connected to stake 1, and the lowest three wires (L5, L6, and L7) are connected to stake 2. The distances of the stakes from the tower and their azimuth angles relative to the TCI antenna were optimized, in addition to the active lengths of each wire and the height of the wire on the tower. We constrained the distance of the stakes to a reasonable value and set a minimum distance criterion between all wires to avoid sparking conditions.

We modified the antenna hybrid and corresponding chromosome over time as our genetic optimizations revealed strong and weak points in our model and as construction requirements changed. We had initially connected the highest wire (L2) to the edge of the TCI curtain; however, our SGA continually minimized the length of this wire to almost zero, removing it or rendering it only viable for high frequency contributions. We subsequently connected L2 to stake 1 (with L3 and L4) and achieved much better performance. We also did not know whether all of the existing TCI antenna wires were needed or if only a subset would suffice; however, our optimizations achieved best performance when the entire TCI curtain was used. We initially varied the load resistance from the original 600Ω resistors to a variety of values. However, while our optimizations determined that a higher resistance (1350Ω) increased the overall performance, these were not readily available for our power requirements, so we decided to either maintain the existing resistors or remove them entirely. When a new operational constraint was introduced, which required the antenna to be well-behaved down to 1MHz, the option of removing the resistors was discarded. We also

compared feeding the new antenna elements as 1) six independent bent dipoles, 2) two pairs of 3-element LPDAs or 3) as a six-element LPDA. We found that the independent bent dipole configurations performed similarly to the LPDA-wired structures and were easier to model and build.

2.3.8.2 Genetic Algorithm and Optimization Goals

For this antenna optimization problem, a simple genetic algorithm proved sufficient in obtaining a good solution. We encoded the chromosome (shown in Figure 32) using a binary, cyclic gray-code in order to avoid the Hamming cliffs that are typically present in binary representations (see Section 2.3.5). One could argue that a real-valued GA may have provided greater accuracy; however, when considering our wire lengths (in terms of wavelengths for the given frequency band) and our field-construction tolerances, the granularity of a binary representation was not a limiting factor. Our cost function consisted of a weighted sum of the following: the total effective gain error (*i.e.*, any gain < 3 dBi) and the total VSWR error (*i.e.*, any VSWR > 3) across the entire bandwidth; the standard deviation of the gain error; and the standard deviation of the VSWR error. We included the standard deviation to avoid large error spikes in either effective gain or VSWR. As shown in Figure 30, using this figure of merit, the TCI antenna scored ~305.

2.3.8.3 Results

In Figure 32, we show a DISS transmit antenna resulting from the genetic algorithm. With the 600Ω resistors permanently included, our optimization algorithm converged to this basic shape, with minor variations as we balanced the cost function by adjusting the relative weighting between gain and VSWR error. This solution yields a similar structure to the LPDA-augmented TCI in that the new elements are orthogonal in nature. However, where the former had lower-frequency elements higher up, the genetically designed hybrid places them lower on the tower. The score of this genetic DISS hybrid greatly exceeds the simulated scores of the TCI, with or without LPDA augmentation, and this hybrid antenna structure should prove to be an easy retrofit. In March 2004, a genetic DISS hybrid transmit antenna replaced the existing operational transmit antenna of the DISS station on Ascension Island. Initial measurements taken by the AFRL Space Weather Center of Excellence with this antenna confirm that this new design performs up to 10dB better than the original transmit antenna!

2.3.9 Technology Transfer to Other Applications

While a full review of other GA applications that have resulted from our genetic antenna research is beyond the scope of this paper, we present here one example of a technology transfer that resulted directly from our research and our experience with gray codes and our application of SGAs to antenna pattern synthesis.

2.3.9.1 Reflector Dish Array Pattern Synthesis for RCS Measurements

In 1998-99, scientists from ARCON Corporation (Waltham, MA) successfully applied a SGA to an array pattern synthesis problem in which gray codes were used to (see Section 2.3.5) represent phase. This SBIR (Small Business Innovative Research) project supported the 46th Test Group (RATSCAT) at Holloman AFB. Among other antenna initiatives, one goal of the SBIR was to determine if a vertical reflector array could be configured, on both transmit and receive, to create a flat pattern across a target area for radar cross section (RCS) measurements as shown in Figure 33. This pattern was particularly difficult to synthesize, both because of time-dependent, ground-bounce range characteristics and the need to have as little energy as possible reflected off the support pylon.

The genetic algorithm optimization determined, at X-band, the best relative amplitudes and phases, heights above the ground, and orientations (*i.e.*, bore sights) of three identical reflector dishes to produce a desired flat far-field power taper across the target zone at the pylon range gate. The parameters encoded in the chromosome included the three amplitudes of the reflector dish feeds; two phase differences referencing dishes #2 and #3 to dish #1; the heights of the three dish centers; and the bore sight angle deviations within +/- 8^{o} off a normal bore sight. While a perfect null across the entire pylon area was physically impossible, the genetic algorithm did produce reflector array configurations which successfully created flat power tapers across the target area, as shown in Figure 34, while minimizing power on the pylon [24].

2.4 Conclusions

This section has demonstrated the successful application of the SGA to several antenna design and optimization problems covering diverse areas of interest – no doubt, a testament to the robust nature of the SGA as an efficient optimization tool. For some of these projects, however, a great deal of fiddling with SGA parameters and operators was necessary in order to achieve acceptable, high quality solutions. In particular, we found that the

chromosomal encoding scheme has a large impact on the solution quality, as well as the convergence time. Thus, one of the major drawbacks of using a SGA is that it provides no method for matching coding and operators.

The next section investigates a new breed of GA, namely a *competent* GA, for which genetic operators are no longer arbitrary. In fact, the competent GA is designed to learn the correlations between different sets of variables encoded along the chromosome. In this manner, the recombination operator can be designed to exploit these correlations and minimize the probability of disrupting good "building blocks" while maximizing the probability of finding a global solution to the problem.

3.0 Optimization of a Constrained Feed Network for a Linear Array: SGA vs. Competent GA

3.1 Introduction

This section describes the optimization of a constrained feed network for a linear array. Both a simple and competent GA were applied to this problem in an effort to find a solution that meets the system requirements/specifications. Section 3.2 starts by briefly reviewing the design theory and principles necessary for the invention and implementation of fast, scalable genetic algorithms. A particular procedure, the hierarchical Bayesian optimization algorithm (hBOA) is then briefly outlined.

Section 3.3 begins with an overview of the optimization problem, including the antenna system parameters and specifications. Next, we discuss the details of each approach – SGA vs. hBOA. Here, we describe how each algorithm was applied to the problem. Then we describe three variations of the objective function used in this experiment. Following this, we present the results, demonstrating that the problem was sufficiently difficult that acceptable solutions were not obtainable using the SGA (regardless of the objective function used). The competent GA, however, was able to obtain an acceptable solution for at least one of the objective functions. We analyze the performance of the SGA and hBOA for all three cases in order to gain insight into the inherent strengths and weaknesses of these approaches.

Finally, in Section 3.4, we summarize our results and draw some interesting conclusions concerning the fundamental differences between simple and competent GAs.

3.2 Introduction to hBOA

The hierarchical Bayesian optimization algorithm (hBOA) [3-6] is one of the most successful, probabilistic, model-building GAs empowered by Bayesian learning. It has been

shown that hBOA is capable of solving boundedly difficult problems within a sub-quadratic number of function evaluations. This subsection provides a brief description of hBOA. To better understand the motivation behind hBOA, we first give an introduction to GA design theory and competent GAs. Subsequently, three primary components of hBOA are discussed, namely (1) the Bayesian optimization algorithm [6,25], (2) learning Bayesian networks, and (3) conquering hierarchical difficulty [6].

3.2.1 GA Design Theory and Competent GAs

In section 2, we introduced the simple genetic algorithm (SGA) and witnessed its robust optimization capabilities in a variety of practical antenna problems. A SGA, however, does not always address the issue of *linkage* (*i.e.,* chromosomal encoding) adequately. This was most evident in the electrically-small antenna design of Section 2.3.7 in which different chromosomal encodings led to very different solutions. This section gives an introduction to *competent* GAs – advanced optimization techniques designed to determine the optimal linkage of bounded, difficult problems in order to find a global, or near global solution within a sub-quadratic number of function evaluations.

To better understand the concept of boundedly difficult problems, let us first consider two extreme cases: the OneMax problem and the needle-in-a-haystack problem (NIAH). In the OneMax problem, the fitness value of a binary chromosome is defined as the number of ones in the chromosome (the chromosome with the highest fitness value would be of all ones); in NIAH, the fitness value of some specific chromosome is highest (*e.g.*, 1), and those of all other chromosomes are equally low (*e.g.*, 0). On one hand, the genes in the OneMax problem are independent to each others, and hence the OneMax problem is fully decomposable. It is considered to be a GA-simple problem, and a SGA can easily solve it within a sub-quadratic number of function evaluations. At the other extreme, the order of linkage in the NIAH problem is equal to the length of chromosome, and hence the NIAH problem is not decomposable. The NIAH problem is considered to be GA-difficult, and it has been shown that no algorithm can do any better than a random search for this type of problem. On average, it requires an exponential number of function evaluations to find the optimal solution to the NIAH problem. So we ask ourselves, how efficiently can we solve a problem that falls somewhere between these two extremes? For example, suppose that we have an additive NIAH problem, which is merely a concatenation of several bounded-order NIAHs. Can we find the optimum quickly? More generally, can we design GAs that solve such nearly decomposable problems quickly, reliably, and accurately?

Using Holland's notion [26] of a building block, a decomposition methodology has been proposed for the successful design of competent GAs [2,27,28]:

1. Know what GAs process—building blocks (BBs).
2. Know thy BB challenges—BB-wise difficult problems.
3. Ensure an adequate supply of raw BBs.
4. Ensure increased market share for superior BBs.
5. Know BB takeover and convergence times.
6. Make decisions well among competing BBs.
7. Mix BBs well.

Much work has been done investigating each of these critical categories, including problem difficulty [29,30], adequate supply [31,32], decision making [33,34], and mixing [35].

Of all the categories mentioned above, effective mixing has been found to be one of the most essential and challenging issues for GA success, and efforts have been put forth in developing competent GAs. A number of competent GAs are described in [2] and [36]. These differ primarily in two aspects: (1) the strength of the identification of important sub-solutions, and (2) the recombination of those important subsolutions into promising solutions. In this paper, we used hBOA [3-6], one of the most successful competent GAs to date, to optimize an antenna system. hBOA is described in the next few subsections.

3.2.2 The Bayesian Optimization Algorithm (BOA)

For a given problem, the Bayesian optimization algorithm (BOA) [6,25] evolves a population of candidate solutions by building and sampling Bayesian networks [37]. The BOA first generates a population of candidate solutions either randomly or according to some prior knowledge of the given problem. Subsequently, the population is updated each generation via the following four steps:

(1) Promising candidate solutions are selected using a GA-selection operator, such as tournament selection or truncation selection.
(2) A Bayesian network is built to estimate the distribution of those promising candidate solutions that were selected by selection.
(3) New candidate solutions are generated by sampling the Bayesian network.
(4) The population for the next generation is generated by incorporating the new candidate solutions into the original population, replacing part or all of it.

The above four steps repeatedly update the population of candidate solutions until some termination criteria are met. For example, the optimization procedure can be terminated when the population contains an acceptable solution to the problem, a bound on the number of generations is reached, or a bound on the overall computational time is reached.

3.2.3 Learning Bayesian Networks

This subsection briefly describes how the BOA constructs Bayesian networks during evolution. A Bayesian network is a network that describes the following Bayesian, joint-probability distribution.

$$p(X) = \prod_{i=1}^{n} p(X_i \mid \Omega_i), \tag{1}$$

where $X = (X_1, X_2, \ldots, X_n)$ is a vector of all variables in the given problem, Ω_i is a set of X_i, and $p(X_i \mid \Omega_i)$ is the conditional probability of X_i given Ω_i.

A Bayesian network can be visualized as a directed, acyclic graph, where the nodes represent variables and the edges represent conditional dependencies. For instance, the following Bayesian joint probability can be expressed by the graph shown in Figure 35.

$$p(A,B,C,D,E) = p(A)p(B)p(C \mid A)p(D \mid A,B)p(E \mid C,D). \tag{2}$$

Learning a Bayesian network consists of two subtasks: (1) learning the structure, and (2) calculating the conditional probabilities. In BOA, calculating the conditional probabilities for a given structure is straightforward, because the value of each variable in the population is specified. The maximum likelihood of the conditional probabilities can be obtained by simply calculating the relative frequencies observed in the population. To learn the structure, the current version of BOA adopts a minimum-description-length scoring metric and a greedy algorithm to search for the structure with the minimal description length. Initially, the structure is a graph with no edges. The greedy algorithm then updates the graph by one of the following three operators: (1) edge addition, (2) edge removal, or (3) edge reversal. This is done under the guidance of the scoring metric. The greedy algorithm terminates when no improvement can be made.

3.2.4 Hierarchical Decomposition – From BOA to hBOA

Hierarchical structures have appeared in real-world systems and real-world problems [38]. By hierarchy, we mean that interactions between higher levels are not revealed until interactions between lower levels have been recognized. We humans often utilize hierarchical decomposition to solve problems in either a bottom-up or top-down manner. Hierarchical decomposition adds a new level of decomposition that reduces the problem difficulty, and hence enables us to solve more difficult problems.

Pelikan [6] recognized three key issues regarding hierarchy success:

(1) Proper decomposition. At each level, the algorithm needs to be capable of properly decomposing the problem. A proper decomposition reduces the problem complexity and hence improves the scalability of the algorithm.
(2) Chunking. Each sub-solution in a lower level can be seen as a chunk. The algorithm should be capable of properly representing each chunk as one single variable when solving the next upper level.
(3) Preservation of alternative candidate solutions. Since the interactions at a higher level do not reveal themselves until the interactions at the lower levels have been recognized, preserving alternative candidate solutions is important for a hierarchical problem solver.

By recognizing these three issues, the hierarchical BOA (hBOA) was constructed as follows. hBOA utilizes BOA to decompose the problem at each level. Chunking is incorporated by recognizing the local structures in the Bayesian networks. One of the niching techniques, called *restricted tournament replacement* [3,6] is adopted in hBOA to preserve alternative candidate solutions. hBOA has been shown to have a stronger ability than BOA to conquer hierarchical difficulties [3,6,39].

3.3 Optimization of a Constrained Feed Network for a Linear Array

3.3.1 Problem Statement

This section describes an antenna system designed for space-based and airborne radar applications. The goal of this system is to produce a far-field radiation pattern having at least –30-dB sidelobes over a 20% bandwidth. This is accomplished by implementing an optimized, constrained feed network. The following overview is intended to provide the reader with enough background information to understand the details of the system optimization. For further information about the system design and implementation, the reader is referred to [40].

The following discussion assumes an ideal system. A single section of the system is shown in Figure 36. At the front end of the section is an N-element, linear array. It can be shown that when the array is illuminated by a plane wave, the element excitations can be computed via the following [41]:

$$a_n(\theta) = e^{j\frac{2\pi}{\lambda} nd \sin\theta}, \qquad (3)$$

where n is the element index, λ is the wavelength of the incoming plane wave [in meters], d is the inter-element spacing [in meters], and θ is the angle of incidence with respect to the normal.

Each element is connected to a single input port of an N by M Rotman lens. Thus, the element excitations become the inputs to the Rotman lens. It can be shown that the Rotman lens output signals are described by the following:

$$I_i(\theta) = \sum_{n=1}^{N} e^{j\frac{2\pi}{\lambda} nd\left(\sin\theta - \frac{i}{N}\right)}, \qquad (4)$$

where i is the Rotman lens output index. Each output signal is then multiplied by a complex weight, w_i.

Next, these signals are input to an M by M Butler matrix, the outputs of which are computed from:

$$J_m(\theta) = \sum_{i=1}^{M} I_i(\theta) w_i e^{j2\pi\left(\frac{m}{M}\right)\left(\frac{\lambda_0}{\lambda}\right)}, \qquad (5)$$

where m is the Butler-matrix output index, and λ_0 is the center-frequency wavelength of the system. The center $M/2$ output signals from each of P sections are time-delayed, weighted (*e.g.* fixed weights like a Taylor distribution, *etc.*), and combined to compute the final radiation pattern of the system.

In order to optimize the system, the set of complex weights, w_i, must be determined for each of P sections, such that the final radiation pattern exhibits -30-dB sidelobes over a 20% bandwidth. For our particular system, the specifications and parameter values are as follows:

- Frequency band of operation: 9.0 – 11.0 GHz
- Center frequency f_0 = 10.0 GHz (where $\lambda_0 = c/f_0$, and c = velocity of electromagnetic waves in free space = 3×10^8 m/s)
- $N = 64$
- $d = 0.5\lambda_0$
- $M = 8$
- $P = 3$.

Figure 37 shows far-field radiation patterns for an ideal system at 9.0, 10.0, and 11.0 GHz. The x-axis represents *u-space* (where $u = \sin[\theta]$), and the y-axis shows the amplitude of the pattern measured in decibels (dB). The peak of each pattern is normalized to 0 dB. This system was optimized for a beam-steering angle of 45° (*i.e.*, $u = 0.7071$) in [40] using the method of alternating projections. The weights, w_i, were assumed to be real-valued and that weights for a particular index i were assumed identical across the three sections (*i.e.*, w_5 for section 1 = w_5 for section 2 = w_5 for section 3). Note that when these weights are applied to the ideal system, the maximum sidelobe level is well below –30 dB across the entire 20% bandwidth, thus meeting the system requirements.

We built three Rotman lenses and measured the transfer function for each of them. The inverse-transfer-function of a single lens (for phase only) is plotted in Figure 38. The x-axis represents the input ports (as seen in Figure 36), and the y-axis measures the unwrapped phase in degrees. Each shape represents a different output port, where the solid line is ideal data and the dotted line is experimental data. For example, when output port 1 is excited, the resulting ideal phase distribution across the input ports is denoted by the solid curve (squares), whereas the experimental phase distribution is represented by the dotted curve (squares). (We show the *inverse* transfer function here rather than the forward transfer function, because it is easier to analyze and visualize 8 curves versus 64). For comparison purposes, each phase curve is normalized to 0° at a fictitious input port that lies exactly halfway between ports 32 and 33. Note that the experimental phase differs from the ideal by as much as 40°.

These phase deviations between the ideal and experimental Rotman lens transfer functions have a detrimental effect on the system radiation patterns, as illustrated in Figure 39. For these plots, we replaced the ideal Rotman lens data with the experimental data and applied the same weights, w_i, found in [40] for the ideal system. Not only is the integrity of the main beam compromised, but also the maximum sidelobe level well exceeds the –30-dB limit across the entire frequency band. In other words, the weights that we successfully implemented for the ideal system cause the current system to break down. Thus, we need to re-optimize the system now that the experimental Rotman lens data is incorporated into the system model. The following section describes our approach to this problem.

3.3.2 Approach

3.3.2.1 Implementation of the Simple Genetic Algorithm (SGA)

This section describes our first optimization approach, which involves the implementation of a simple genetic algorithm (SGA). Recall that our antenna system consists of three sections, each containing eight complex weights (*i.e.*, amplitude and phase). Our

goal then is to obtain a set of 24 complex weights that will allow us to meet the system requirements outlined in Sec. 3.3.1.

We used the chromosome representation shown in Figure 40. We chose a binary encoding scheme and chose to represent each complex weight with 16 bits (*i.e.*, 8 bits amplitude and phase). Therefore, the length of our chromosome was: 24 weights × 2 components/weight × 8 bits/component = 384 bits. We arbitrarily chose to encode the complex-weight amplitudes along the first half of the chromosome and the phases along the latter half. Both the amplitudes and phases are numbered sequentially along their respective halves of the chromosome. We restricted the amplitudes to lie in the interval [0 1] and the phases to lie in the interval [0° 360°]. It is also worth noting that we used an 8-bit gray code for both the amplitude and phase encoding schemes (for more details about gray coding, see Section 2.3.5)

At the start of the algorithm, we formed a random population of 200 chromosomes (parents). Each member of the population was evaluated and ranked (the details of the objective function will be described in Sec. 3.3.2.3). Then, we formed a mating pool of 200 individuals via binary tournament selection (*n*-tournament selection is when *n* individuals are randomly selected from the population, and the one with the highest fitness is selected for the mating pool [1]). Next, two individuals from the mating pool (*i.e., parents*) were chosen randomly to create a *child* via two-point crossover. This process was repeated until 200 children had been generated. Each child was passed to an operator having a constant mutation rate of 0.005. The children were evaluated, ranked, and proceeded to become the parents of the next generation. For this experiment, we always ran the SGA for 5,000 iterations with a constant population size of 200 for a total of 1-million objective-function evaluations.

3.3.2.2 Implementation of the Hierarchical Bayesian Optimization Algorithm (hBOA)

This section describes the implementation and parameter settings of our second optimization approach—hBOA. The encoding scheme is exactly the same as that used for the SGA. Each candidate solution was encoded into a 384-bit binary string. Also, gray coding was used. The population size was set to 5000, and the maximum number of generations was set to 200 for a total of 1-million function evaluations (*i.e.*, same as SGA case).

Restricted tournament selection was used. The window size was set to the problem size (384) to perform niching globally; the tournament size was set to 12 based on empirical observations. Bit-wise mutation was used, with a mutation probability of 0.005. The maximum number of incoming edges for a single node in the Bayesian network was limited to 4 to avoid unnecessary linkage complexity. Elitism was adopted. In each generation,

parental candidate solutions were evaluated, and the bottom half were replaced by newly generated offspring.

3.3.2.3 Objective Function

The objective function is essentially a subroutine written in MATLAB, which was used by both the SGA and hBOA to evaluate potential solutions to the problem (*i.e.*, chromosomes). When the user inputs a set of 24 complex weights, the subroutine computes the corresponding far-field radiation patterns for five discrete frequencies (9.0, 9.48, 10.0, 10.52, and 11.0 GHz). This experiment employed three variations of the objective function, which are described below as Cases 1, 2, and 3.

