



**AFRL-RY-HS-TR-2011-0011**

---

**Antenna Design Using the Efficient Global Optimization (EGO) Algorithm**

**Hugh L. Southall and Terry H. O'Donnell\***

**On-site contractor/AFRL/Ryha Hanscom AFB, MA  
Solid State Scientific Corporation  
27-2 Wright Road  
Hollis, NH 03049**

**\* Formerly: On-site contractor/AFRL/Ryha Hanscom AFB, MA  
ARCON Corporation  
260 Bear Hill Road  
Waltham, MA 02451**

**Currently: Electronic Systems Center  
ESC/XRC  
15 Eglin Street  
Hanscom AFB, MA 01731**

**Final Report**

**20 May 2011**

**APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED**

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

## NOTICE AND SIGNATURE PAGE

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 U.S. 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.

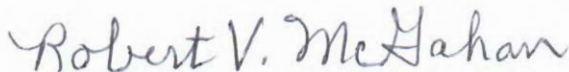
This report was cleared for public release by the Electronic Systems Center Public Affairs Office for the Air Force Research Laboratory Electromagnetic Technology Division and is available to the general public, including foreign nationals. Copies may be obtained from the Defense Technical Information Center (DTIC) (<http://www.dtic.mil>).

AFRL-RY-HS-TR-2011-0011 HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION IN ACCORDANCE WITH ASSIGNED DISTRIBUTION STATEMENT.



---

DAVID D. CURTIS  
Chief, Antenna Technology Branch



---

ROBERT V. McGAHAN  
Technical Communications Advisor  
Electromagnetic Technology Division

This report is published in the interest of scientific and technical information exchange, and its publication does not constitute the Government's approval or disapproval of its ideas or findings.

# REPORT DOCUMENTATION PAGE

*Form Approved*  
*OMB No. 0704-0188*

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing this collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. **PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.**

<b>1. REPORT DATE</b> ( <i>DD-MM-YYYY</i> ) 05-20-2011		<b>2. REPORT TYPE</b> FINAL REPORT		<b>3. DATES COVERED</b> ( <i>From - To</i> ) 1 Sep 2007 – 1 Jun 2011	
<b>4. TITLE AND SUBTITLE</b>  Antenna Design Using the Efficient Global Optimization (EGO) Algorithm				<b>5a. CONTRACT NUMBER</b>	
				<b>5b. GRANT NUMBER</b>	
				<b>5c. PROGRAM ELEMENT NUMBER</b>	
<b>6. AUTHOR(S)</b>  Hugh L. Southall, Terry H. O'Donnell				<b>5d. PROJECT NUMBER</b> 4916	
				<b>5e. TASK NUMBER</b> HA	
				<b>5f. WORK UNIT NUMBER</b> 11	
<b>7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)</b> Solid State Scientific Corporation 27-2 Wright Road Hollis, NH 03049 ESC/XRC, 15 Eglin Street Hanscom AFB MA 01731				<b>8. PERFORMING ORGANIZATION REPORT</b>	
<b>9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)</b> Electromagnetics Technology Division Sensors Directorate Air Force Research Laboratory 80 Scott Drive Hanscom AFB MA 01731-2909				<b>10. SPONSOR/MONITOR'S ACRONYM(S)</b> AFRL-RY-HS	
				<b>11. SPONSOR/MONITOR'S REPORT NUMBER(S)</b> AFRL-RY-HS-TR-2011-0011	
<b>12. DISTRIBUTION / AVAILABILITY STATEMENT</b> APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED					
<b>13. SUPPLEMENTARY NOTES</b> Public Affairs release number: 66 ABW-2011-0502					
<b>14. ABSTRACT</b> In this report, we discuss antenna design optimization using the efficient global optimization (EGO) algorithm. The first design is a parasitic super directive array where we compare EGO with a classic optimization technique called Nelder-Mead and then with GA optimization. The second antenna design optimization problem is a wideband antenna element backed by both a metamaterial ground plane and a conducting ground plane. The antenna element over a conducting ground plane was fabricated and we present a comparison of test results with predicted results. The third antenna design is an infinite, periodic array of fragmented patch elements. We also investigate two potential areas of improvement to the EGO algorithm. The first is in the "endgame" where we consider techniques to make the algorithm more accurate in the final estimate of the location of the global minimum. The last area involves the selection of initial data sets with the goal of more efficient algorithms.					
<b>15. SUBJECT TERMS</b> Antenna design optimization, efficient global optimization (EGO), genetic algorithms (GA), kriging, metamaterials, evolutionary computation, wideband antennas, electrically small antennas, design and analysis of computer experiments					
<b>16. SECURITY CLASSIFICATION OF:</b>			<b>17. LIMITATION OF ABSTRACT</b>	<b>18. NUMBER OF PAGES</b>	<b>19a. NAME OF RESPONSIBLE PERSON</b>
<b>a. REPORT</b>	<b>b. ABSTRACT</b>	<b>c. THIS PAGE</b>			David D. Curtis
Unclassified	Unclassified	Unclassified	UU	92	<b>19b. TELEPHONE NUMBER</b> ( <i>include area code</i> ) N/A



## Table of Contents

<b>1. SUMMARY</b> .....	1
<b>2. INTRODUCTION</b> .....	1
<b>3. METHODS , ASSUMPTIONS AND PROCEDURES</b> .....	2
<b>3.1 The Efficient Global Optimization (EGO) Algorithm</b> .....	2
<b>3.1.1 EGO: A Balance between Local and Global Search</b> .....	2
<b>3.1.2 The DACE Stochastic Process Model</b> .....	4
<b>3.1.3 Estimation of Correlation Parameters</b> .....	5
<b>3.1.4 Model Validation and the DACE Predictor</b> .....	6
<b>3.1.5 Selecting the Next Design Point</b> .....	7
<b>3.1.6 Convergence Criteria</b> .....	9
<b>3.1.7 A Simple One-Dimensional Example Using EGO</b> .....	9
<b>3.2 Genetic Algorithm (GA) Optimization</b> .....	12
<b>3.2.1 The Simple Textbook GA</b> .....	12
<b>3.2.2 An Enhanced GA</b> .....	14
<b>4. RESULTS AND DISCUSSIONS</b> .....	16
<b>4.1 Parasitic Super Directive Arrays (PSA) Design Optimization</b> .....	16
<b>4.1.1 The Test131 PSA and Associated Function Space</b> .....	18
<b>4.1.2 EGO Algorithm Parameters Used for Testing</b> .....	18
<b>4.1.3 Results from the EGO Design Optimization</b> .....	19
<b>4.1.4 Comparison of EGO and Nelder-Mead Downhill Simplex Algorithm</b> .....	21

4.1.5	Comparison of EGO with a GA for the PSA Design Problem .....	22
4.2	The Wideband Folded Triangular Bowtie Antenna (FTBA) Element .....	27
4.2.1	The FTBA Element .....	27
4.2.2	The FTBA Element Over a Metamaterial Ground Plane .....	29
4.2.3	The FTBA Element Over a PEC Ground Plane .....	31
4.2.4	The Antenna Balun/Feed for the FTBA Element .....	32
4.2.5	Multi-Parameter Design Optimization Using EGO .....	33
4.2.6	Coupling EGO with a Full Wave CEM Simulation .....	35
4.2.7	Antenna Designs using EGO/HFSS .....	36
4.2.8	The Experimental FTBA Over a PEC Ground Plane .....	37
4.2.9	Measured Data for the FTBA over a PEC Ground Plane .....	38
4.3	The Wideband Fragmented Patch Antenna Element .....	44
4.3.1	Introduction .....	44
4.3.2	The Fragmented Patch Antenna Element .....	45
4.3.3	Patch Modeling .....	46
4.3.4	Cost Function .....	47
4.3.5	Simple GA Optimization .....	47
4.3.6	EGO Optimization .....	50
4.3.7	Optimization Results for the Fragmented Patch Antenna .....	51
4.4	Endgame Techniques for EGO .....	54
4.4.1	Generalized Expected Improvement .....	55
4.4.2	An Ad Hoc Approach .....	57
4.4.3	Engineer-in-the-Loop Approach .....	57
4.4.4	Results using Endgame Techniques .....	57
4.4.4.1	Results Using Modifications to the EGO Algorithm .....	58