Figure 41 shows the objective function for Case 1. The pink curve is a typical far-field radiation pattern produced by the system for a given frequency and set of complex weights. The x-axis represents u-space (*i.e.*, $\sin[\theta]$), and the y-axis measures the normalized amplitude of the pattern in decibels. The black "mask" represents the objective function, showing a *main-beam* and *sidelobe* region. For this case, we perform a point-by-point subtraction of the mask from the pattern. For a given frequency and set of complex weights, an error value E_k is computed by calculating the mean sum of the squared differences between the pattern and mask:

$$E_k(w, f_k) = \frac{1}{U}\sum_{i=1}^{U}[pattern_i - mask_i]^2), \qquad (6)$$

where w represents the vector of complex weights, f_k is the k^{th} discrete frequency, and U represents the total number of points in the radiation pattern. Note that no penalty is administered when the pattern lies below the mask in the sidelobe region (*i.e.*, if the difference between the pattern and mask is negative, it is not used in the computation). In essence, we're trying to force the pattern to conform to the mask in the main-beam region while forcing the pattern to lie below the mask in the sidelobe region. Also note that we are "overshooting" by trying to force the algorithm to find a solution with –40-dB sidelobes in hopes that it will at least be able to obtain –30-dB sidelobes. This lack of efficiency is an inherent weakness of this approach. The overall Case 1 fitness value, $F_1(w)$, is the average of the error across the entire frequency band:

$$F_1(w) = \frac{1}{K}\sum_{k}^{K} E_k, \qquad (7)$$

where K is the total number of discrete frequencies (and is equal to 5 for our experiment).

Figure 42 shows the general objective function used for both Cases 2 and 3. For Case 2, for a given frequency and set of complex weights, the error has two components, the first of which is as follows:

$$E_{k,1}(w, f_k) = [1 - pattern(u_0)]^2, \tag{8}$$

where u_0 is the desired steering angle of the pattern peak ($u_0 = 0.7071$, corresponding to $\theta = 45°$, for this experiment). In essence we need to ensure that the peak of the normalized pattern in the main-beam region coincides with the desired steering angle u_0. The second error component is as follows:

$$E_{k,2}(w, f_k) = [1 - (1 - MSL)]^2, \tag{9}$$

where *MSL* refers to the "maximum sidelobe level" (*i.e.,* the maximum level of the radiation pattern in the sidelobe region). In other words, we're trying to maximize the difference between the normalized pattern peak and the maximum sidelobe level as illustrated in the Figure 42. The overall Case 2 fitness value, $F_2(w)$, is the mean summation of the error components across the entire frequency band:

$$F_2(w) = \frac{1}{K}\sum_{k=1}^{K}(E_{k,1} + E_{k,2}). \tag{10}$$

It is clear that the objective function for Case 2 involves only two subtractions, rather than a point-by-point comparison of the pattern to the mask – this property renders Case 2 more computationally efficient than Case 1. Similar to Case 1, however, the overall fitness value for a given set of complex weights is the average of the error across the entire frequency band.

Case 3 is identical to Case 2, except the overall fitness value, $F_3(w)$, is equal to the maximum error across frequency:

$$F_3(w) = \max_k(E_{k,1} + E_{k,2}) \tag{11}$$

In other words, Cases 1 and 2 are aimed at minimizing the *mean* error across frequency, whereas Case 3 minimizes the *maximum* error across frequency. Of the three objective functions, Case 3 is the most relevant to our particular problem, since we are ultimately trying to minimize the maximum sidelobe level across frequency.

3.3.3 Results: SGA vs. hBOA

This section analyzes the results for Cases 1, 2, and 3 by considering the computational performance of SGA versus hBOA. We also compare SGA and hBOA on the basis of sidelobe attenuation.

Each case was run three times for both the SGA and hBOA, the results of which are tabulated below. For convenience, the runs are sorted according to fitness from best to worst. (Note that for this experiment, we defined fitness such that *lower* values correspond to higher quality solutions. Traditionally, fitness is defined such that *higher* values correspond to higher quality solutions). Table 1 contains the results for Case 1. If we compare the performance of the *best* run for each algorithm (*i.e.,* Run 1), we see that the overall fitness value F_1 is practically identical for the SGA and hBOA. In addition, Figure 43 shows that there is virtually no difference between the maximum sidelobe levels of the resulting radiation patterns at 9.0 GHz. Thus, one may be tempted to assume that both algorithms perform equally well for this objective function. However, it is also evident from the table that the *mean* fitness across runs is 33% higher for the SGA compared to hBOA, and the standard deviation of the fitness across runs for the SGA exceeds that for hBOA by a factor of 175! These results imply that, given enough runs, the best-quality SGA solution may be comparable to the best-quality hBOA solution; however, hBOA seems to be a much more consistent and reliable search mechanism.

Case 2 (Table 2 and Figure 44) exhibits the same general trends as seen for Case 1, namely the mean of the fitness F_2 across runs is 34% greater for the SGA vs. hBOA, and the standard deviation of F_2 across runs for the SGA exceeds that for hBOA by almost a factor of 4. However, for this case, the best-quality hBOA solution is 28% better than the best-quality SGA solution. This is apparent in the figure, where we see that the highest sidelobe resulting from the SGA solution exceeds that of hBOA by approximately 2.5 dB. Thus, for this case, the SGA performance trails that of hBOA for *both* best fitness and average fitness.

The performance of the SGA is by far the worst for Case 3 (Table 3 and Figure 45). hBOA outperforms the SGA by several orders of magnitude when comparing both best fitness and average fitness. In addition, the standard deviation of the fitness F_3 across runs is zero for hBOA! The sidelobe attenuation of hBOA solution is -27dB, which is quite close to the desired value of -30dB. By contrast, in the best SGA solution, it is difficult to identify a mainlobe and worst-case sidelobes exist at -6dB.

Overall, these results are not surprising. The different objective functions represent drastically different solution spaces. Case 1 involves forcing a function to a mask, which is considered a GA-easy problem because taking average makes the fitness function landscape smooth. Thus, we see that in this case the SGA performs comparably to hBOA. Cases 2 and

3 are considered GA-difficult, because the min/max nature of the objectives give rise to a solution space that contains many local minima. The SGA, therefore, easily fell into some local minimum and was not capable of exploring the landscape globally. hBOA, on the other hand, was able to better identify the linkage of the problem, which allowed it to recombine salient pieces of information without disrupting good building blocks. The solutions obtained from hBOA for Case 3 correspond to –27–dB sidelobes across the entire frequency band of operation. Thus, hBOA allowed us to come within 3 dB of our goal (*i.e.*, -30 dB across frequency). We are quite pleased with this result, considering the slightly degraded state of the experimental Rotman lenses as discussed in Section 3.3.1.

Table 1: The results of the SGA and hBOA for CASE 1.

	SGA						HBOA					
F [GHz]	9.00 (E_1)	9.48 (E_2)	10.0 (E_3)	10.52 (E_4)	11.00 (E_5)	F_1 [$\times10^{-3}$]	9.00 (E_1)	9.48 (E_2)	10.0 (E_3)	10.52 (E_4)	11.00 (E_5)	F_1 [$\times10^{-3}$]
Run 1 error [$\times10^{-3}$]	0.2009	0.0743	0.1307	0.0711	0.1212	0.1196	0.1917	0.0800	0.1407	0.0619	0.1232	0.1194
Run 2 error [$\times10^{-3}$]	0.2310	0.1471	0.2196	0.1225	0.1629	0.1766	0.2034	0.0760	0.1433	0.0588	0.1167	0.1196
Run 3 error [$\times10^{-3}$]	0.2920	0.1117	0.2856	0.0889	0.1793	0.1835	0.1927	0.0785	0.1447	0.0608	0.1225	0.1198
Mean error [$\times10^{-3}$]	0.2413	0.1100	0.1986	0.0942	0.1545	(Mean) 0.1599	0.1958	0.0782	0.1429	0.0605	0.1208	(Mean) 0.1196
STD error [$\times10^{-3}$]	0.0464	0.0364	0.0603	0.0261	0.0300	(STD) 0.0351	0.0067	0.0020	0.0020	0.0016	0.0036	(STD) 0.0002

Table 2: The results of the SGA and hBOA for CASE 2.

	SGA						HBOA					
F [GHz]	9.00 (E_1)	9.48 (E_2)	10.0 (E_3)	10.52 (E_4)	11.00 (E_5)	F_2 [$\times10^{-3}$]	9.00 (E_1)	9.48 (E_2)	10.0 (E_3)	10.52 (E_4)	11.00 (E_5)	F_2 [$\times10^{-3}$]
Run 1 error [$\times10^{-3}$]	3.0140	2.0519	2.2108	1.3882	1.7584	2.0847	2.3269	1.1363	1.5282	1.5482	1.6063	1.6296
Run 2 error [$\times10^{-3}$]	2.8951	1.9392	2.5851	2.0724	1.8038	2.2587	2.1459	1.2727	1.5955	1.5421	1.9166	1.6946
Run 3 error [$\times10^{-3}$]	4.1890	1.6209	2.0234	1.4106	2.8395	2.4167	2.1133	1.3130	1.6334	1.5957	1.8963	1.7103
Mean error [$\times10^{-3}$]	3.3654	1.8707	2.2731	1.6237	2.1339	(Mean) 2.2534	2.1954	1.2413	1.5857	1.5620	1.8064	(Mean) 1.6782
STD error [$\times10^{-3}$]	0.7148	0.2215	0.2860	0.3887	0.6115	(STD) 0.1661	0.1151	0.0915	0.0533	0.0293	0.1736	(STD) 0.0428

Table 3: The results of the SGA and hBOA for CASE 3.												
	SGA						HBOA					
	9.00 (E1)	9.48 (E2)	10.0 (E3)	10.52 (E4)	11.00 (E5)	F_3	9.00 (E1)	9.48 (E2)	10.0 (E3)	10.52 (E4)	11.00 (E5)	F_3
						0.9338						0.0019
						1.0124						0.0019
						1.4588						0.0019
						(Mean) 1.1350						(Mean) 0.0019
						(STD) 0.2832						(STD) 0

3.4 Summary/Conclusions

This section investigated the application of both a simple GA and a competent GA to the optimization of a constrained feed network for a linear array. Specifically, hBOA, one of the most successful competent GAs, was adopted. Several competent GA techniques, including hierarchical decomposition, chunking, and niching, were discussed. Three objective functions were designed to compare the performance of the SGA and hBOA. The first objective function was designed to be GA-easy but did not completely reflect the desired objective of the problem. The remaining two objective functions were GA-difficult but were much more relevant to the problem objectives. For the GA-easy objective function, the performance of the SGA and hBOA were comparable. When the objective functions became more complicated (*i.e.,* GA-difficult), the competent GA technique demonstrated significant improvement over the SGA. In all three cases, the quality of the solutions that hBOA gave is more reliable than that the SGA gave.

To conclude, for simple problems, SGAs are preferred since they are computationally inexpensive and the solution quality is comparable to that of competent GAs. However, for difficult problems, competent GAs should be adopted, because based on our observations, competent GA techniques are able to achieve higher-quality solutions than SGAs.

4.0 Summary/Conclusions

The first half of the paper demonstrated the application of the SGA to several antenna design and optimization problems covering diverse areas of interest –a testament to the broad capacity of the SGA. We saw, however, that a great deal of fiddling with SGA parameters and operators was necessary in order to achieve acceptable, high quality solutions. In particular, we found that the chromosomal encoding scheme has a large impact on the

solution quality, as well as the convergence time. Thus, one of the major drawbacks of using a SGA is that it provides no method for matching genetic operators and coding.

The latter half of the paper discussed a new breed of GA, namely a *competent* GA, for which the coding-operator match is no longer arbitrary. In fact, the competent GA is designed to learn the correlations between different sets of variables encoded along the chromosome. In this manner, the recombination operator can be designed to exploit these correlations and minimize the probability of disrupting good "building blocks" while maximizing the probability of finding a global solution to the problem.

A specific type of competent GA – the hierarchical Bayesian optimization algorithm (hBOA) – was introduced. The SGA and hBOA engaged in head-to-head combat, both attempting to find an acceptable solution to a challenging optimization problem involving a complex, constrained feed network for an antenna array. The results demonstrated that the SGA competes hBOA when the problem was GA-easy. When the problem became more and more difficult, however, hBOA constantly outperformed the SGA in both computational and electromagnetic aspects. This case study demonstrates the utility of using more advanced GA techniques to obtain acceptable solution quality as problem difficulty increases.

Based on the success of this project, future collaborative efforts between the AFRL Antenna Technology Branch and IlliGAL look promising. In particular, we would like to take a closer look at the electrically small antenna research described in Section 2.3.7. Here, we saw that three different chromosomal encoding schemes led to very different solutions. In theory, a competent GA should be able to find a global, or near global, solution regardless of the encoding scheme. Thus, this project will be a true test of the competent GA's capabilities.