## List of Figures

Figure 1. Flowchart of the EGO algorithm .....	3
Figure 2. Uncertainty in the value of the cost function $y(x)$ at the point $x^*$ .....	8
Figure 3. One-D EGO Example; Iteration Number One.....	10
Figure 4. One-D EGO Example; Iteration Number Two .....	10
Figure 5. One-D EGO Example; Iteration Number Three .....	11
Figure 6. (a) Two electrically small antennas in a parasitic super directive array configuration. (b) A comparison of the driven super directive gain achievable with these elements with the array at element resonance; the parasitic gain if the array is kept at element resonances (i.e. not frequency shifted); and the parasitic gain possible if the frequency is optimized at each separation .....	17
Figure 7. (a) Configuration of Test131 parasitic array, with elements separated by 20 mm. (b) The full simulated function (input) space showing the possible directivities of the Test131 PSA for element separations ranging from 5mm to 60mm and frequencies from 800 to 900 MHz .....	18
Figure 8. (a) A poor initial sample point distribution, as generated by the Latin hypercube. Note the sample point very close to the non-optimal minima; (b) Final distribution of sample points after the algorithm converged, and (c) Location of the predicted minimum. The DACE predictor function space in (c) typically provides a closer match to the exact function space; however, even under these extremely poor starting conditions, the algorithm was still able to find the global minimum and approximate the function space reasonably well .....	21
Figure 9. The FTBA element backed by a metal cavity. The coordinates are $x,y,z$ counterclockwise from the lower left arrow. The two narrow folding arms are at either side of the bowtie element .....	28
Figure 10. The FTBA element backed by a disk of indefinite material. Principal design parameters are shown. The back surface of the disk is covered by a conducting plane (not visible) .....	28
Figure 11. Side view of the FTBA element backed by an indefinite material. Principal design parameters are shown .....	29
Figure 12. Transverse Electric (TE) wave incident on a metamaterial.....	29
Figure 13. Total radiated power for three cases of an ideal dipole over an infinite ground plane consisting of: a Perfect Magnetic Conductor (PMC), an anti-cutoff indefinite material, or a Perfect Electric Conductor (PEC).....	31
Figure 14. FTBA over a finite PEC ground plane (disk) .....	32
Figure 15. Front view of the balun/feed structure.....	33
Figure 16. (a) The “under side” of the balun/feed structure on a double-sided Rogers 5880 substrate. (b) The “trace” side of the balun/feed structure showing the coaxial SMA connector on the bottom of the ground plane. The center conductor of the SMA connector is soldered to the top trace of the microstrip portion of the feed/balun .....	37
Figure 17. (a) Top view of the FTBA element over both square and circular ground planes. Only the circular ground plane was used for measurements to compare with predicted results. (b) Side	

view of the FTBA element showing a solder tab at the top of the feed structure soldered to a leg of the dipole .....	38
Figure 18. Predicted and measured return loss in dB for the FTBA over a PEC ground plane.....	39
Figure 19. Predicted (blue) and measured (green) results for the broadside gain (in dB over an isotropic radiator) for the FTBA element over a circular PEC ground plane. The results from Qu, et al, for the same antenna configuration but different antenna dimensions are shown by the red curve .....	40
Figure 20. The FTBA element on the antenna mount at the AFRL/Ryha Ipswich Antenna Research Facility, Ipswich, MA. The standard gain horns to the left of the mount were used in the gain measurements. The white frame is foam structural support. In this configuration the antenna is horizontally polarized. ....	41
Figure 21. E-plane co-polarized response and cross-polarized responses at 2.0 GHz .....	41
Figure 22. E-plane co-polarized and cross-polarized responses at 3.5 GHz .....	42
Figure 23. E-plane co-polarized and cross-polarized responses at 5.0 GHz .....	42
Figure 24. H-plane co-polarized and cross-polarized responses at 2.0 GHz.....	43
Figure 25. H-plane co-polarized and cross-polarized responses at 3.5 GHz.....	43
Figure 26. H-plane co-polarized and cross-polarized responses at 5.0 GHz.....	44
Figure 27. Infinite planar array comprised of fragmented patch unit cells. Horizontal and vertical cross sections of unit cell are shown. The cell has two substrates and two superstrates. Conducting pixels are shown in red.....	45
Figure 28. Design optimization utilizes only the upper left quadrant of the patch antenna element as shown above.....	46
Figure 29. Fragmented patch parameter encoding for the simple GA design optimization .....	48
Figure 30. Flow diagram for the simple GA design optimization algorithm.....	49
Figure 31. Pre-defined pixel geometry known as a “butterfly” patch.....	49
Figure 32. Encoding Scheme for the EGO design optimization algorithm .....	50
Figure 33. Convergence comparison of EGO with the best GA results .....	52
Figure 34. Comparison of GA and EGO patch solutions to the Butterfly patch.....	52
Figure 35. Wideband scan characteristics for the GA and EGO solutions .....	54
Figure 36. Results for the “g parameter” endgame technique where the algorithm gets trapped out in the very flat valley. (a) Test131 function space with 101 data points (b) DACE predictor (c) Standard error (d) Expected improvement.....	60
Figure 37. Results for the “g parameter” endgame technique where the “exact” global minimum was found to be -10.263 dB at $\mathbf{x} = [8.37\text{mm} \ 847 \text{MHz}]$ .....	61