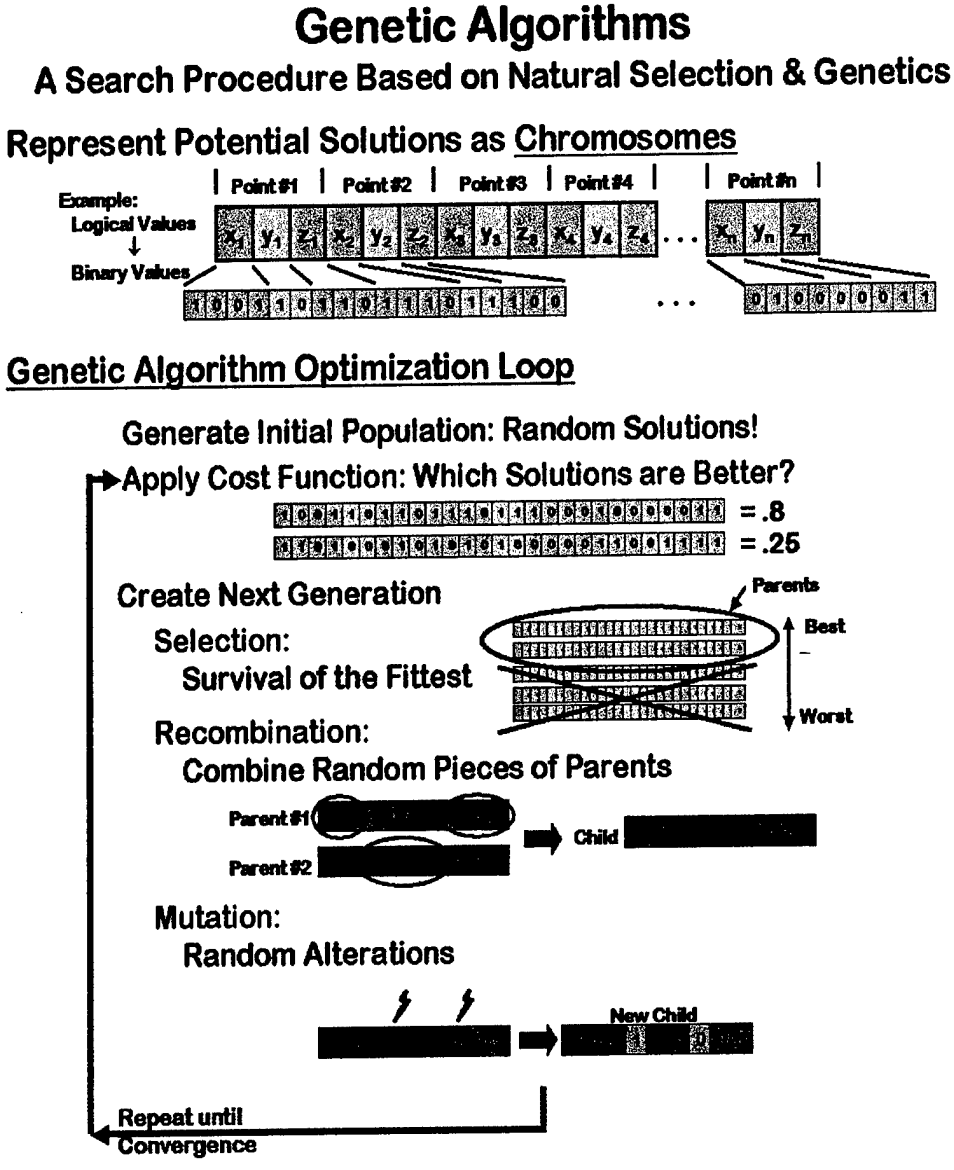


Figure 1: SGA parameters and operators

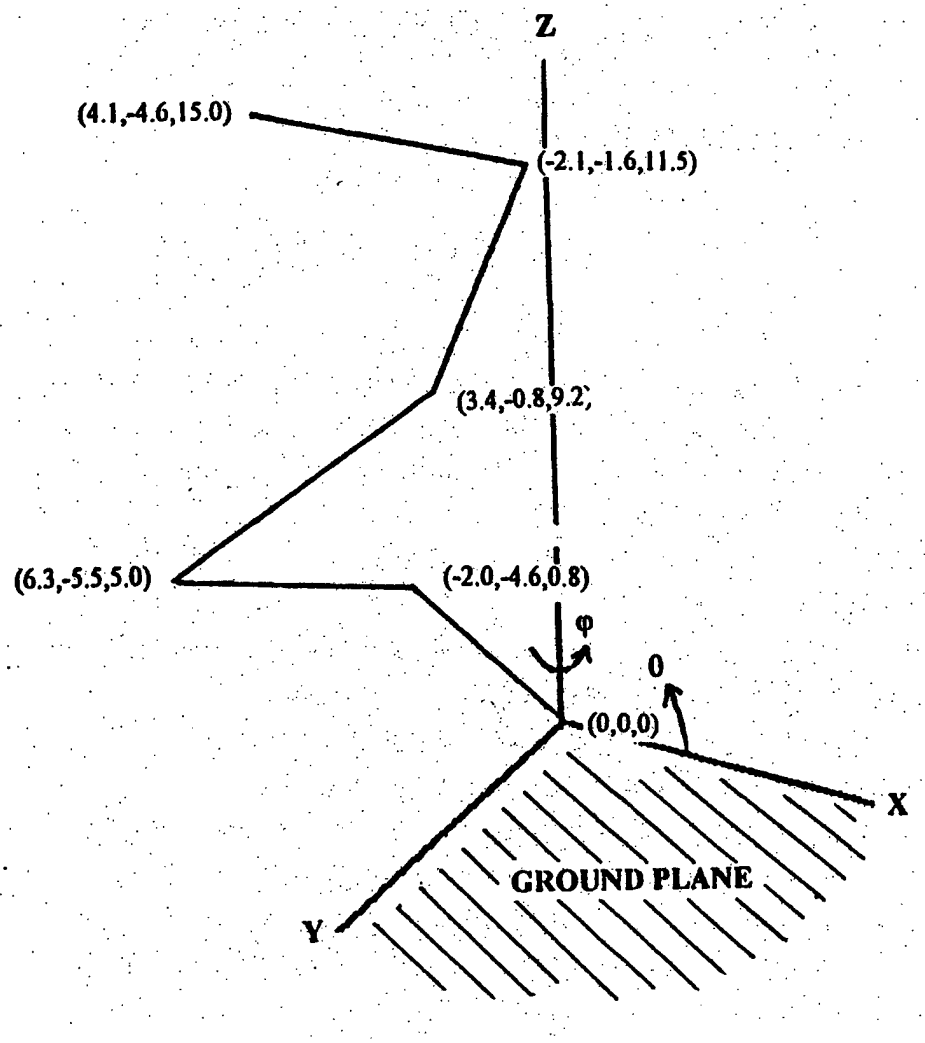


Figure 2: Sketch of GPS/Iridium Antenna

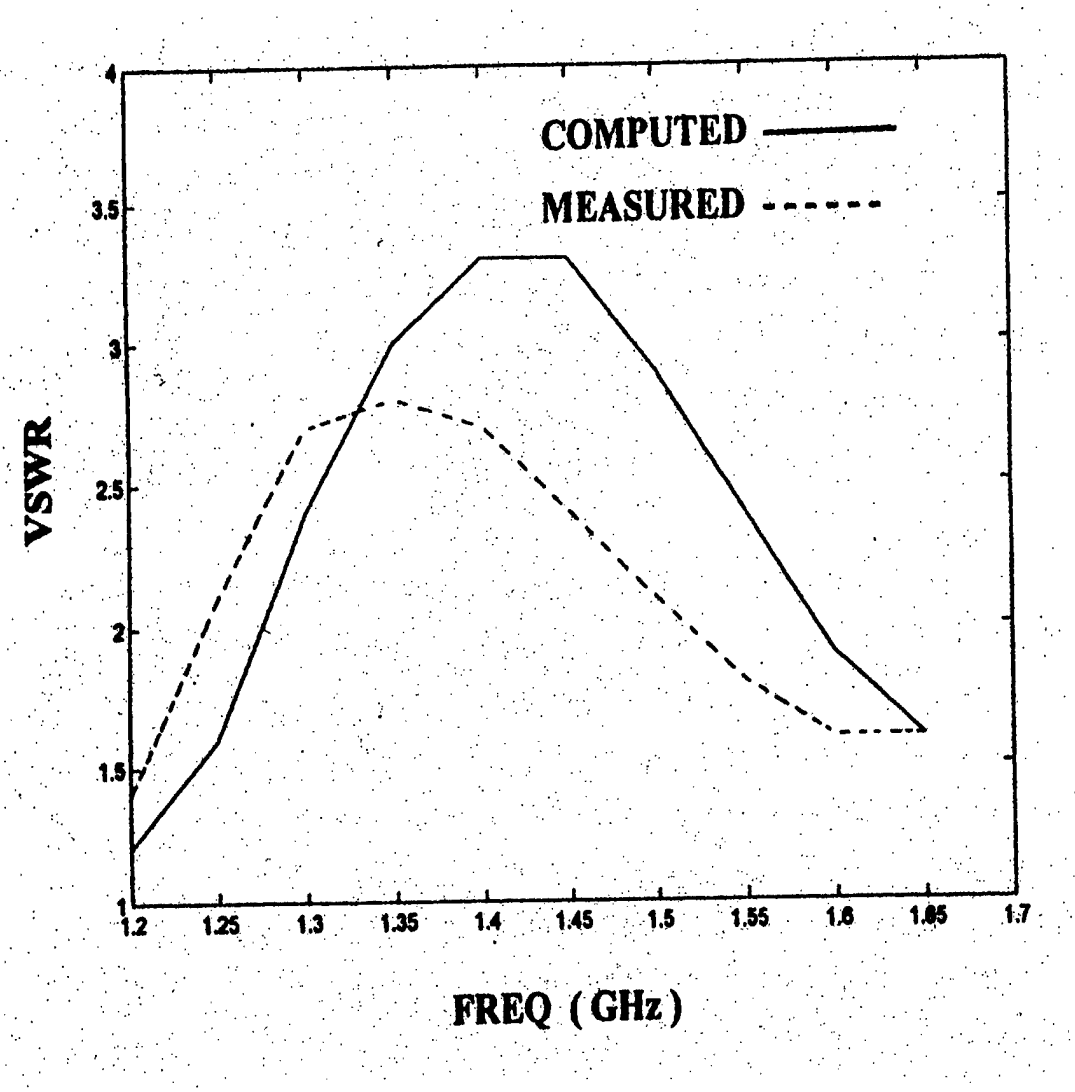


Figure 3: (Fig.2) VSWR of GPS/Iridium Antenna

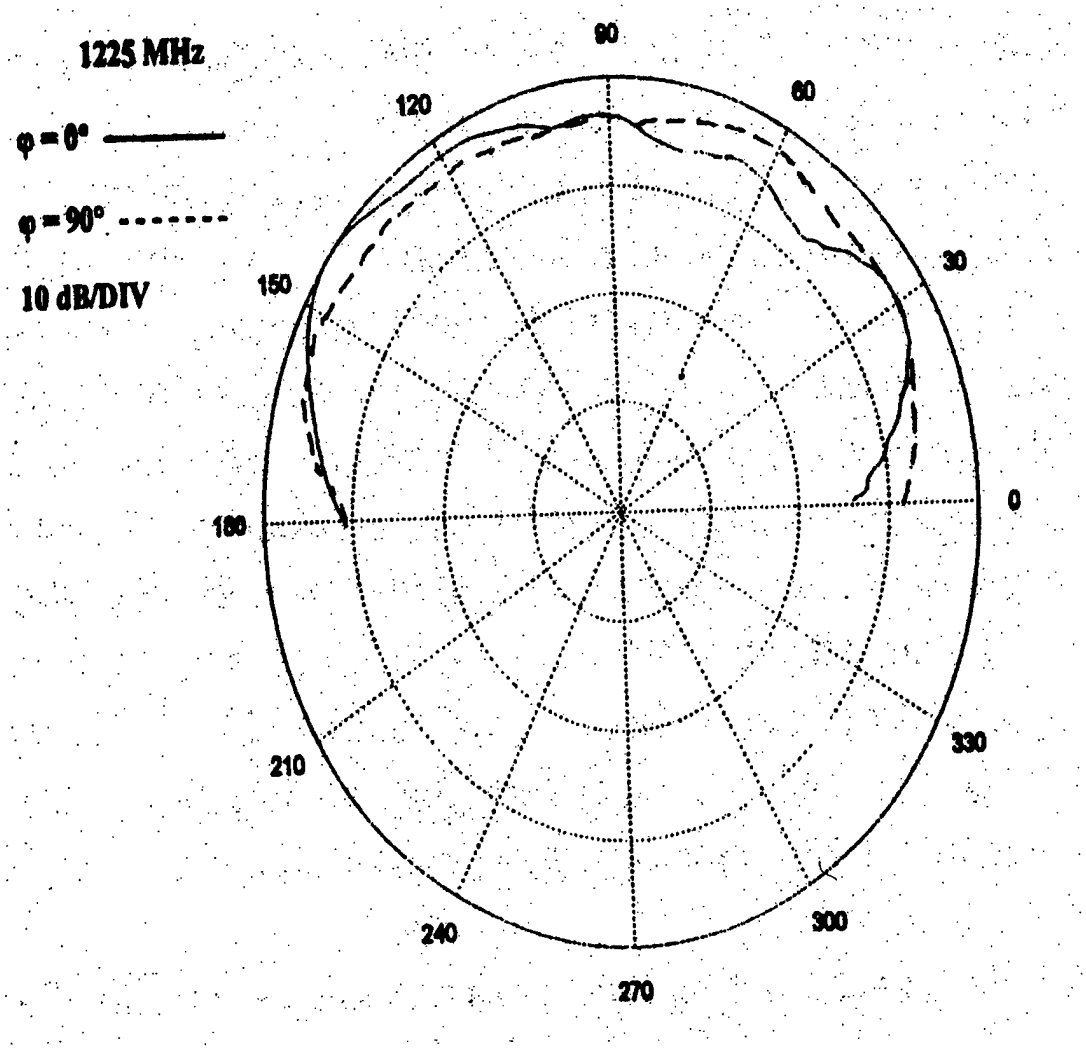


Figure 4: Radiation Pattern of GPS/Iridium Antenna

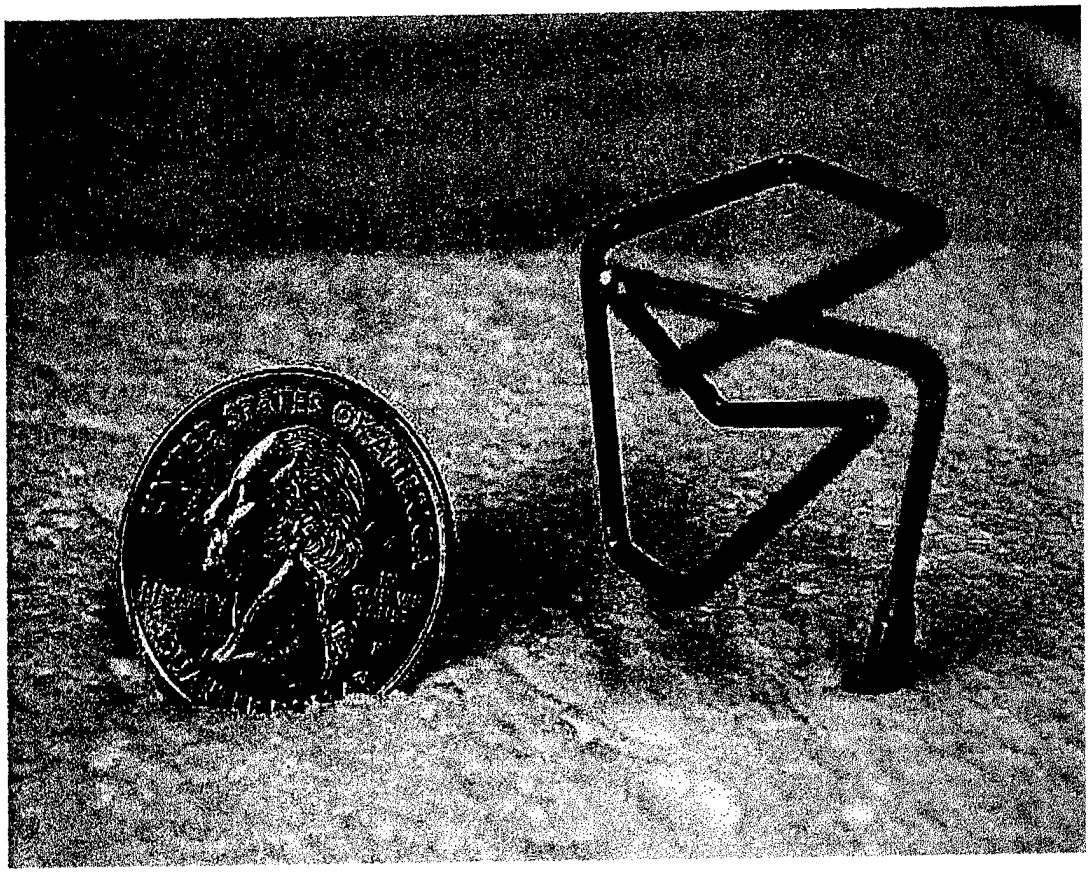


Figure 5: Electrically Small Genetic Antenna

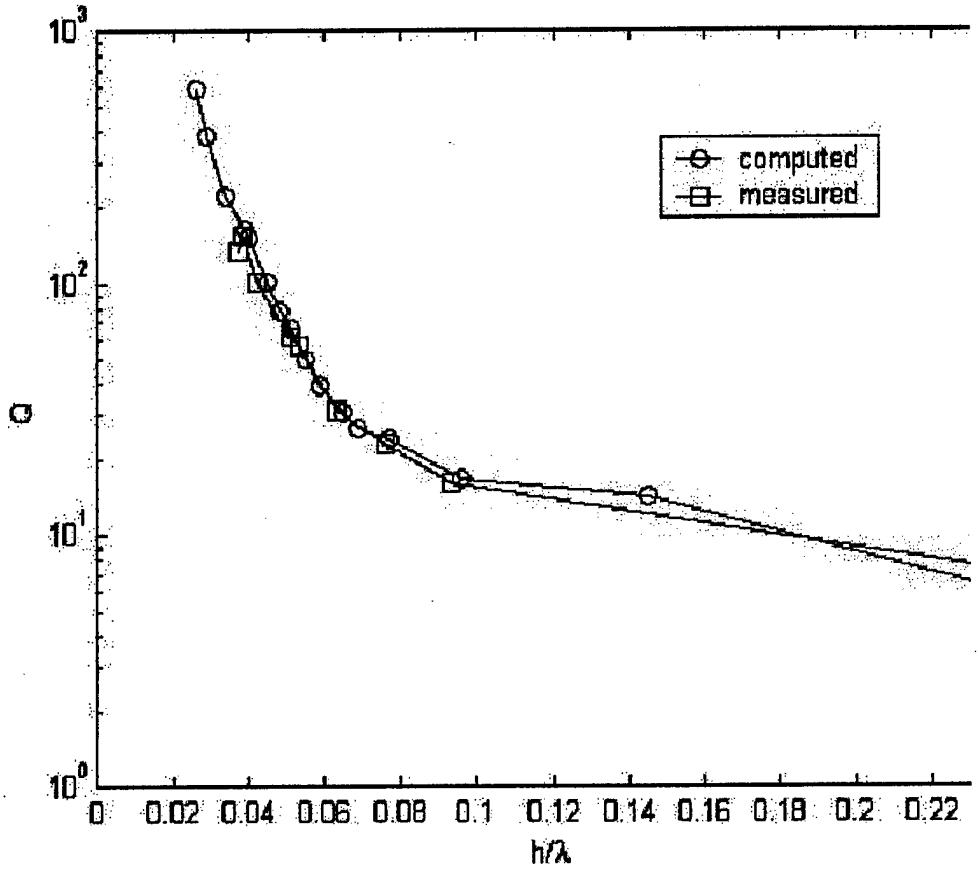


Figure 6: *Q* of Electrically Small Genetic Antenna

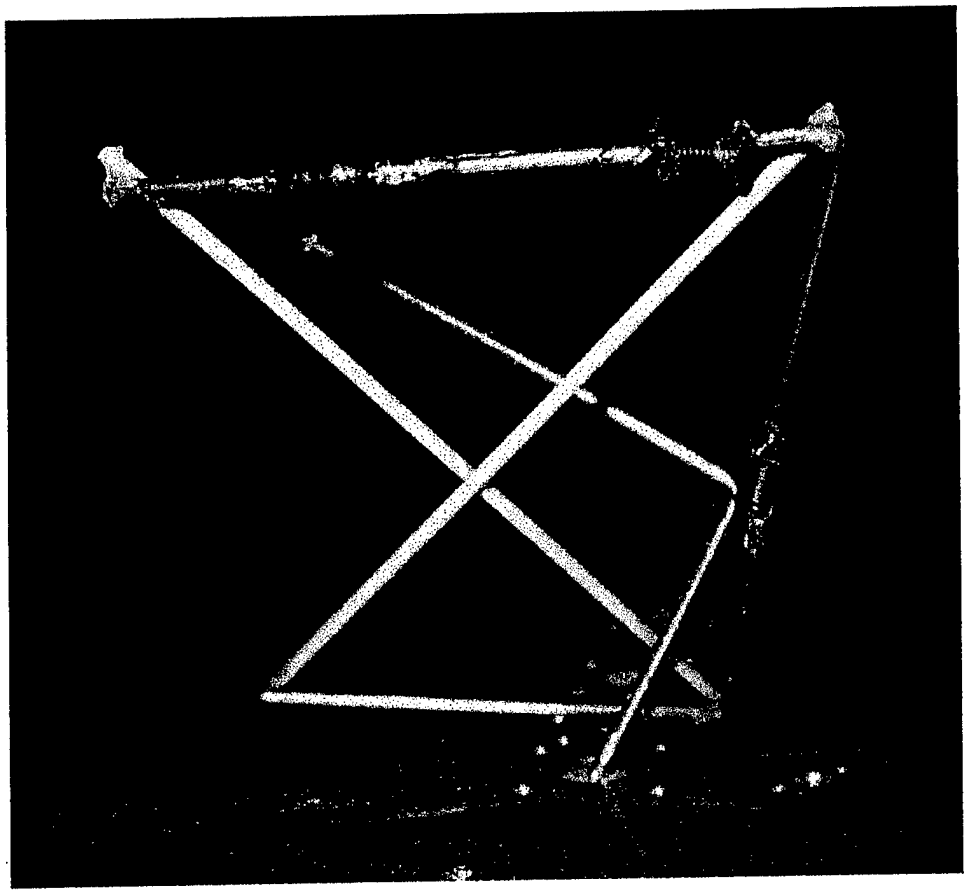


Figure 7: Ultra-Wideband, Impedance-Loaded Genetic Antenna

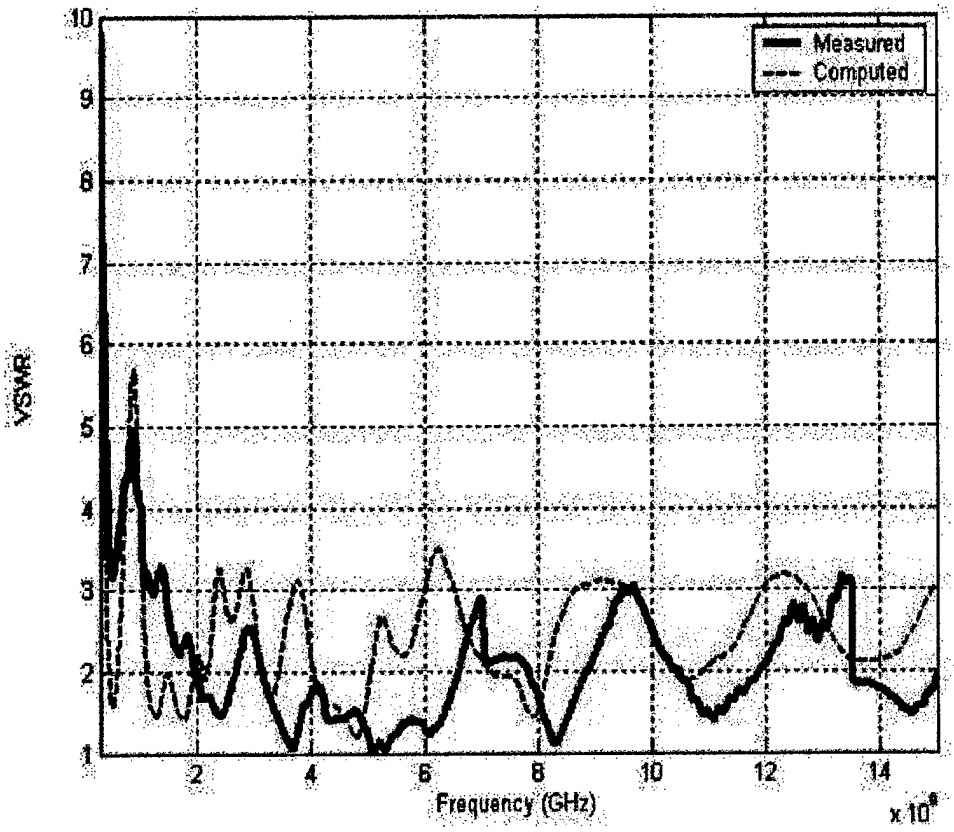


Figure 8: VSWR of Ultra Wideband Antenna

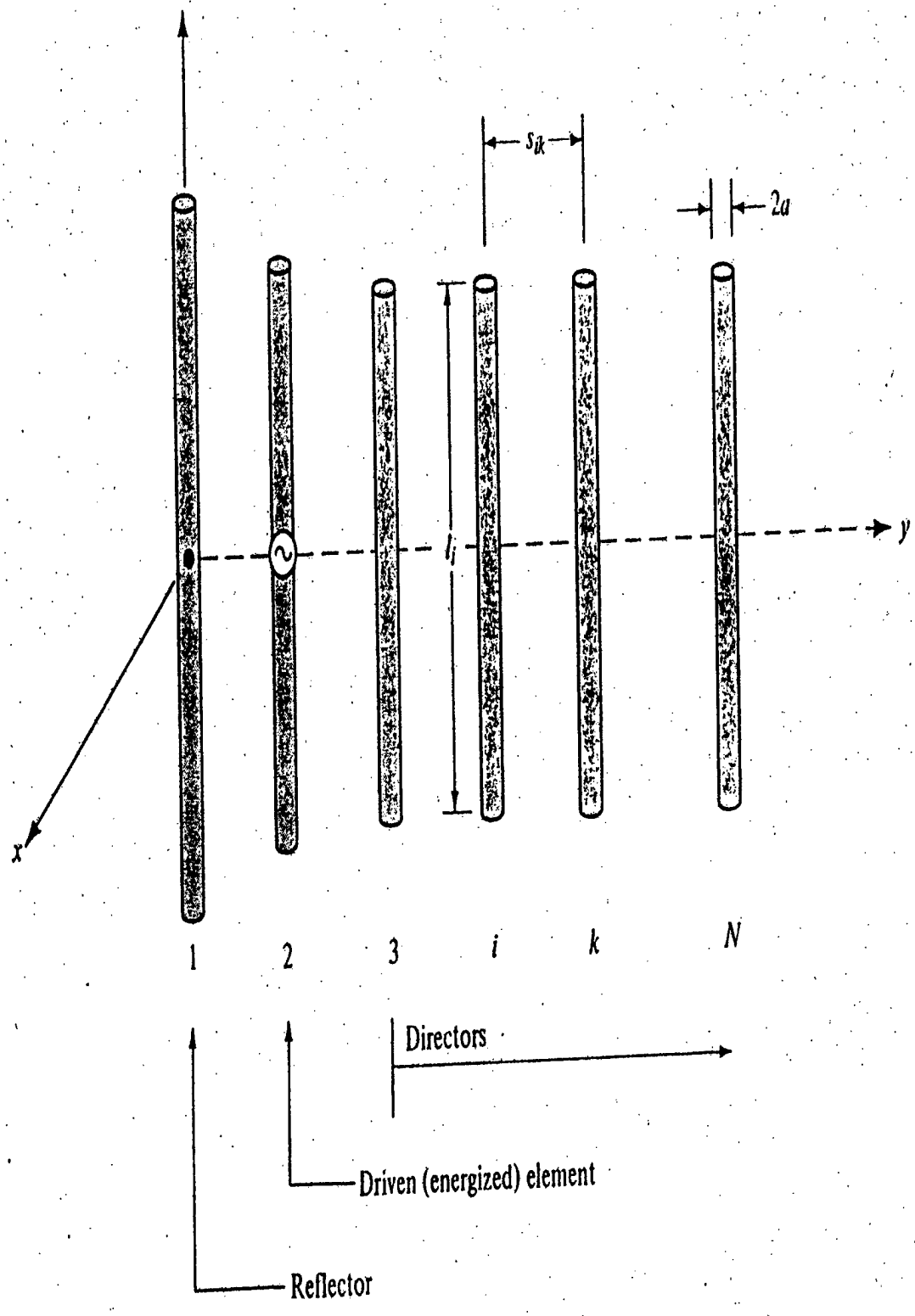


Figure 9: Conventional Yagi Antenna

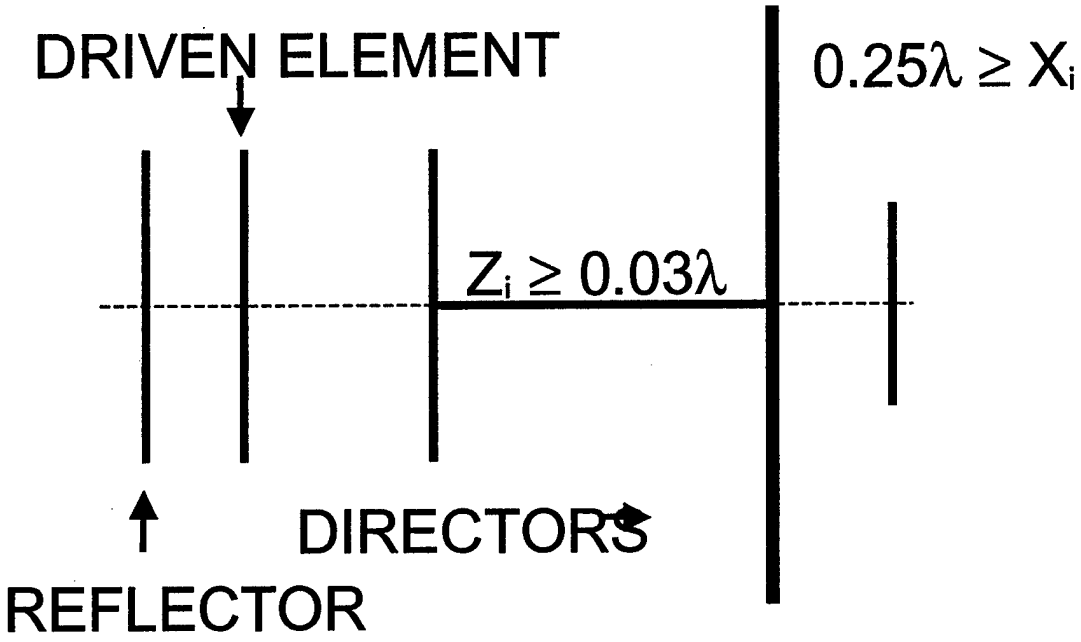


Figure 10: Genetic Yagi

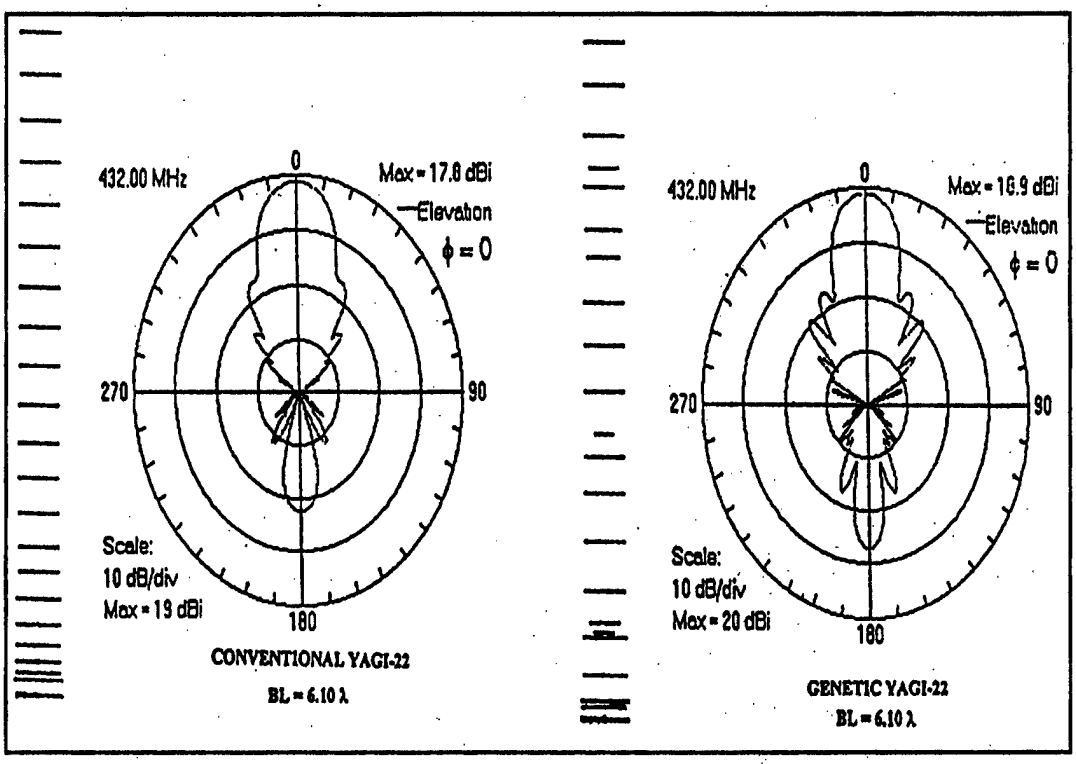


Figure 11: Gain of Yagi Antennas

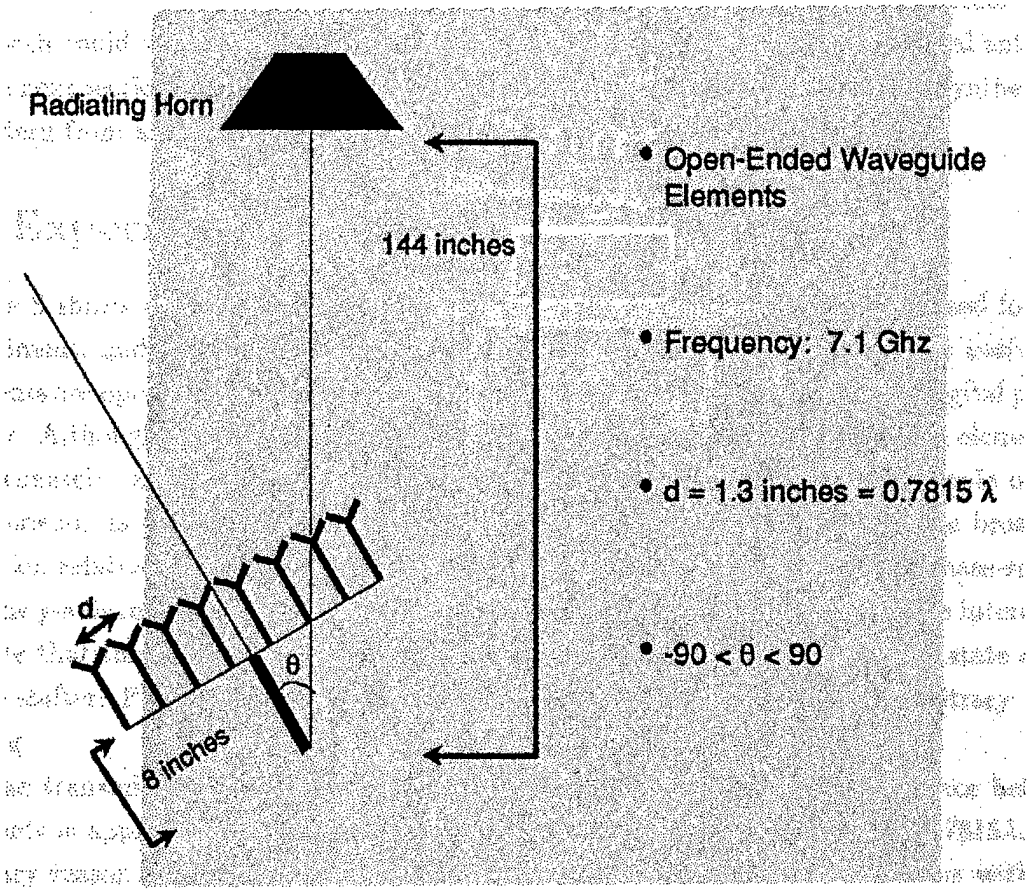


Figure 12: Experimental Configuration

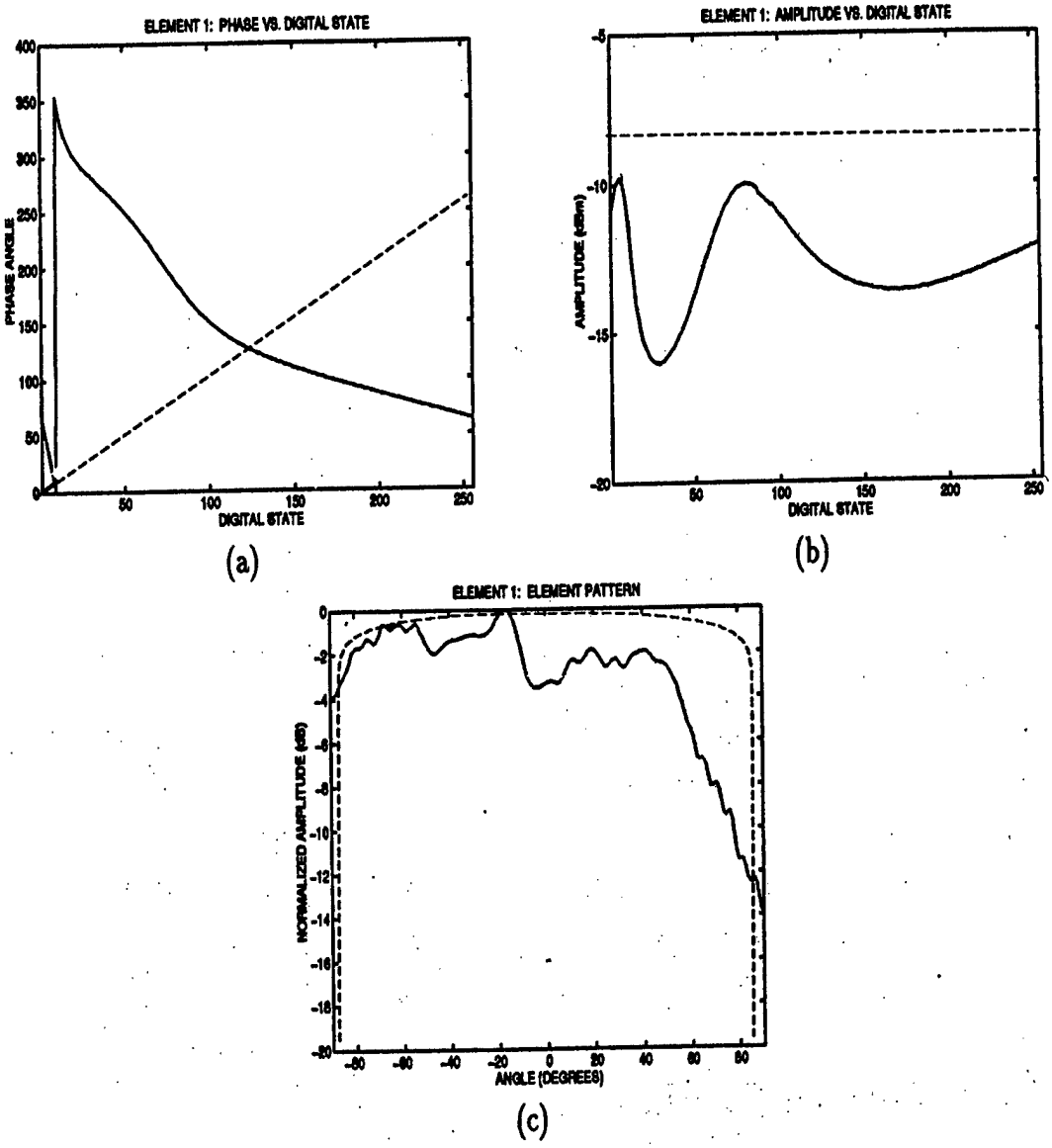


Figure 13: Single Element Transfer-Function Characteristics

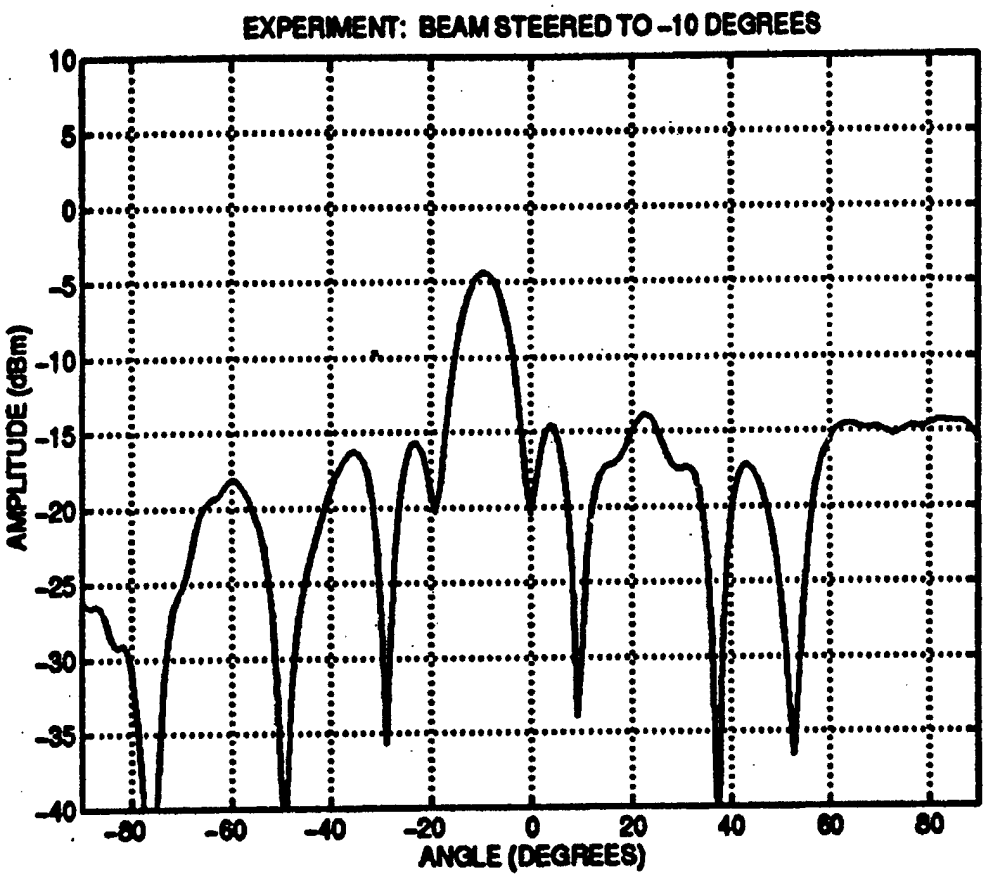
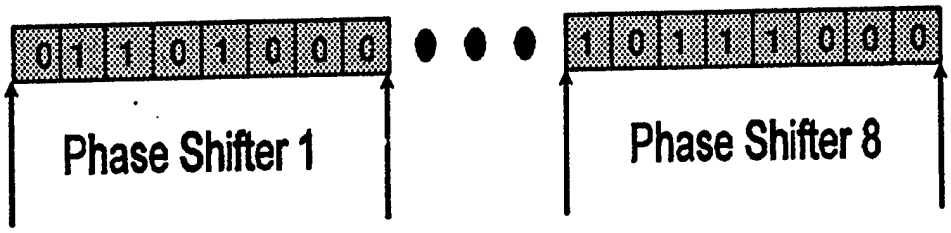


Figure 14: Experimental Radiation Pattern

Problem Representation



- Eight 8-bit Phase Shifters

Figure 15: Chromosomal Representation

```
GRAYTABLE5 = [[0,0,0,0,0]
              [0,0,0,0,1]
              [0,0,0,1,1]
              [0,0,0,1,0]
              [0,0,1,1,0]
              [0,0,1,1,1]
              [0,0,1,0,1]
              [0,0,1,0,0]
              [0,1,1,0,0]
              [0,1,1,0,1]
              [0,1,1,1,1]
              [0,1,1,1,0]
              [0,1,0,1,0]
              [0,1,0,1,1]
              [0,1,0,0,1]
              [0,1,0,0,0]
              [1,1,0,0,0]
              [1,1,0,0,1]
              [1,1,0,1,1]
              [1,1,0,1,0]
              [1,1,1,1,0]
              [1,1,1,1,1]
              [1,1,1,0,1]
              [1,1,1,0,0]
              [1,0,1,0,0]
              [1,0,1,0,1]
              [1,0,1,1,1]
              [1,0,1,1,0]
              [1,0,0,1,0]
              [1,0,0,1,1]
              [1,0,0,0,1]
              [1,0,0,0,0]];
```

Figure 16: Example Gray-Coding Scheme

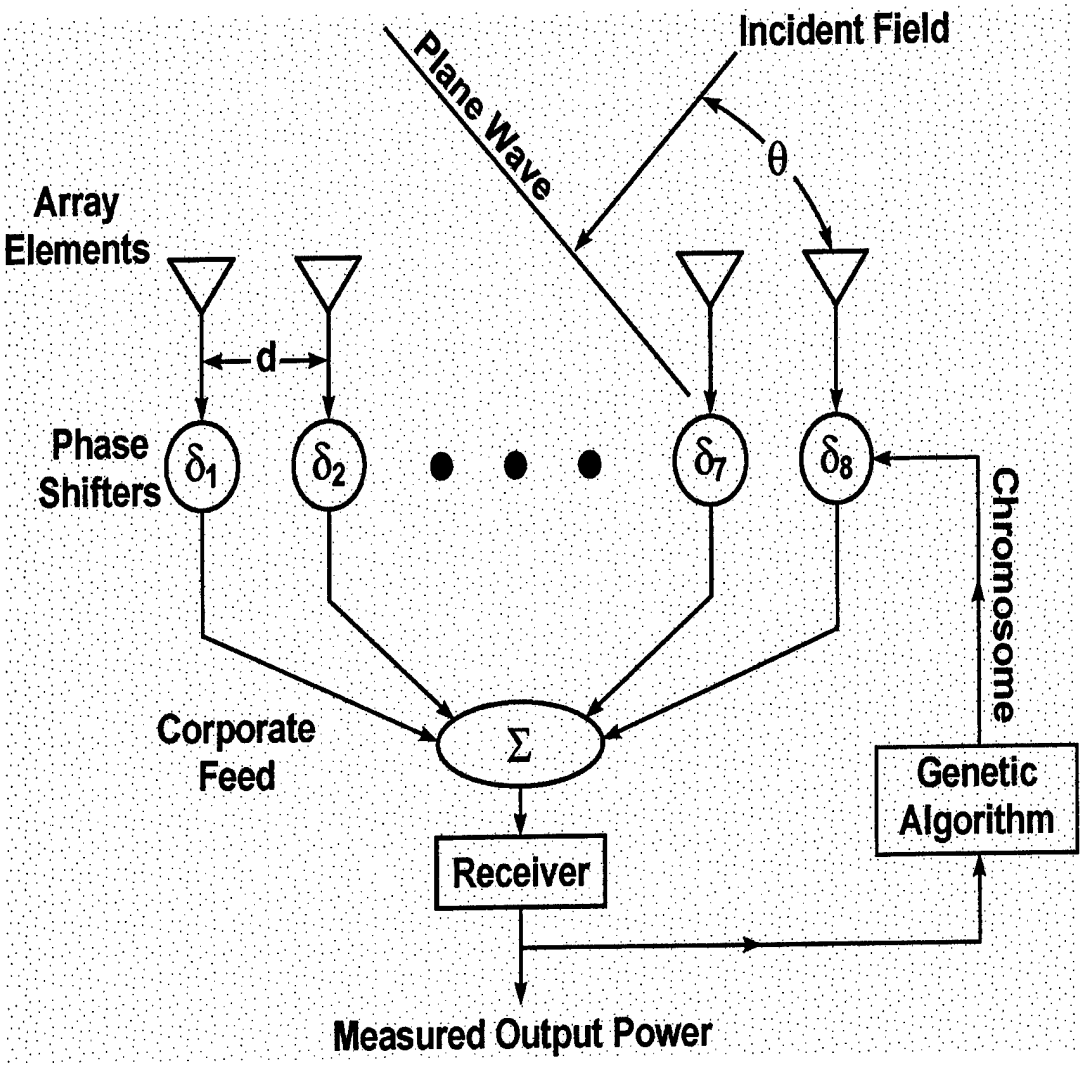


Figure 17: The adaptive array model with the SGA [12].

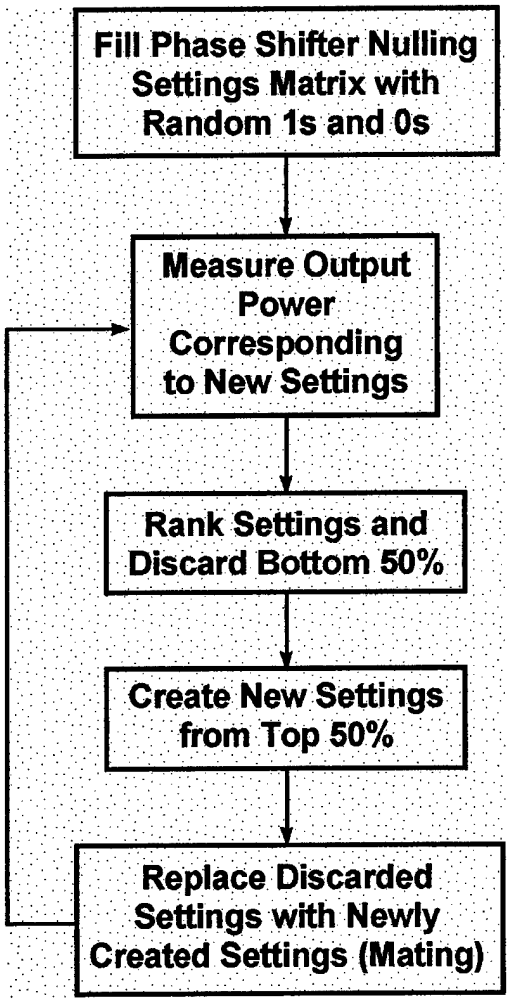


Figure 18: Flow chart of the adaptive SGA [12].