Figure 38. Results for the “linearly decreasing ratio” endgame technique which resulted in a very local search. The difference from the global minimum was 0.08% .....	61
Figure 39. Correlation coefficients $\theta_1$ and $\theta_2$ as a function of the algorithm iteration number.....	62
Figure 40. (a) <b>Test55</b> antenna array geometry. The colors represent current density magnitude. (b) The response surface. The red dot is the global minimum where the directivity is -9.981 dB at $\mathbf{x} = [100.5\text{mm } 435\text{ MHz}]$ found using an exhaustive search.....	62
Figure 41. Comparison of results obtained using the different optimization algorithms. The error bars represent slightly different estimates of the global minimum which are found when a different set of initial samples is selected for each run.....	64
Figure 42. The <i>egg crate</i> function.....	68
Figure 43. (a) OMLHD (21, 2) design $\mathbf{D}_{\text{test2}}$ (b) Typical LHD (21, 2) design .....	70
Figure 44. EGO performance with (a) OMLHD initial data set. (b) LHD initial data set. Shown is the number of occurrences (frequency) of the total number of function evaluations required for convergence .....	71
Figure 45. CPGA performance for (a) OMLHD (24, 4) and (b) LHD (24, 4) initial data sets.....	72
Figure 46. Convergence for the OMLHD initial data set and EGO design optimization of the indefinite material unit cell. The optimum values found for the four design variables are shown below the picture of the unit cell configuration. ....	74
Figure 47. Performance curves for the OMLHD initial data set and EGO design optimization of the indefinite material unit cell. Real and imaginary parts of the permittivity, permeability, normalized impedance and refractive index are shown clockwise from upper left. The performance goals are negative one for both the real part of the permittivity and the real part of the permeability.....	74
Figure 48. Convergence for the LHD initial data set and EGO design optimization of the indefinite material unit cell. The optimum values found for the four design variables are shown below the picture of the configuration. ....	75
Figure 49. Performance curves for the LHD initial data set and EGO design optimization of the indefinite material unit cell. Real and imaginary parts of the permittivity, permeability, normalized impedance and refractive index are shown clockwise from upper left. The performance goals are negative one for both the real part of the permittivity and the real part of the permeability .....	75

## List of Tables

Table 1. Summary of the EGO run for the simple 1D problem.....	11
Table 2. Summary of the Model Validation Results for the 1D Problem.....	12
Table 3. Fifteen test runs of EGO on the Test131 PSA. Initial data set consists of 21 samples. The global minimum is at -10.262 dB .....	20
Table 4. Fifteen runs of the Nelder-Mead Simplex Algorithm on the Test131 PSA. The global minimum in the end fire directive gain is -10.262 dB .....	22
Table 5. Average (over 30 test runs) results for the textbook GA at convergence as a function of the initial population and the mutation rate, $\delta$ .....	23
Table 6. Distribution of the number of each stopping criterion used for the textbook GA for the three initial populations and five mutation rates, $\delta$ .....	24
Table 7. Average predicted directivity for the textbook GA at 71 cost function calls.....	25
Table 8. ERGA Results for the PSA Problem .....	26
Table 9. Ranges for the six design parameters for the folded triangular bowtie antenna (FTBA) over an indefinite material .....	34
Table 10. Antenna design variables from the EGO/HFSS design optimization for the three FTBA element designs.....	36
Table 11. Comparison of GA and EGO best results .....	53
Table 12. Estimates of the directivity (dB) and the difference (in %) from the true global maximum are shown for both EGO endgame techniques for 10 typical runs .....	59
Table 13. List of parameters for the optimum indefinite material unit cell designs for OMLHD and LHD initial data sets.....	73



4.4.4.2	Results Using the Engineer-in-the-Loop Technique.....	62
4.4.5	Observations on Endgame Techniques.....	63
4.4.6	Comparison of Results using EGO (with Endgame) and the GA.....	64
4.5	Orthogonal-Maximin LHD (OMLHD) Initial Data Sets.....	65
4.5.1	Introduction.....	65
4.5.2	Generating OMLHD Data Sets.....	65
4.5.3	A Two-Dimensional Problem using EGO.....	68
4.5.4	A Four-Dimensional Design Problem using a GA.....	71
4.5.5	A Four-Dimensional Design Problem using EGO.....	73
<b>5.</b>	<b>CONCLUSIONS.....</b>	<b>76</b>
<b>6.</b>	<b>RECOMMENDATIONS.....</b>	<b>77</b>
	<b>APPENDIX.....</b>	<b>78</b>
	<b>REFERENCES.....</b>	<b>79</b>

## 1. SUMMARY

In many engineering design problems one is required to find the “best” values of input parameters (design variables) based on the evaluation of an objective (cost) function. Modern computer simulations used in such problems often have objective functions which can be very expensive in time and/or cost. These so-called expensive black-box functions, as described in the paper “Efficient Global Optimization of Expensive Black-Box Functions” by Jones, et al, [1] can require a significant amount of computation time. For example, a single automotive crash simulation can take up to 20 hours [1]. Complex computational electromagnetic (CEM) codes used for antenna simulation, modeling and design can also require significant resources.

The Efficient Global Optimization (EGO) algorithm is a competent, data adaptive, evolutionary-computation (EC) algorithm suited for problems with a limited number of design parameters and expensive cost functions [1]. Many electromagnetic problems fall into this class. This makes evolutionary algorithms such as genetic algorithms (GA) or particle swarm optimization (PSO) very expensive, since iterations of large populations involving many cost function evaluations are usually required. When physical experiments are necessary to perform tradeoffs or to determine effects which cannot be simulated, use of these algorithms may not be practical due to the large numbers of measurements required.

In this report, we discuss antenna design optimization using EGO. The first antenna design is a parasitic super directive array where we compare EGO with a classic optimization technique called Nelder-Mead and also with a GA optimizer. The second antenna design problem is a wideband antenna element backed by a metamaterial ground plane in the first case and by a perfect electric conductor (PEC) ground plane in the second case. The antenna element over the PEC ground plane was fabricated and tested. We compare measured results with predicted results. The third design is an infinite, periodic antenna array of fragmented patch radiating elements. We also investigate two potential areas of improvement to the EGO algorithm: (1) the “endgame” where we consider techniques to make the algorithm more accurate in the final estimate of the location of the global minimum; and (2) the selection of a “better” initial data set (to start the algorithm) with the goal of obtaining more efficient algorithms.

## 2. INTRODUCTION

Many EC algorithms have been applied to the design and optimization of electromagnetic problems. These have ranged from simple GAs [2, 3] to more complex competent GAs and genetic programming techniques [4, 5 and 6]. Successful applications have been demonstrated in a wide variety of military and other government antenna systems [5, 7, 8 and 9]. However, many evolutionary algorithms require large numbers of cost function evaluations. This can be prohibitive for complex computational electromagnetic problems, in which each proposed solution takes significant computational resources, or where physical experiments are required to evaluate a proposed solution.

We propose the EGO algorithm as an alternative to other evolutionary search algorithms for optimizing electromagnetic design problems with expensive cost functions. Like GAs, EGO performs both global and local searches simultaneously in order to fully explore the function space and avoid becoming trapped

in local minima. Unlike the GA, EGO creates a model of the response surface called the DACE predictor (Design and Analysis of Computer Experiments) [1, 10] which is refined throughout the search. The model is used to predict areas of the function space that warrant further exploration, either because they are close to known good areas (local search) or because they have been insufficiently explored and exhibit uncertainty (global search). A single data point is evaluated per iteration and the results of this evaluation are used to further refine the response surface model. In GA optimization most of the time is spent evaluating numerous proposed solutions; the EGO algorithm spends time refining the response surface model. This reduces the number of solutions that must be evaluated using the expensive cost function. The model then predicts a single solution which has the maximum value of a quantity called expected improvement. This proposed solution is then evaluated using the expensive cost function. If the cost function meets a convergence criterion the proposed solution is selected as the optimum design. If the convergence criterion is not met, the evaluation is used to further refine the model of the response surface and the iteration continues. We now describe the EGO and the GA optimization algorithms.