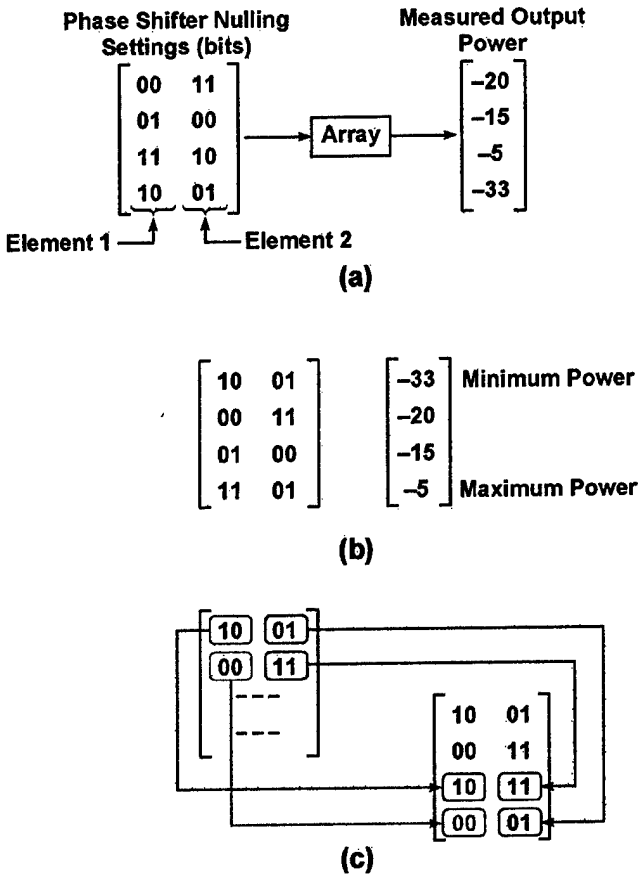


Figure 19: A simple, bit-level example of the genetic algorithm for a two-element array [12]. (a) Initial random matrix with corresponding output powers; (b) Chromosomes are ranked from lowest to highest output power (Note: power is measured in dBm, *i.e.*, dB relative to 1 milliwatt, therefore more negative numbers represent smaller power measurements); (c) Illustration of the mating process used to produce M/2 children.

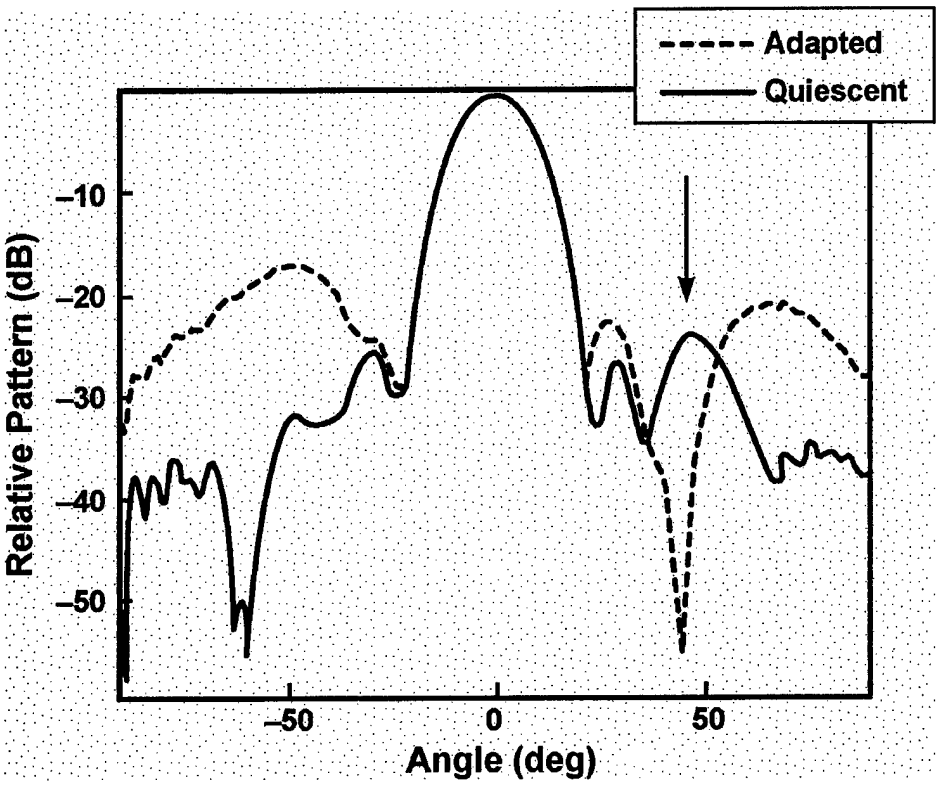


Figure 20: The antenna pattern resulting from weights (phase shifter settings) produced by the SGA with an interfering source at 45° [12].

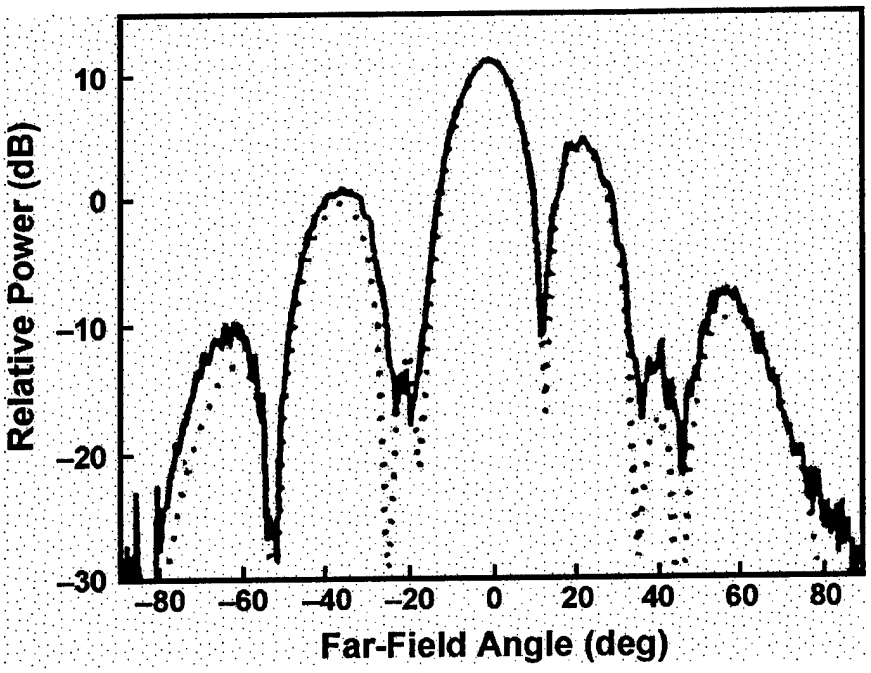


Figure 21: Measured (solid) and theoretical (dashed) array beam patterns for the 20° null. Theoretical patterns were calculated from SGA-produced phase-shifter settings and their corresponding phase shifts, which were found in look up tables (*i.e.*, calibration tables for the individual phase shifters [13]). The phase shifts were used in a far-field antenna pattern simulation.

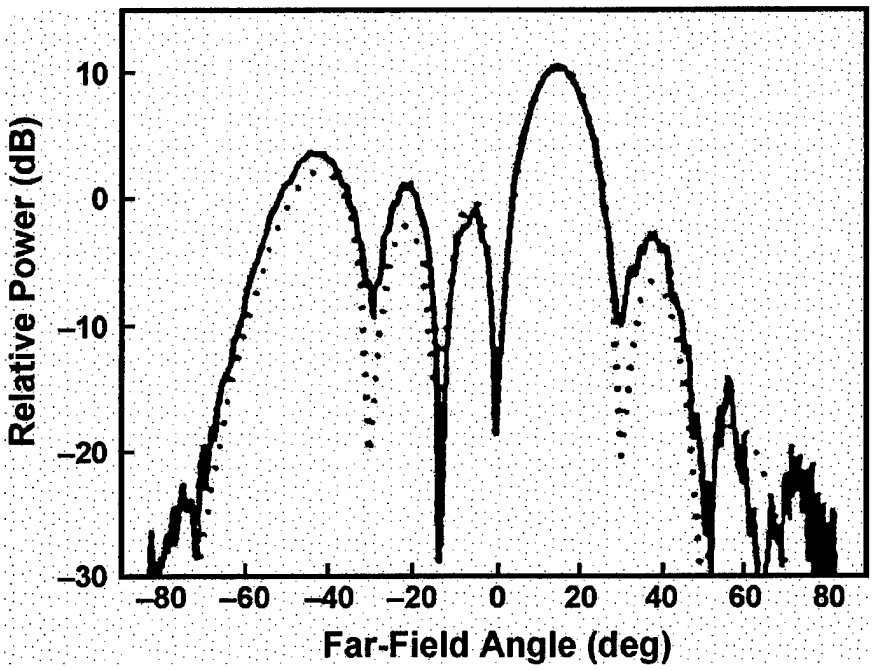


Figure 22: Measured (solid) and theoretical (dashed) array beam patterns for the 15° scanned beam [13].

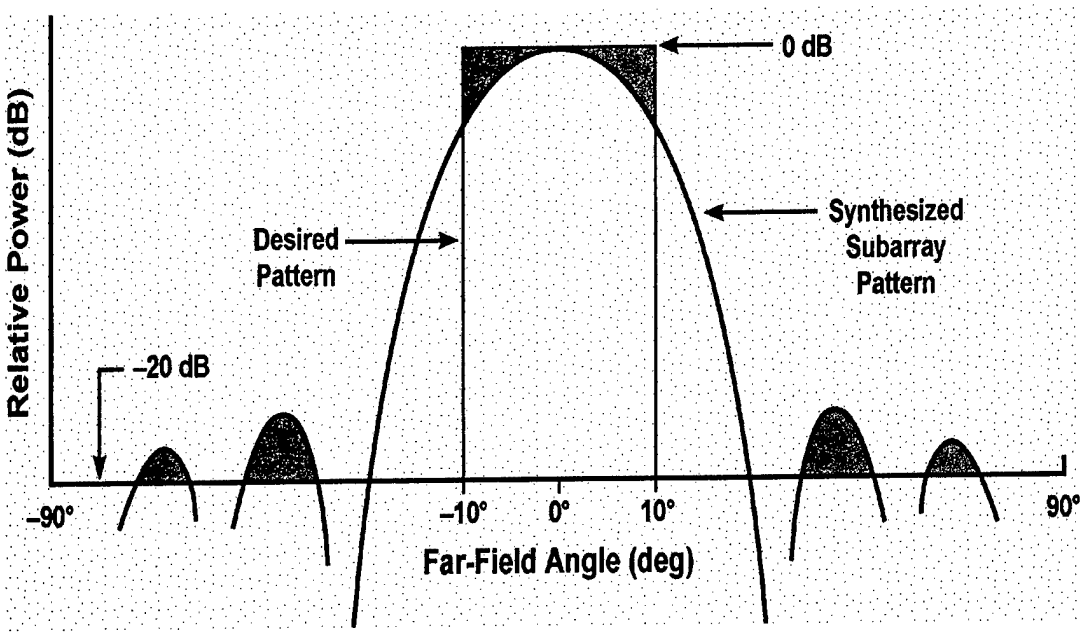


Figure 23: Typical subarray pattern illustrating the shaded areas used to calculate the cost function for the genetic algorithm optimization.

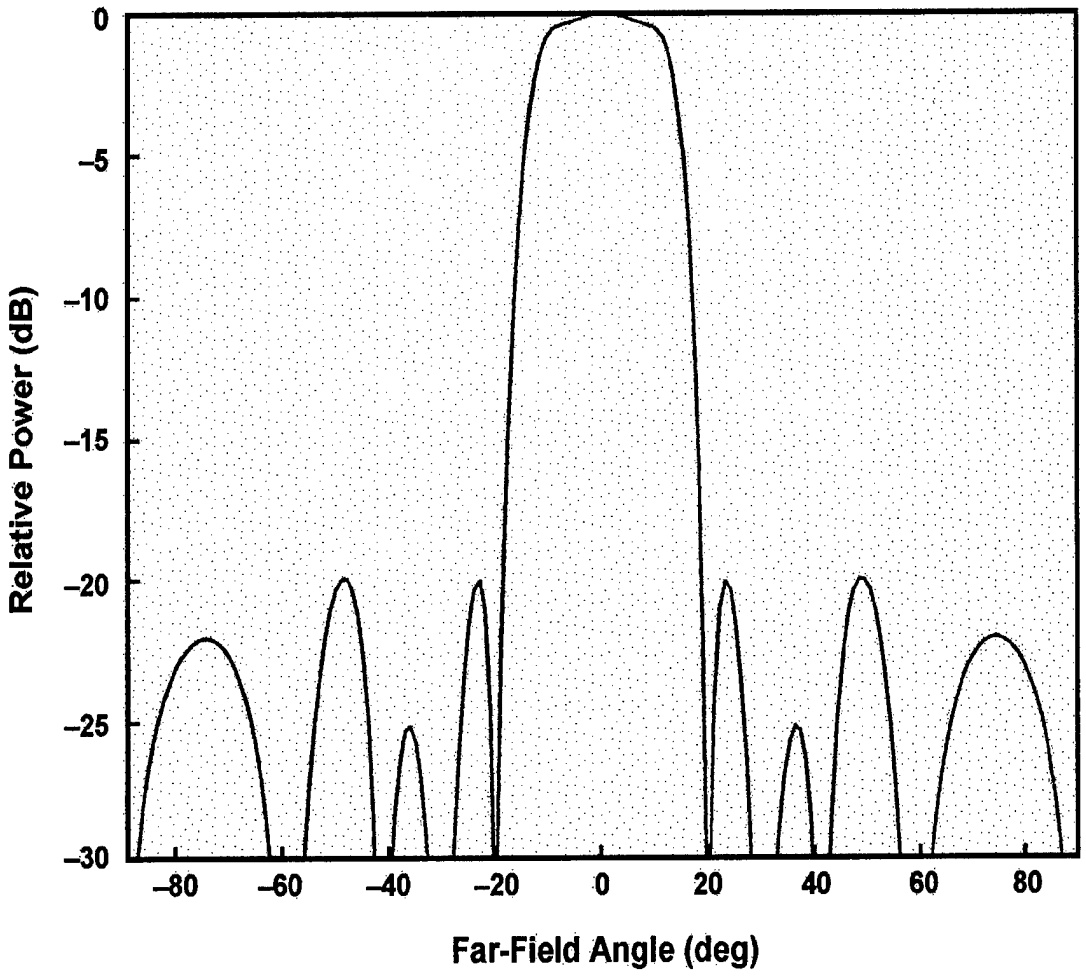
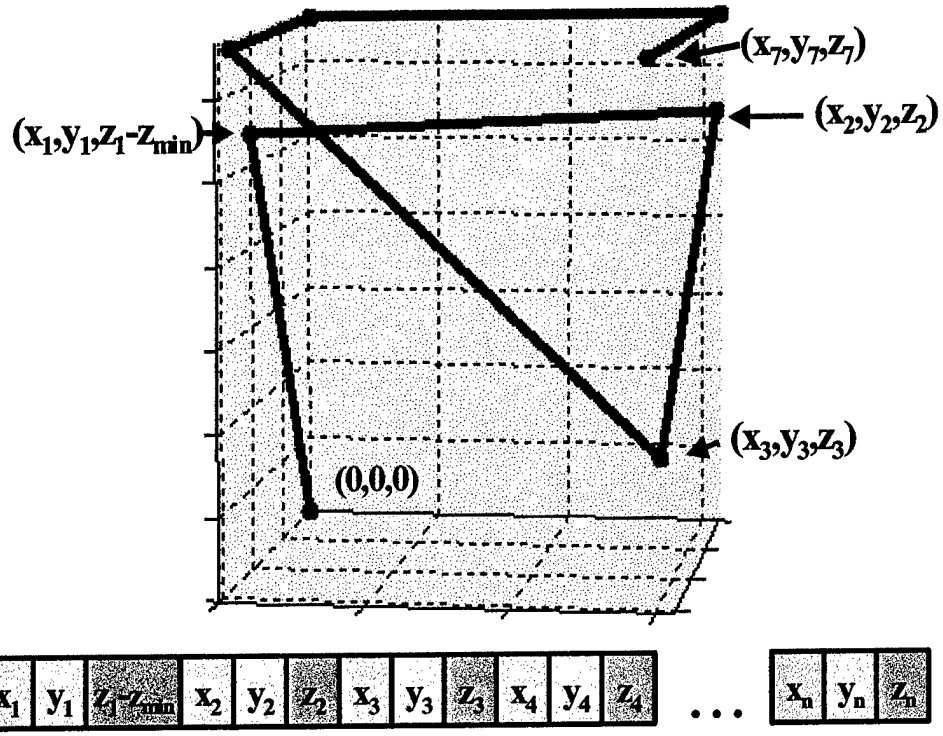
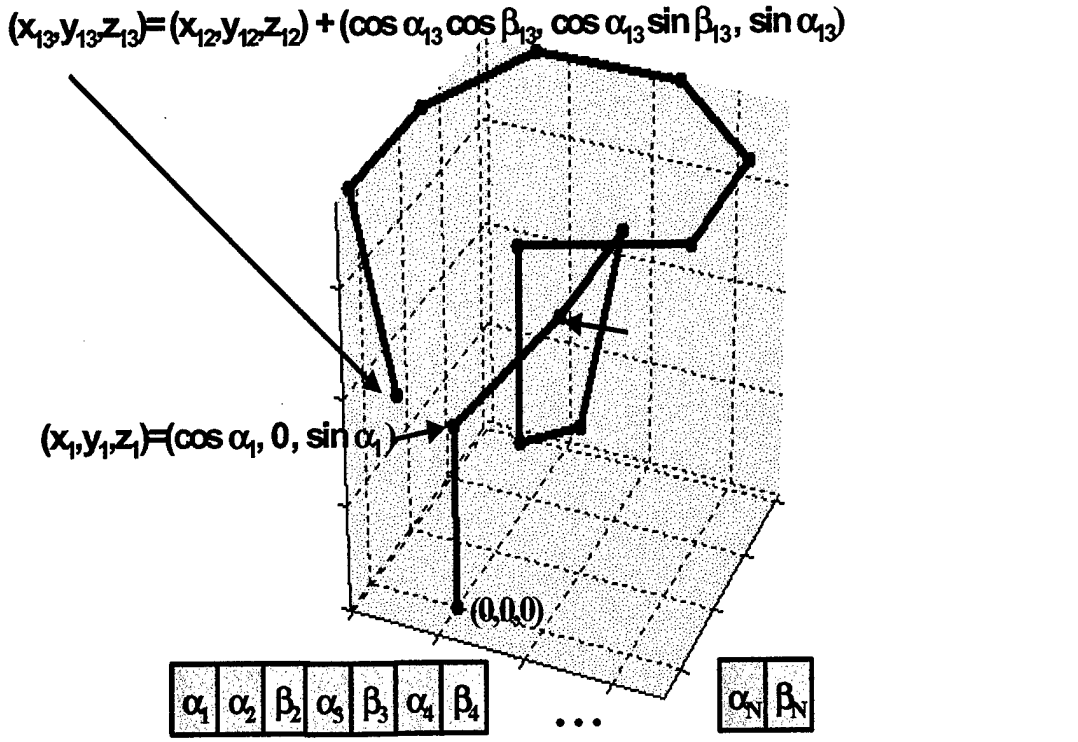


Figure 24: Skobelev (N = 2 case) subarray pattern simulated using quadrature hybrid coupler coupling coefficients obtained from the SGA.



Chromosome: Genes represent wire endpoints.
Gene: a floating point number (0-1 inclusive)
x_n, y_n, z_n, are pseudo-coordinates
Pseudo-coordinates x desired cube size = NEC wire coordinates

Figure 25: The real-valued, Cartesian coordinate-based genetic antenna chromosome and a typical resulting antenna. Note the straight-line configuration.



Chromosome: 2N - 1 genes, Az/El angles of N equal-length wire pieces (connected in series) comprising antenna of fixed length, L (in λ)

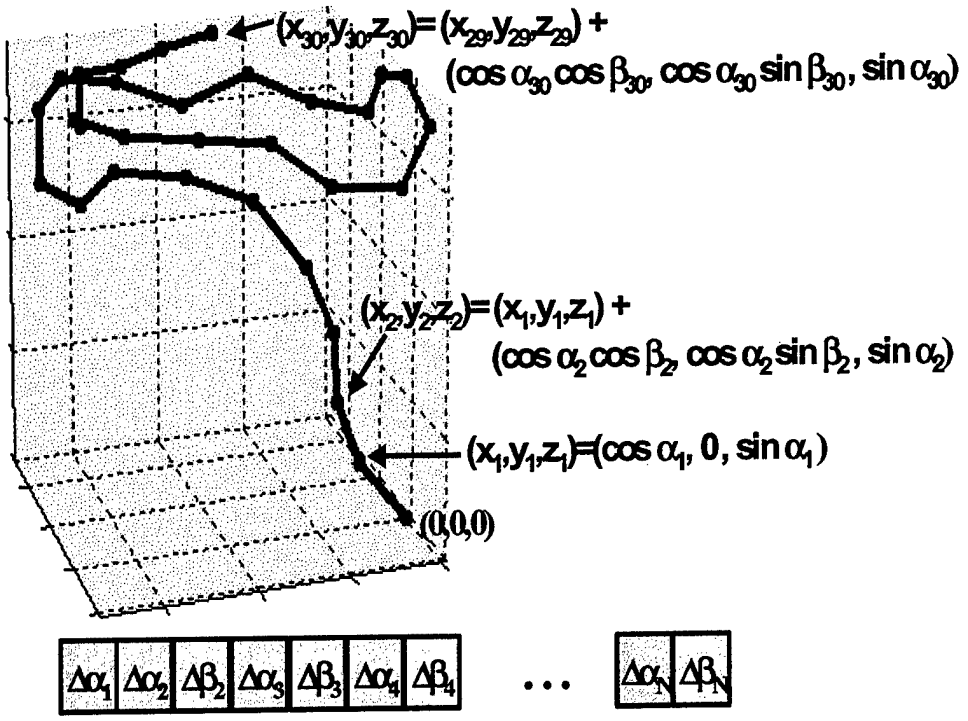
Gene: a n-bit, binary cyclic gray-coded angle (0-2π)

α_n are elevation angles, β_n are azimuth angles.

Pseudo-coord. offsets $[\Delta x_n, \Delta y_n, \Delta z_n] = ([\cos\alpha_n \cos\beta_n, \cos\alpha_n \sin\beta_n, \sin\alpha_n])$

NEC wire coordinates = $[x_{n-1}, y_{n-1}, z_{n-1}] + [\Delta x_n, \Delta y_n, \Delta z_n]\,\lambda L/N$

Figure 26: The absolute angle genetic antenna chromosome and a typical resulting antenna. Some attempt to represent a top-loaded helix is evident, but suppressed by the non-optimal chromosome encoding.