### **3. METHODS, ASSUMPTIONS AND PROCEDURES**

In this section we discuss the two design optimization algorithms used in this report. First we describe the theory and implementation of the EGO algorithm. We also describe the GA, a very important and well known optimization algorithm. We use both algorithms on the same antenna design problems so we can compare performance. In Section 4 (RESULTS AND DISCUSSION) we present design optimizations for parasitic, super directive arrays; wideband antenna design; and the design of metamaterials. To better appreciate the results and issues associated with these applications we present a description of each algorithm. Since the GA is widely used and better known than EGO, we describe that algorithm in less detail but provide excellent references.

#### **3.1 The Efficient Global Optimization (EGO) Algorithm**

##### **3.1.1 EGO: A Balance between Local and Global Search**

We use the EGO technique presented by Jones, et al, [1] which approximates the response surface using data obtained by evaluating the cost function at a few initial sample data points (an initial data set). The goal is to find the global minimum with as few additional function evaluations as possible. The key to the EGO approach is in the selection of additional data points that exploit the response surface by sampling near a minimum (local search to more accurately find a minimum) while balancing the need to improve the approximation by sampling where the prediction error may be large (global search to ensure that a global minimum is found) [1].

A stochastic process is used to fit the response surface and exploit it for global optimization. The model is stochastic since the results for the next data point are unknown until we actually evaluate the cost function there (also see the last paragraph of Section 3.1.2 and Figure 2 in Section 3.1.5). The stochastic model is calibrated using sample data points and captures how the cost function typically behaves [1].

One behavior might be how much the cost function tends to change as we change a certain input variable.

Using the response surface fit, one can develop figures of merit for selecting new data points. The stochastic approach, with the figures of merit, is a very powerful, effective, and efficient method to select new data points. It automatically achieves a balance between local and global search as described above. The response surface technique for global optimization of expensive black-box functions has two major advantages. First, it often requires the fewest function evaluations of all techniques. Second, it provides a credible stopping criterion for determining convergence. Both advantages are a result of having confidence intervals on the function's value at data points which have not yet been evaluated. The confidence intervals are provided by the statistical model.

The EGO algorithm begins with the selection of an initial set of data points (also called measurements, computer experiments or sample points) which are evaluated using the expensive cost function. The following steps are performed iteratively until convergence is reached: (a) determine correlation parameters using maximum likelihood; (b) select the next data point for cost function evaluation (next data point); and (c) test for algorithm convergence. A flow chart of the algorithm is shown in Figure 1.

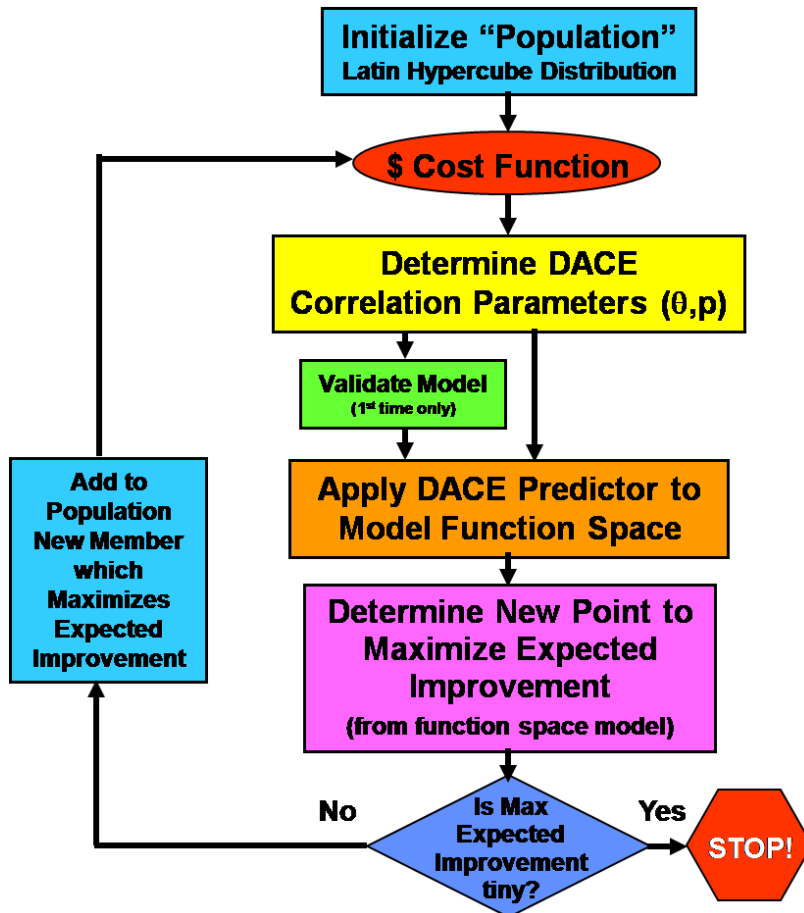


Figure 1. Flowchart of the EGO algorithm.

### 3.1.2 The DACE Stochastic Process Model

The DACE stochastic process model was originally described in a paper by Sacks, et al [10]. We sample the response surface at  $n$  initial data points and represent the surface by  $\mathbf{y}$  which contains the  $n$  samples. Each  $y^{(i)}$  is a function of the  $k$  independent input variables in vector  $\mathbf{x}^{(i)}$ . For the initial  $n$  samples, the  $n$ -vector  $\mathbf{y}$  and  $k$ -vector  $\mathbf{x}^{(i)}$  are:

$$\mathbf{y} = [y^{(1)} \quad y^{(2)} \quad \dots \quad y^{(n)}]_{n \times 1} \quad (1a)$$

$$\mathbf{x}^{(i)} = [x_1^{(i)} \quad x_2^{(i)} \quad \dots \quad x_k^{(i)}]_{k \times 1} \quad i = 1, 2 \dots n \quad (1b)$$

The input data  $\mathbf{x}$  is just an  $n \times k$  matrix of all the  $\mathbf{x}^{(i)}$  ( $i=1,2,\dots,n$ ) stacked in a column. If the problem is one dimensional, i.e.  $k=1$ , we have only a single input variable,  $x$ , sampled at  $n$  points. The initial number of data points is typically chosen to be equal to  $11k-1$  or approximately 10 times the dimensionality [1]. For multi-dimensional problems, space-filling techniques such as Latin hypercube designs [11] can be used to obtain initial sample points to start the algorithm.

The DACE stochastic process model is very different from regression analysis where error terms are assumed to be normally distributed and independent from data point to data point. The primary emphasis in regression is on finding regressors, i.e. the coefficients of assumed linear (or higher order polynomial or functional) terms used to approximate the function. The primary emphasis in DACE modeling is on estimating correlation parameters which describe how a function typically behaves [1, 10]. We will show that correlation parameters, along with a DACE predictor, provide much more than typical behavior, i.e. we can approximate the response surface and accurately find the global minimum. The DACE model assumes that the regression term,  $\mu$ , is a constant as shown below:

$$y(\mathbf{x}^{(i)}) = y^{(i)} = \mu + \varepsilon(\mathbf{x}^{(i)}). \quad i = 1, 2, \dots, n \quad (2)$$

The measurement error term,  $\varepsilon(\mathbf{x}^{(i)})$ , is assumed to be normally distributed with zero mean and standard deviation  $\zeta$ , i.e.  $N(0, \sigma^2)$ ; however, in the DACE model the error terms are not assumed to be independent but correlated from one measurement (data point) to another. The correlation is assumed to be high when data points  $\mathbf{x}^{(i)}$  and  $\mathbf{x}^{(j)}$  are close and lower when they are farther apart. The correlation between errors is therefore related to the distance between corresponding data points. We do not use the Euclidean distance but, instead, use a weighted distance formula [1] as follows:

$$d(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) = \sum_{h=1}^k \theta_h |x_h^{(i)} - x_h^{(j)}|^{p_h} \quad (3)$$

The correlation between data points at  $\mathbf{x}^{(i)}$  and  $\mathbf{x}^{(j)}$  is calculated from:

$$\text{Correlation}_{ij} = \exp[ - d(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) ] \quad (4)$$

The correlation parameter  $\theta_h$  in Equation 3 measures the importance, or activity, of input variable  $x_h^{(i)}$ . The correlation parameter  $p_h$  in the exponent is related to the smoothness of the function in input variable direction  $h$ , where  $p_h = 2$  corresponds to smooth functions while values of  $p_h$  near one correspond to less smooth functions [1, 12]. We typically use  $p_h = 2$  for our antenna design optimization problems.