Chromosome: 2N - 1 genes, representing ΔAz/El angles of N equal-length wire pieces (connected in series) comprising antenna of fixed length, L (in λ)

Gene: a n-bit, binary cyclic gray-coded angle (within +/- θ)

$\Delta\alpha_n$ are changes in elevation angle, $\Delta\beta_n$ are changes azimuth angles.

$\alpha_n=\alpha_{n-1}+\Delta\alpha_n \qquad \beta_n=\beta_{n-1}+\Delta\beta_n$

Pseudo-coord. offsets $[\Delta x_n, \Delta y_n, \Delta z_n] = ([\cos\alpha_n \cos\beta_n, \cos\alpha_n \sin\beta_n, \sin\alpha_n])$

NEC wire coordinates $= [x_{n-1}, y_{n-1}, z_{n-1}] + [\Delta x_n, \Delta y_n, \Delta z_n]\,\lambda L/N$

Figure 27: The relative angle genetic antenna chromosome and a typical resulting antenna. Note the approximation to a curvy shape resembling a top-loaded helix.

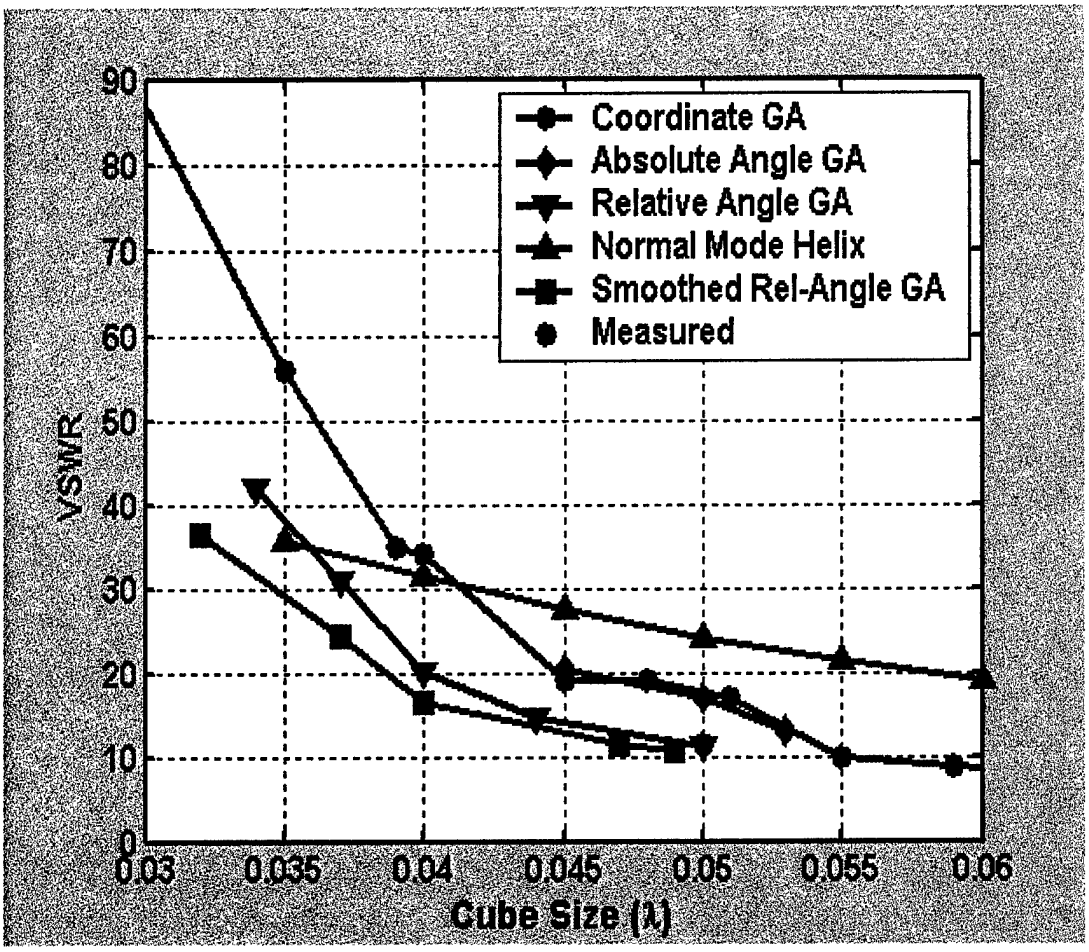


Figure 28: Comparison of VSWR performance for the Cartesian coordinate, absolute-angle, and relative-angle GAs and a normal mode helix. Measured data and the results of hand-smoothing the relative-angle antennas are also included.

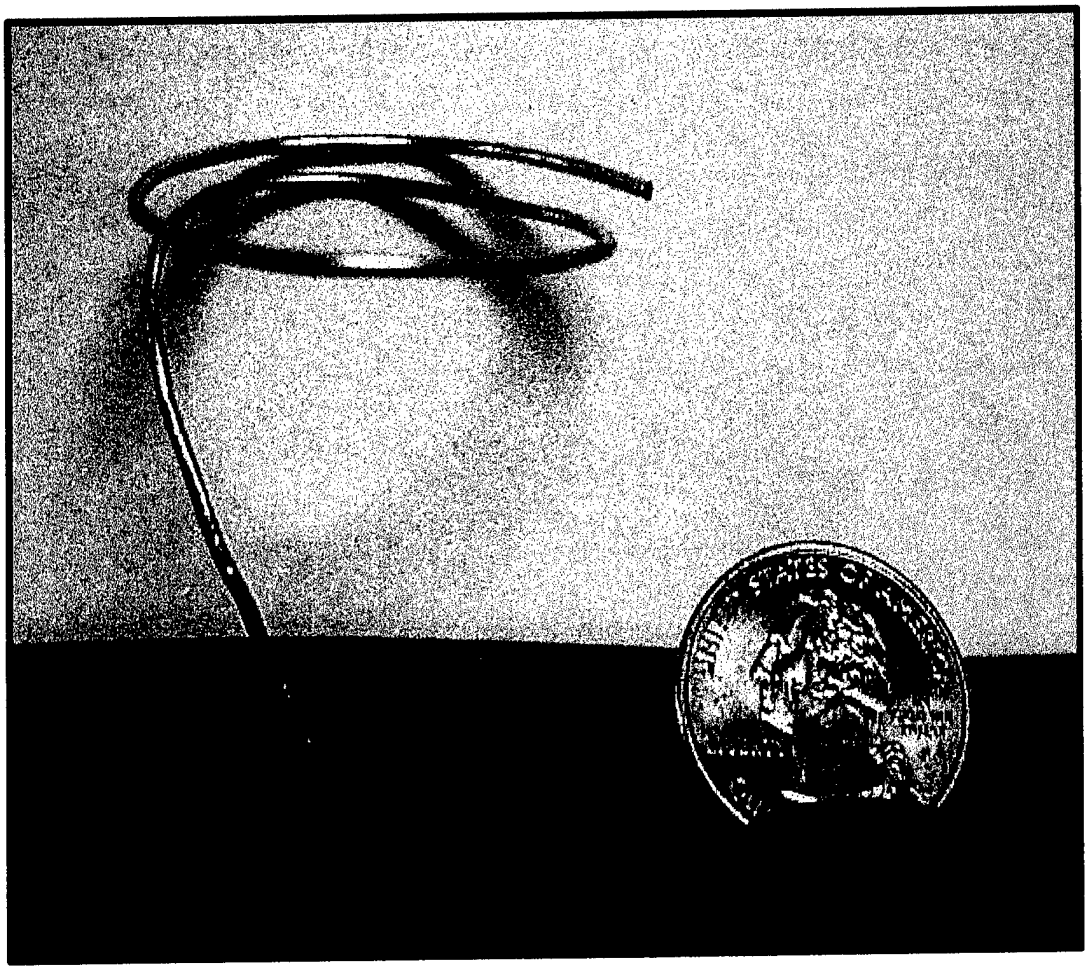


Figure 29: The 0.05λ relative-angle, electrically-small, bent-wire antenna, whose measured VSWR is depicted in Figure 28. This antenna outperformed both the normal mode helix and other previously constructed genetic antennas for this cube-size.

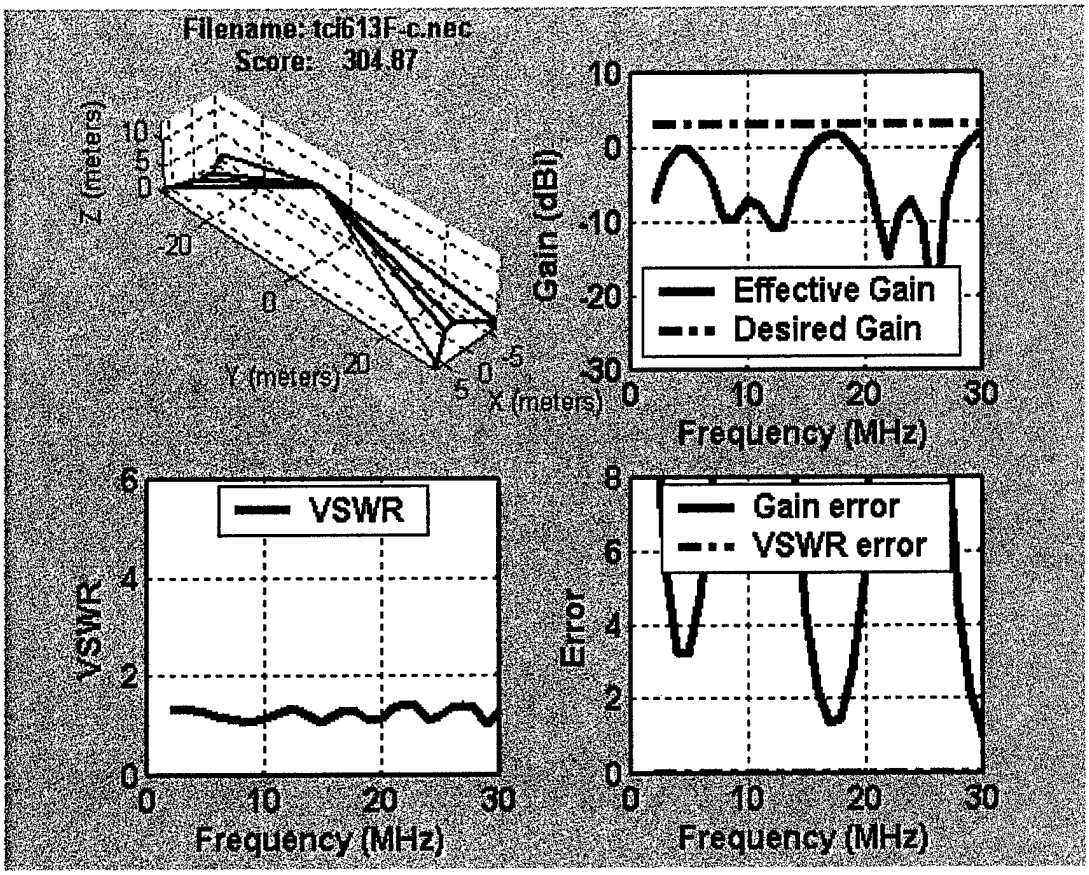


Figure 30: The original DISS Tx antenna – an off-the-shelf TCI 613 communications antenna.

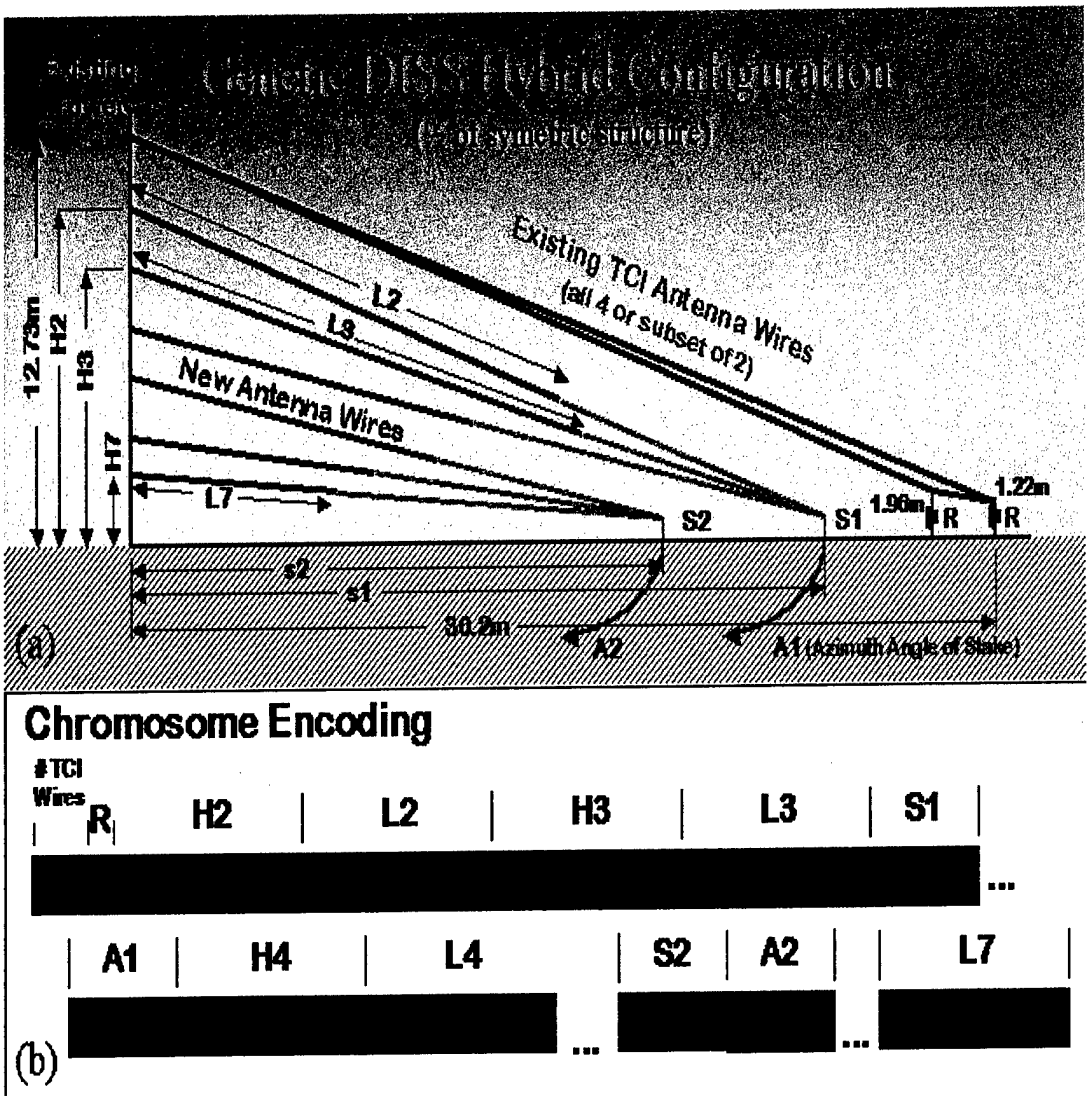


Figure 31: A schematic of the genetic DISS hybrid antenna design and the chromosome used to encode it into the SGA.

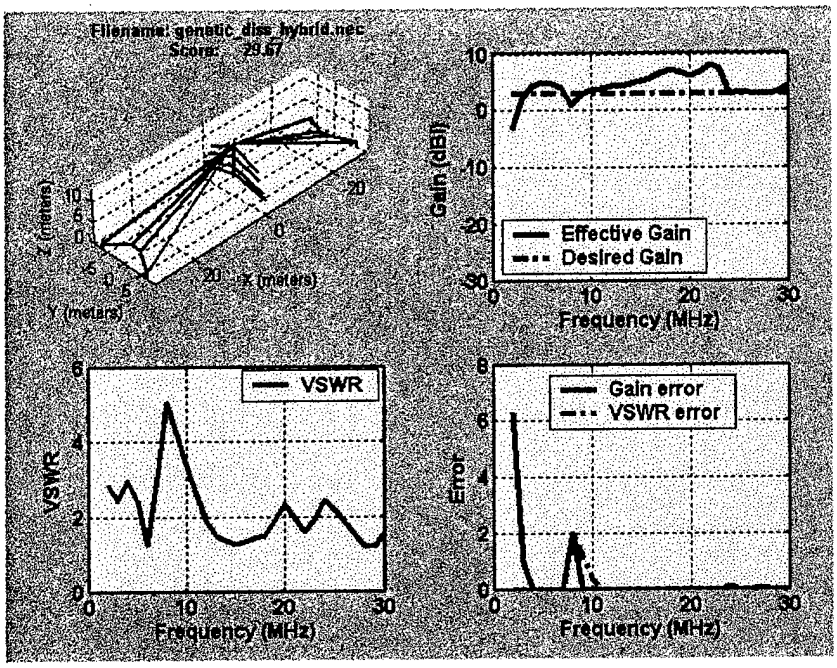


Figure 32: The genetic DISS hybrid Tx antenna. Note the significantly improved gain over that shown in Figure 30. The VSWR, although not as low as the TCI across the entire bandwidth, is acceptable.

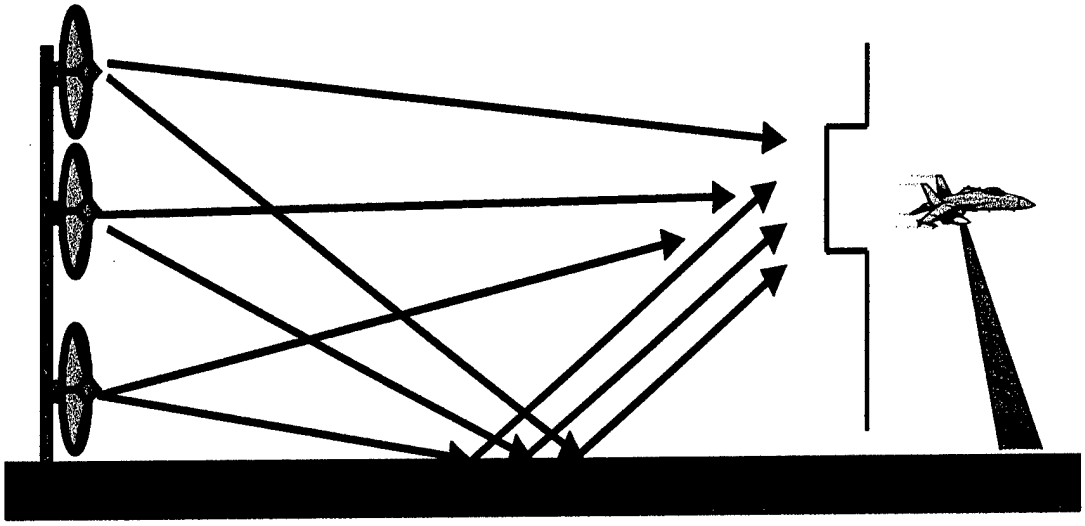


Figure 33: A schematic of the 3-dish reflector array used to illuminate the target on the pylon for RCS (radar cross section) measurements on the RATSCAT ground-bounce range.

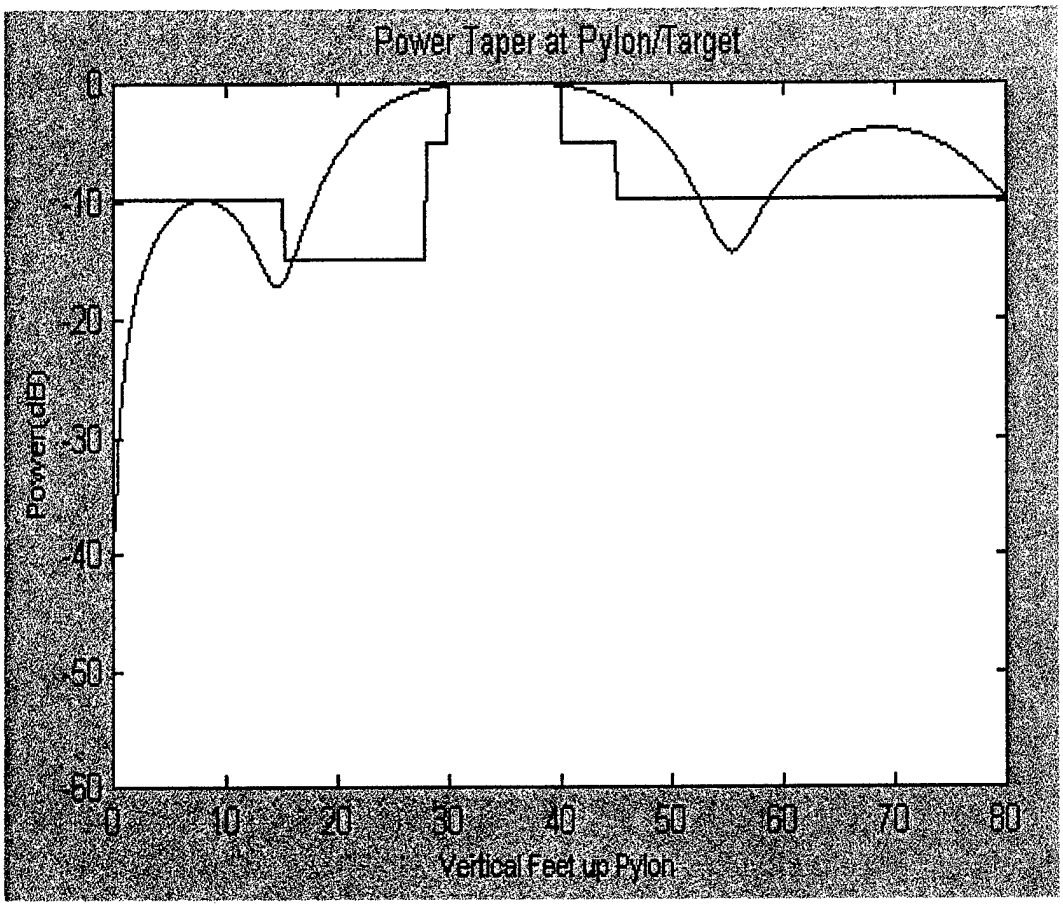


Figure 34: The power taper on the pylon resulting from the SGA configuration of the reflector dish array. Note the flat power pattern achieved across the target area (30-40 ft), which is essential to accurate RCS measurements.

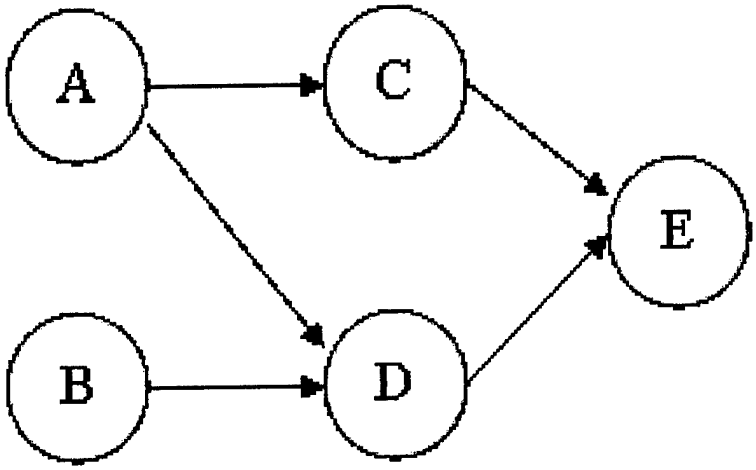


Figure 35: An example of a Bayesian network structure.

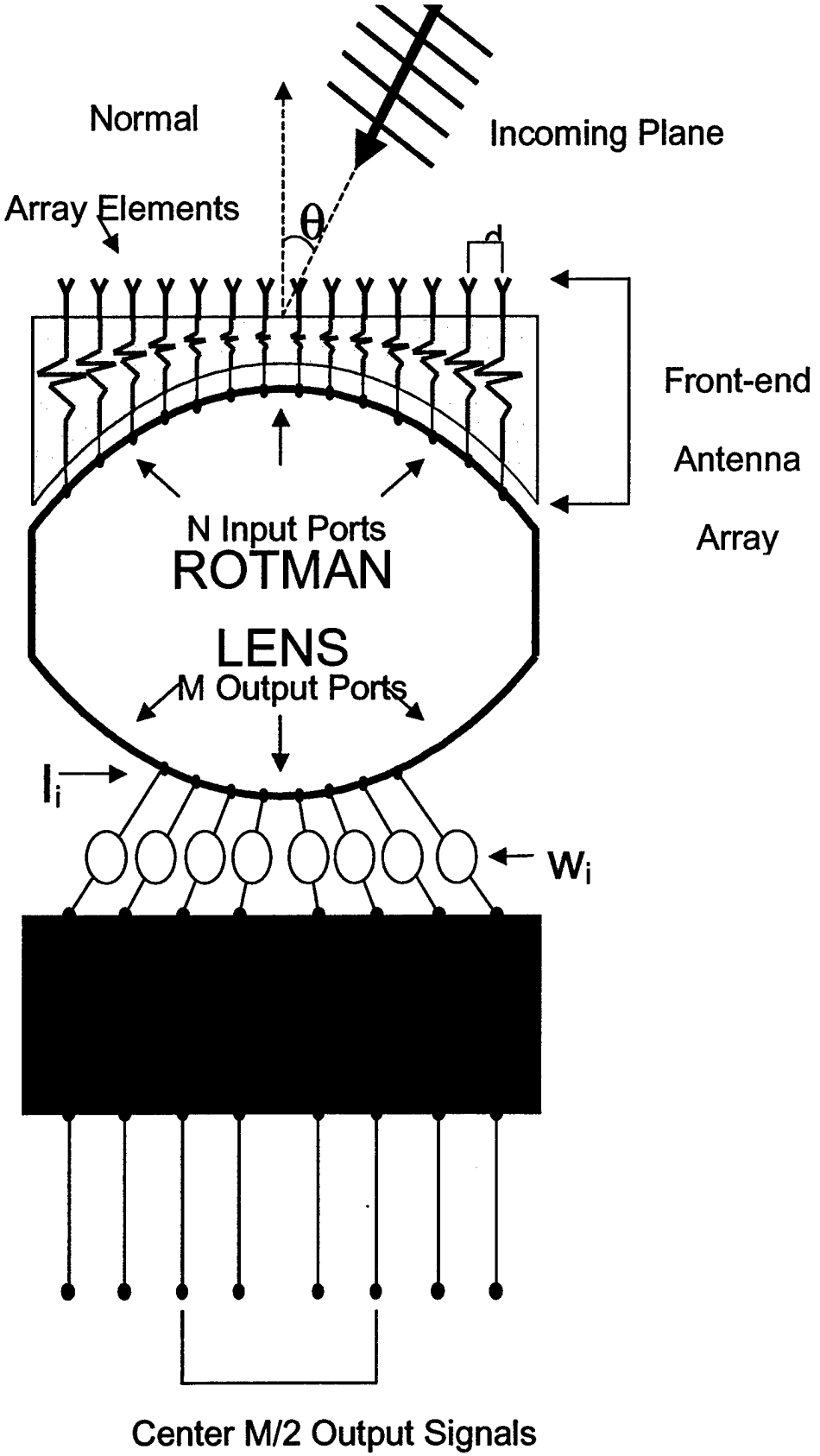


Figure 36: Single section of antenna system, including front-end array, Rotman lens, and Butler matrix.

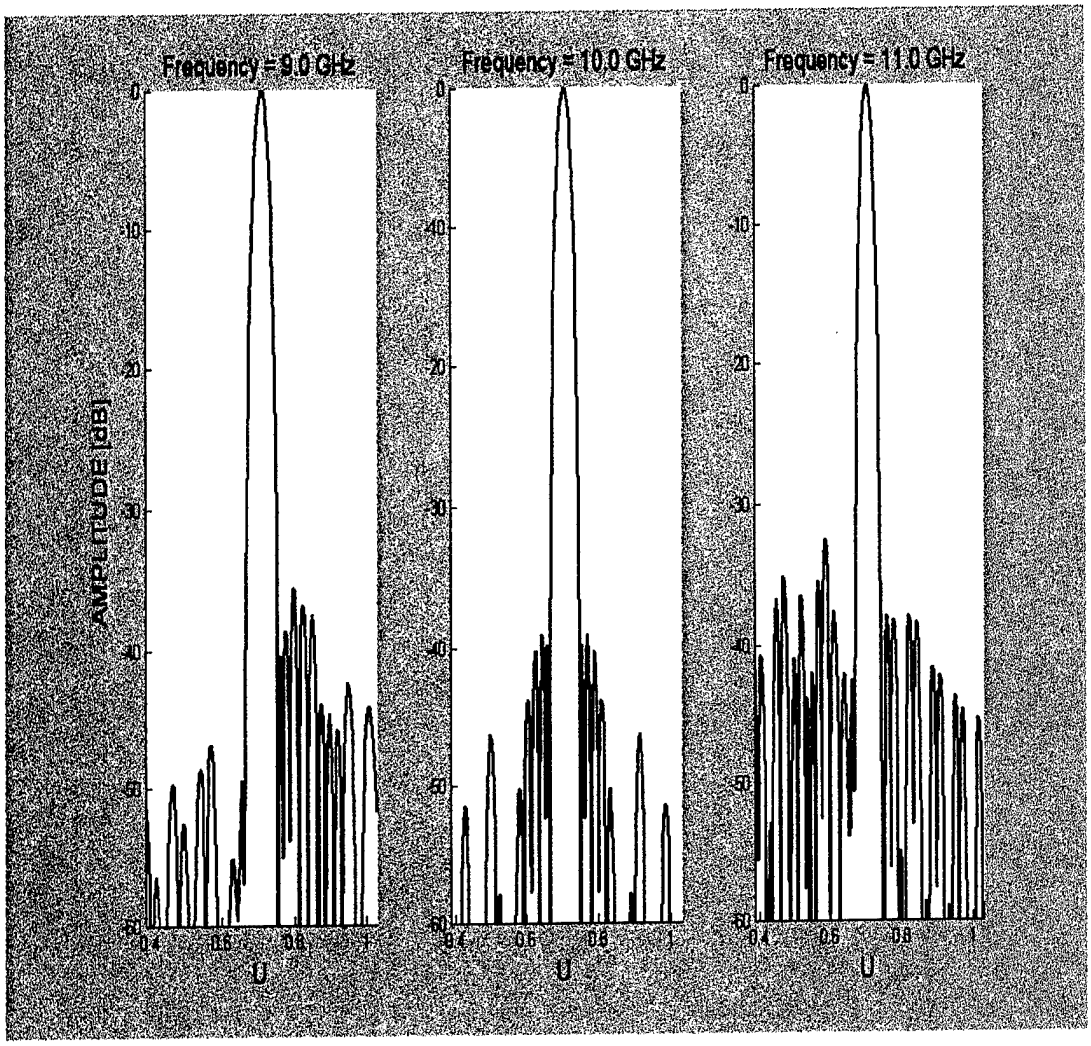


Figure 37: Far-field radiation patterns as a function of frequency for the optimized ideal system.

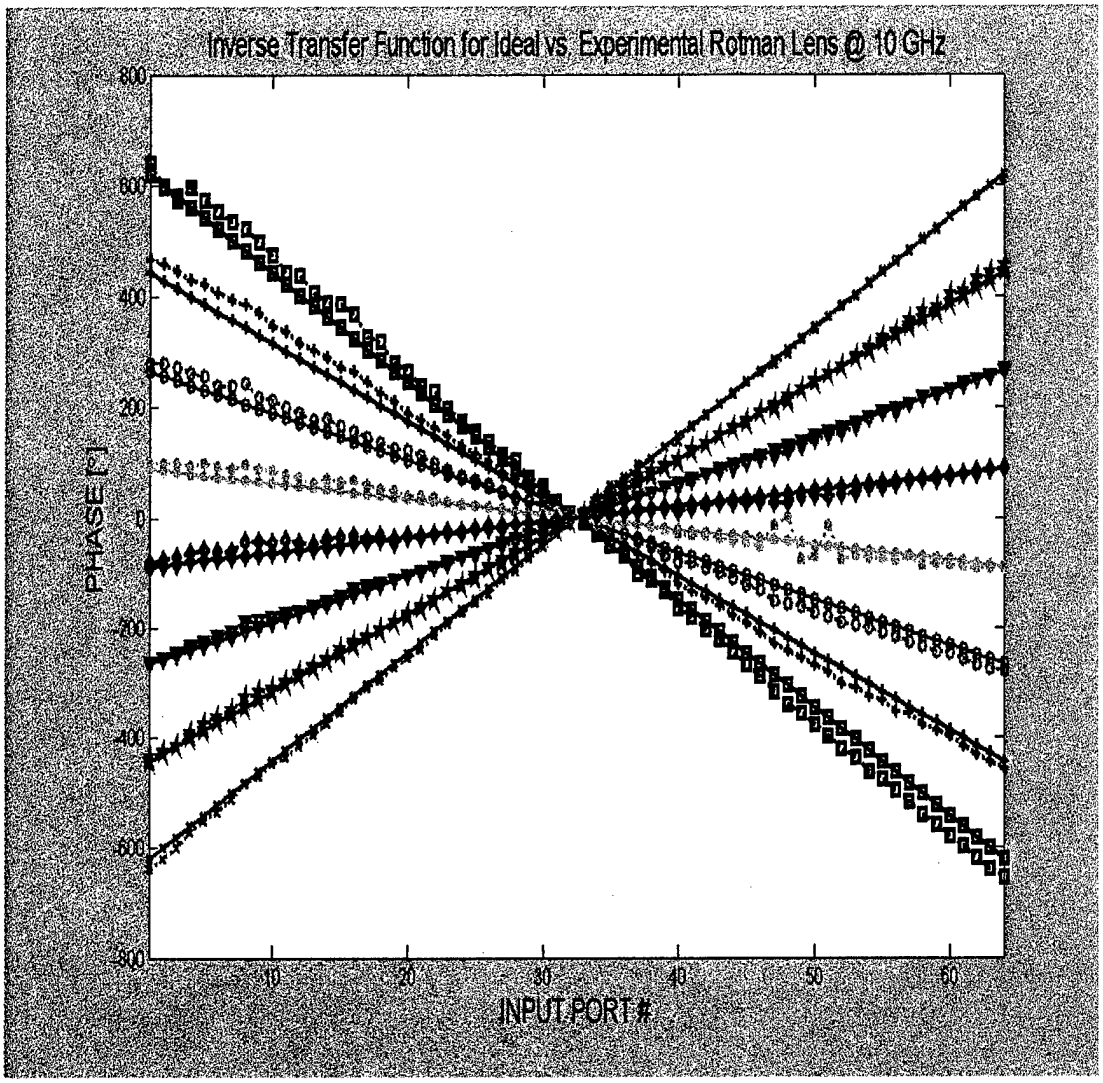


Figure 38: Inverse-transfer-function for a single Rotman lens (phase only). Ideal data: solid, real data: dotted. Output Ports: 1 – square, 2 – "+", 3 – circle, 4 – "*", 5 – diamond, 6 – triangle, 7 – star, 8 – "x".

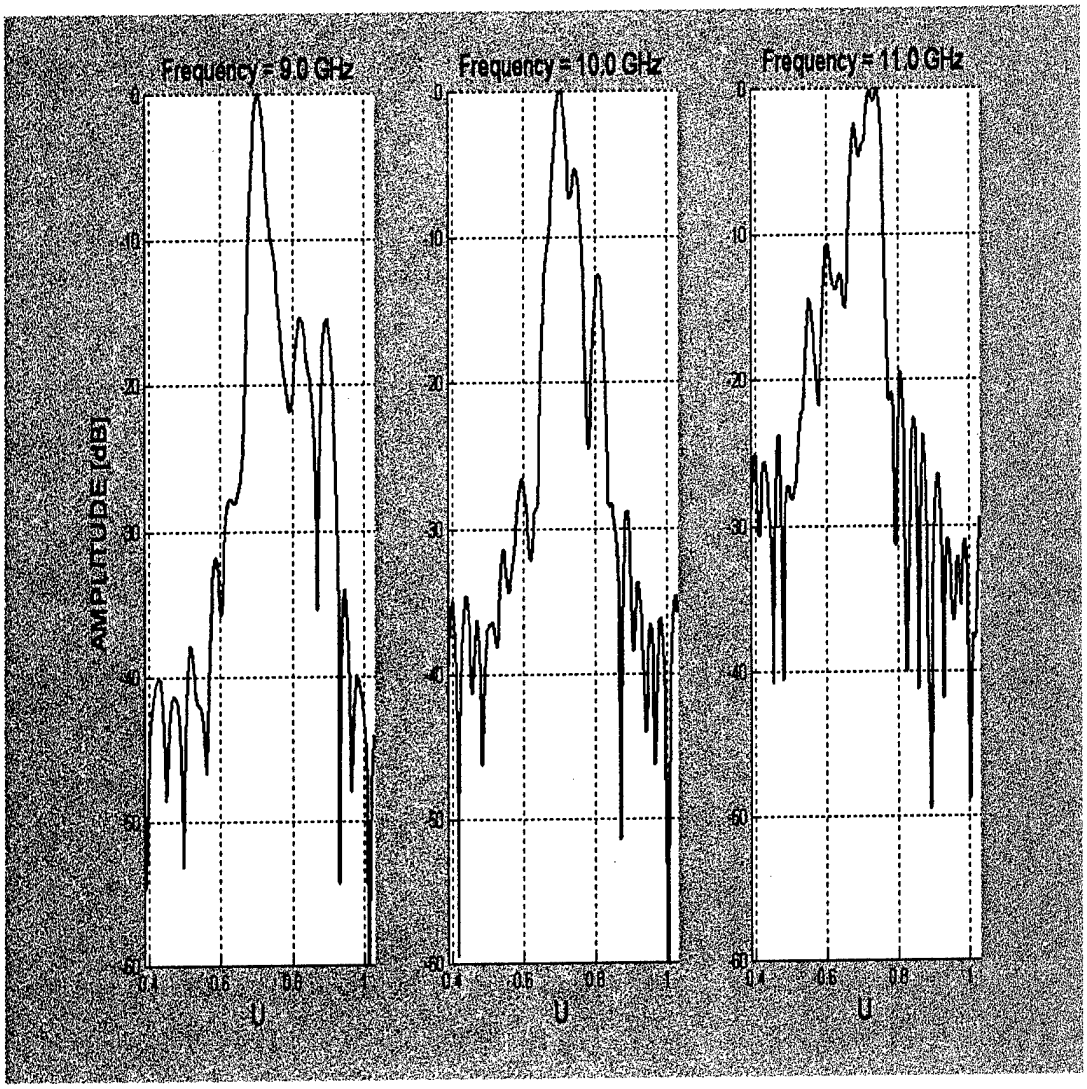
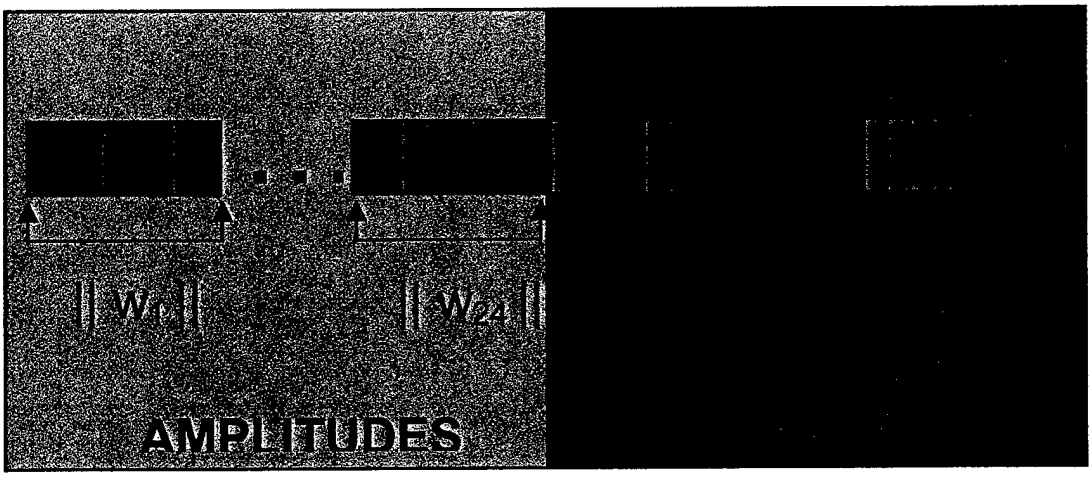


Figure 39: Far-field radiation patterns as a function of frequency when the experimental Rotman lens data is incorporated into the system model and the ideal system weights from [40] are applied.



$$\| w_i \| \subset [0, 1] \qquad \angle w_i \subset [0^\circ, 360^\circ]$$

Figure 40: Chromosomal encoding scheme for SGA

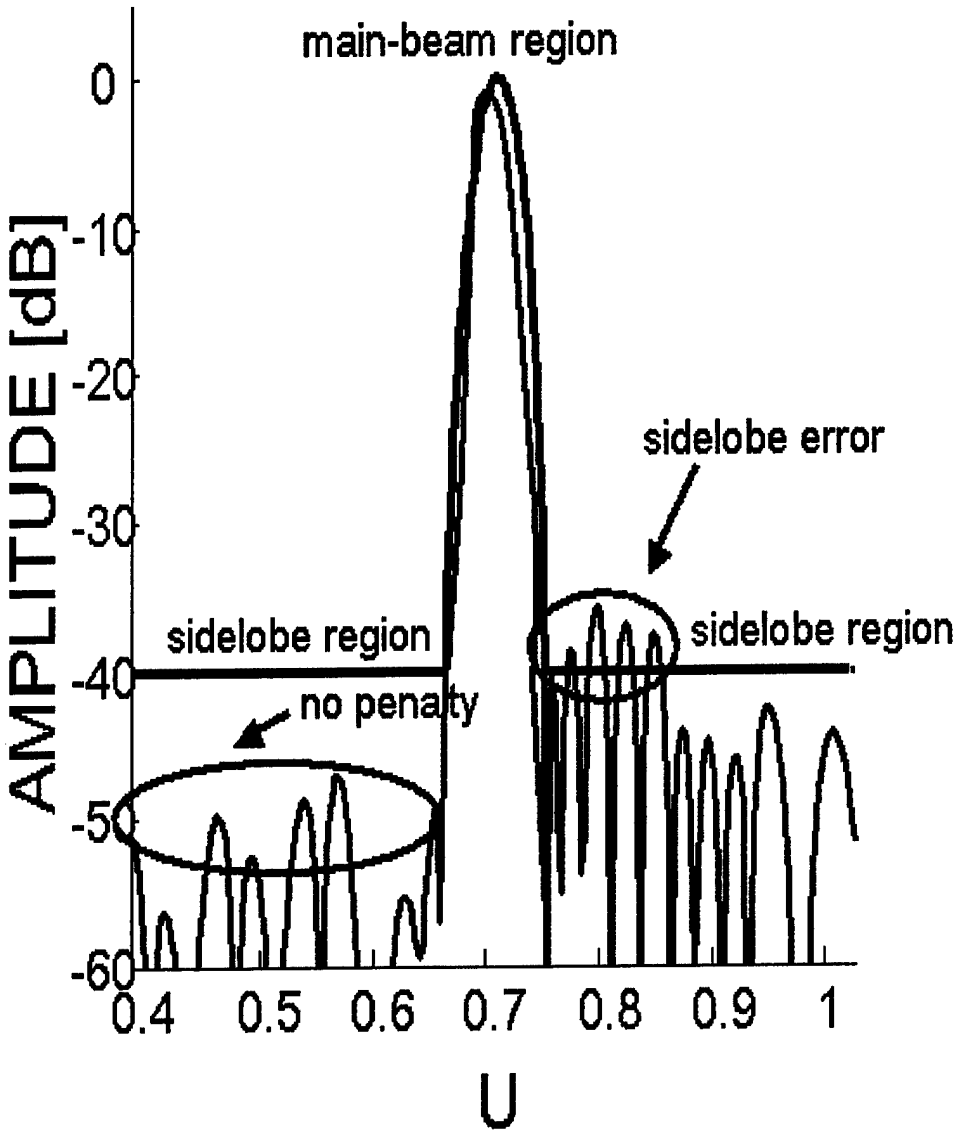


Figure 41: Objective Function for Case 1

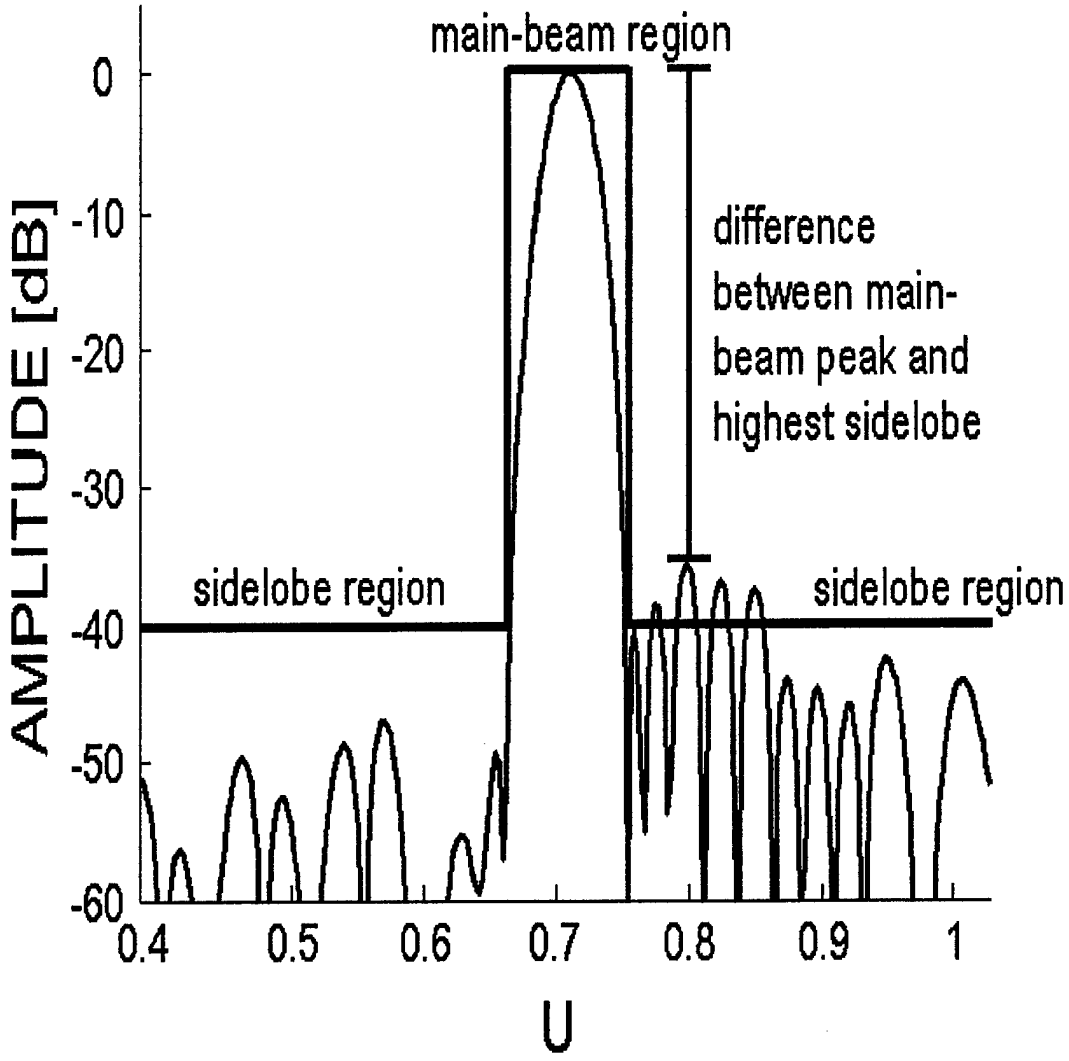


Figure 42: Objective Function for Cases 2 and 3

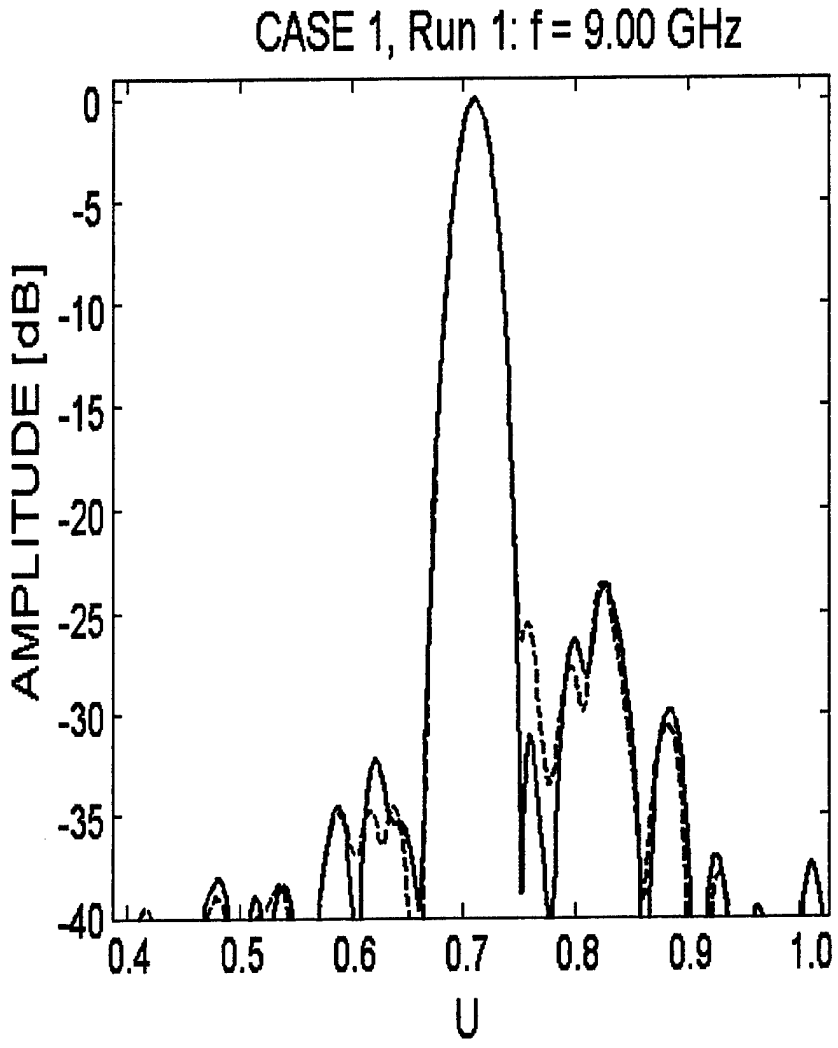


Figure 43: SGA vs. hBOA performance, CASE 1, Run 1. Solid: hBOA, dotted: SGA.

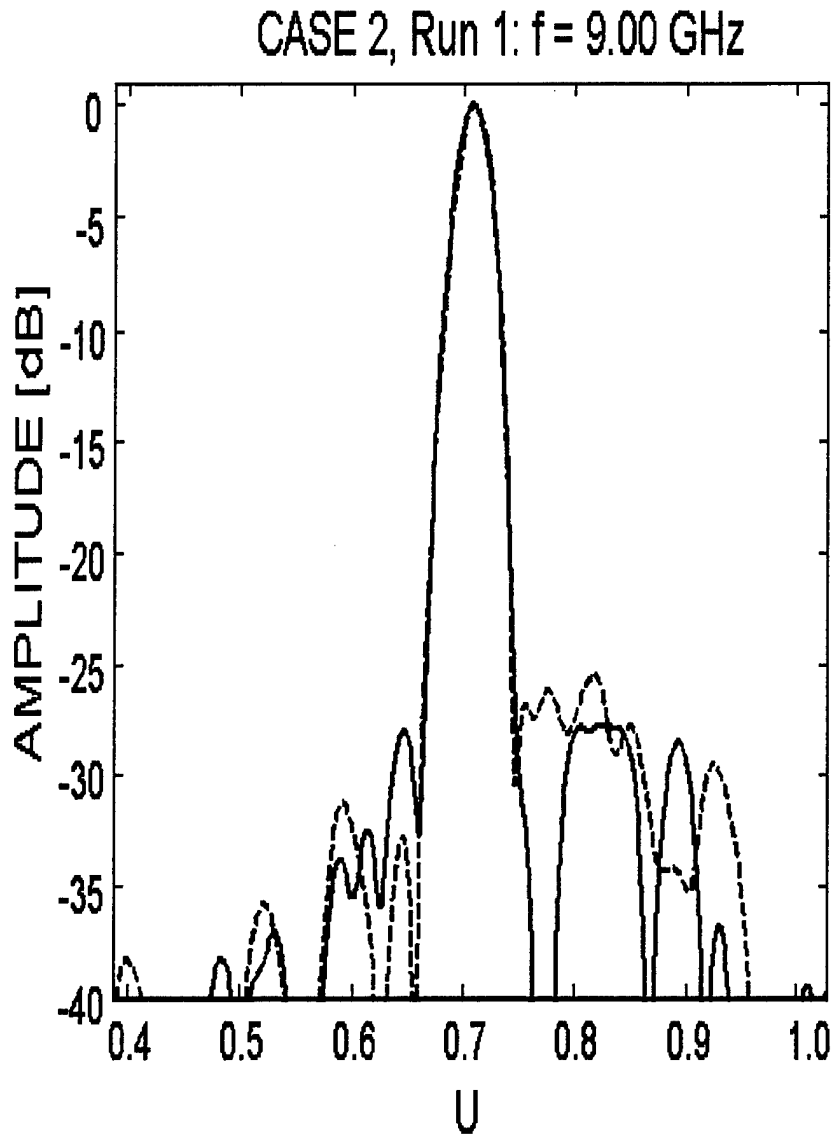


Figure 44: SGA vs. hBOA performance, CASE 2, Run 1. Solid: hBOA, dotted: SGA.

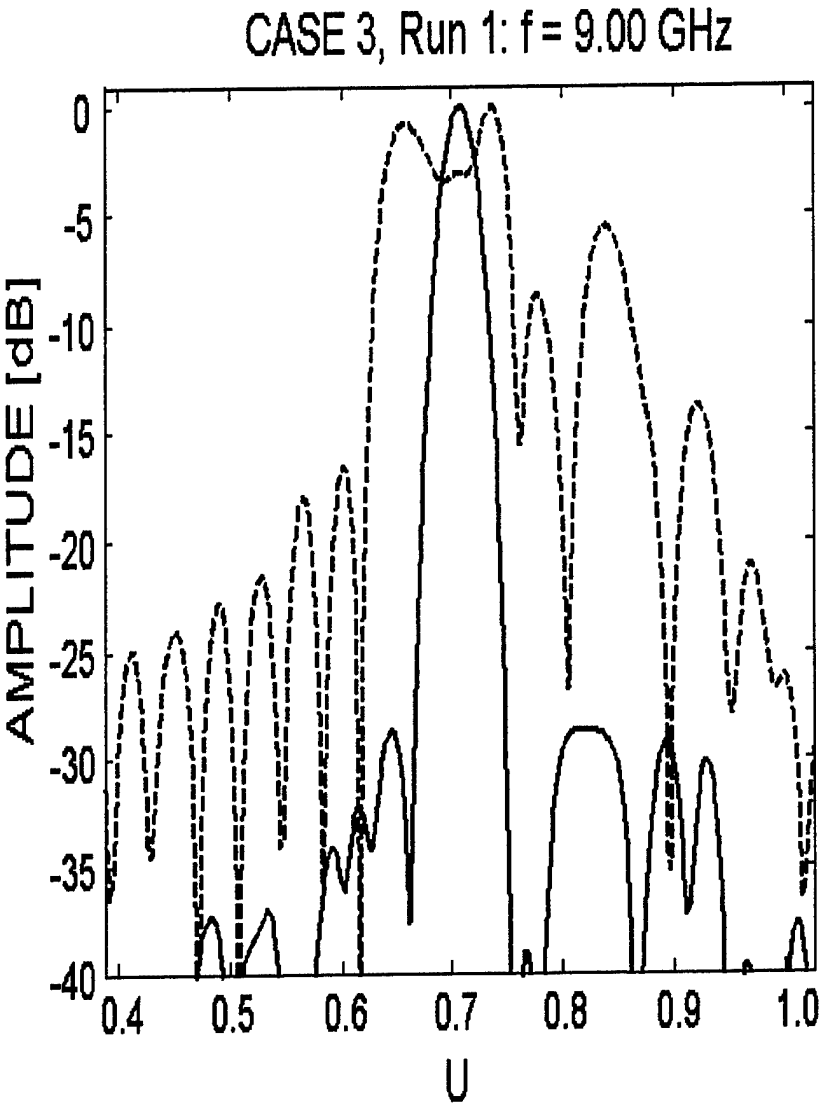


Figure 45: SGA vs. hBOA performance, CASE 3, Run 1. Solid: hBOA, dotted: SGA.

References:

[1] Goldberg, D.E. (1989). *Genetic algorithms in search, optimization, and machine learning.* Reading, MA: Addison-Wesley.

[2] Goldberg, D.E. (2002). *The design of innovation: Lessons from and for competent genetic algorithms.* Boston, MA: Kluwer Academic Publishers.

[3] Pelikan, M., & Goldberg, D.E. (2000). *Hierarchical problem solving by the Bayesian optimization algorithm.* Genetic and Evolutionary Computation Conference (GECCO-2000), pp. 267–274 (also IlliGAL Report No. 2000002).

[4] Pelikan, M., Goldberg D.E., & Sastry, K. (2001). *Bayesian Optimization Algorithm, Decision Graphs, and Bayesian Networks. Genetic and Evolutionary Computation Conference 2001 (GECCO-2001)*, Springer-Verlag, pp. 519-526. Also IlliGAL Report No. 2000020.

[5] Pelikan, M., & Goldberg, D.E. (2001). Escaping Hierarchical Traps with Competent Genetic Algorithms. *Genetic and Evolutionary Computation Conference 2001 (GECCO-2001)*, Springer-Verlag, pp. 511-518. IlliGAL Also IlliGAL Report No. 2001003.

[6] Pelikan, M. (2002). *Bayesian optimization algorithm: From single level to hierarchy.* Ph.D. thesis, University of Illinois at Urbana-Champaign, Urbana, IL. Also IlliGAL Report No. 2002023.

[7] Rahmat-Samii, Y. and Michielssen, E. (Eds.) (1999). *Electromagnetic Optimization by Genetic Algorithms*, Wiley.

[8] Altshuler, E. E. (2000). Design of a Vehicular Antenna for GPS/IRIDIUM using a Genetic Algorithm. *IEEE Trans. Antennas Propagat.*, Vol 48, pp. 968-972.

[9] Burke G. J., & Poffio, A.J. (1981). Numerical Electromagnetics Code (NEC)-Method of Moments. Lawrenc Livermore Lab., CA, Rep.UCID18834, Jan. 1981.

[10] Altshuler, E. E. (2002). Electrically small self-resonant wire antennas optimized using a genetic algorithm. *IEEE Trans. Antennas Propagat.*, vol. 50, pp. 297-300.

[11] Altshuler E. E. (to appear). An Ultra Wideband Impedance-Loaded Genetic Antenna, *IEEE Trans. Antennas Propagat.* (to be published in October 2004 issue).

[12] Altshuler, E.E., Linden, D. S., & Wing, R. A. (1998). Yagi Antenna Design using a Genetic Algorithm, *Communications Quarterly*, Winter 1998.

[13] Haupt, R.L. and Southall, H.L. (1999). Experimental adaptive nulling with a genetic algorithm. *Microwave Journal*, Vol 24, No 1, pp. 78-89.

[14] Southall, H., Kornrumpf, W., Staggs, J., Jonas, F., & King, Y., An array antenna space experiment using transmit/receive antenna module (TRAM) technology. *Proceedings of the 1998 Antenna Applications Symposium*, 16-18 Sep 1998, Allerton Park, Illinois, pp. 160-171.

[15] Franchi, P., Champion, M., Linafelter, D. and Southall, H. (1997). Constrained feed for lightweight overlapped subarray antennas, *Proceedings of the 1997 Antenna Applications Symposium*, Sept. 1997, Allerton Park, Illinois.

[16] Skobolev, S.P. (1990). Analysis and synthesis of an antenna array with sectoral partial radiation patterns, *Radiotechnika*, No 10, pp. 44-47.

[17] Haupt, R.L. (1997). Phase-only adaptive nulling with genetic algorithms. *IEEE APS Transactions*, Vol 45, No 6, pp. 1009-1015.

[18] Haupt, R.L. (1995). An introduction to genetic algorithms for electromagnetics. *IEEE Antennas and Propagation Magazine*, Vol 37, No 2, pp. 7-15.

[19] Altshuler, E.E. (2002). Electrically small self-resonant wire antennas optimized using a genetic algorithm. IEEE *Antennas and Propagation Magazine*, Vol. 50, No 3, pp. 297–300.

[20] Altshuler E.E., & Linden, D.S. (1997). Design of a loaded monopole having hemispherical coverage using a genetic algorithm. *IEEE Antennas and Propagation Magazine*, Vol 45, No 1, pp. 1–4.

[21] Altshuler, E.E., & Linden D. (1997). Wire-antenna designs using genetic algorithms, *IEEE Antennas and Propagation Magazine*, Vol 39, No 2.

[22] O'Donnell, T., Altshuler, E.E, & Best, S. (2003). The significance of genetic representation in genetic antenna design. *2003 IEEE Antennas and Propagation Symposium*, Vol 1, Section 7-3.

[23] O'Donnell, T., Best, S., Altshuler, E.E., Hunter, J., Bullett, T., & Barton, R. (2003). Genetic algorithm optimization: A tale of two chromosomes, *2003 Antenna Applications Symposium*, Monticello, IL Sept 2003, pp. 131–146.

[24] *Adaptive Antenna Array for Radar Cross Section (RCS) Measurements, SBIR Phase I Final Report*, SBIR Topic #AF98-279. Copywrite ARCON Corporation, Waltham, MA 02451.

[25] Pelikan, M., Goldberg, D.E., & Cantú-Paz, E. (1999). BOA: The Bayesian optimization algorithm. *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-99)*, Vol I, pp. 525–532. (Also IlliGAL Report No. 99003)

[26] Holland, J.H. (1975). *Adaptation in natural and artificial systems.* Ann Arbor, MI: University of Michigan Press.

[27] Goldberg, D.E. (1991). Theory tutorial. (Tutorial presented with G. Liepens at the 1991 International Conference on Genetic Algorithms, La Jolla, CA).

[28] Goldberg, D.E., Deb, K., & Clark, J. H. (1992). Genetic algorithms, noise, and the sizing of populations. *Complex Systems*, 6 , pp. 333—362. (Also IlliGAL Report No. 91010).

[29] Goldberg, D.E.,(1989). Genetic algorithms and Walsh functions: Part II, deception and its analysis. *Complex Systems*, 3, pp. 153—171.

[30] Deb, K. & Goldberg, D.E. (1993). Analyzing deception in trap functions. *Foundations of Genetic Algorithms 2*, pp. 93—108.

[31] Goldberg, D.E. (1989). Sizing Populations for serial and parallel genetic algorithms. *Proceedings of the Third International Conference on Genetic Algorithms*, pp. 70–79.

[32] Goldberg, D.E., Sastry, K., & Latoza, T. (2001). On the supply of building blocks, *Proceedings of the Genetic and Evolutionary Computation Conference*, pp. 336–342.

[33] Goldberg, D. E., Deb, K., & Clark, J. H. (1992). Genetic algorithms, noise, and the sizing of populations. *Complex Systems*, 6 , pp. 333—362. (Also IlliGAL Report No. 91010).

[34] Harik, G., Cantú-Paz, E., Goldberg, D.E., & Miller, B.L. (1997). The gambler's ruin problem, genetic algorithms, and the sizing of populations. *Proceedings of 1997 IEEE International Conference on Evolutionary Computation* , pp. 7–12.

[35] Sastry, K. & Goldberg, D. E. (2002). *Analysis of mixing in genetic algorithms: A survey.* IlliGAL report no. 2002012.

[36] Larrañaga, P., Lozano, J.A. (Eds.). (2002). *Estimation of Distribution Algorithms.* Boston, MA: Kluwer Academic Publishers.

[37] Howard, R.A., & Matheson, J.E. (1981). Influence diagrams. In Howard, R. A., & Matheson, J. E. (Eds.), *Readings on the principles and applications of decision analysis*, Volume II, pp. 721-762, Menlo Park, CA: Strategic Decisions Group.

[38] Simon, H.A. (1968). *The sciences of the artificial.* Cambridge, MA: The MIT Press.

[39] Pelikan, M. & Goldberg D.E. (2003). Hierarchical BOA solves Ising spin glasses and MAXSAT. *Genetic and Evolutionary Computation Conference 2003 (GECCO-2003)*, Springer-Verlag, pp. 1271-1282 (Also IlliGAL Report No. 2003001).

[40] Mailloux, R. J. (2001). A low-sidelobe partially overlapped constrained feed network for time-delayed subarrays. *IEEE Transactions on Antennas and Propagation*, Vol 49, No 2, pp. 280–291.

[41] Kraus, J.D. (1988). *Antennas* (Second Edition), McGraw-Hill, Inc.