Equations 2, 3 and 4 represent the DACE stochastic process model. It is a stochastic model since the error term  $\varepsilon(\mathbf{x}^{(i)})$  is a stochastic process, i.e., it is one of a set of correlated random variables indexed by the  $k$ -dimensional space  $\mathbf{x}$ . The stochastic nature of the model is discussed in detail in Section 3.1.5 and is illustrated in Figure 2.

### 3.1.3 Estimation of Correlation Parameters

The first step in the algorithm is to estimate correlation parameters  $\theta_h$  and  $p_h$  for  $h = 1, 2, \dots, k$  [1]. The parameters are estimated by selecting values of  $\theta_h$  and  $p_h$  that maximize the likelihood of the sample and are therefore „tuned“ to the  $n$  data points. Let  $\mathbf{y}$  be the column  $n$ -vector of measured (i.e. evaluated) function values;  $\mathbf{1}$  a column  $n$ -vector of ones; and  $\mathbf{R}_{n \times n}$  be the error correlation matrix whose  $(i, j)$  entry is given by the correlation in Equation 4. The sample likelihood function is given by:

$$L(\mu, \sigma^2) = \frac{1}{(2\pi)^{n/2} (\sigma^2)^{n/2} |\det(\mathbf{R})|^{1/2}} \exp \left[ - \frac{(\mathbf{y} - \mathbf{1}\mu)' \mathbf{R}^{-1} (\mathbf{y} - \mathbf{1}\mu)}{2\sigma^2} \right] \quad (5)$$

We use singular value decomposition (svd) to find the inverse of  $\mathbf{R}$ .

The dependence of the sample likelihood function on the correlation parameters comes from the error correlation matrix  $\mathbf{R}$ . If we had values for the correlation parameters we could calculate  $\mathbf{R}$  and find closed form expressions for  $\mu$  and  $\sigma$ . The maximum likelihood estimates (which maximize the sample likelihood function) are calculated as follows [1]:

$$\hat{\mu} = \frac{(\mathbf{1}'\mathbf{R}^{-1}\mathbf{y})}{(\mathbf{1}'\mathbf{R}^{-1}\mathbf{1})} \quad (6)$$

and

$$\hat{\sigma}^2 = \frac{(\mathbf{y} - \mathbf{1}\hat{\mu})'\mathbf{R}^{-1}(\mathbf{y} - \mathbf{1}\hat{\mu})}{n} \quad (7)$$

Using these values for  $\mu$  and  $\sigma$  in Equation 5 for the sample likelihood function gives the concentrated likelihood function (CLF), which depends on correlation parameters  $\theta_h$  and  $p_h$ . The CLF is then maximized with respect to the correlation parameters  $\theta_h$  and  $p_h$  [1, 10]. We actually maximize the log of the CLF. The values of  $\theta_h$  and  $p_h$  which maximize the CLF are then used to obtain an estimate for  $\mathbf{R}$  which is used in Equations 6 and 7 to give final estimates for  $\mu$  and  $\sigma$ . We maximize the CLF by using either a Nelder-Mead simplex technique (implemented in MATLAB function *fminsearch*); by an exhaustive search across a reasonable span of  $\theta_h$ ; or by using a screening algorithm technique described in Section 4.2.5.

### 3.1.4 Model Validation and the DACE Predictor

The second step in implementing the EGO algorithm is to validate the model using a technique called cross validation [1]. This involves leaving out one of the initial data points, calculating a new  $\mathbf{R}$ , and then estimating a predicted value of the function at that data point to see how well the model predicts the actual value. The value of the function is known at that data point since it has been sampled there. This is done  $n$  times since each data point can be left out once. We only perform the cross-validation procedure one time and we use the initial data set.

We now change notation slightly and indicate a single sample data point by the  $k$ -dimensional vector  $\mathbf{x}$  (instead of  $\mathbf{x}^{(i)}$ ) and an unknown data point as  $\mathbf{x}^*$ . The DACE predictor, which is the best linear unbiased predictor (BLUP) of  $y(\mathbf{x}^*)$ , where  $\mathbf{x}^*$  is a new data point but not one of the sampled data points, is given by [1, 10]:

$$\hat{y}(\mathbf{x}^*) = \hat{y}^{(*)} = \hat{\mu} + \mathbf{r}'\mathbf{R}^{-1}(\mathbf{y} - \mathbf{1}\hat{\mu}), \quad (\text{the DACE predictor}) \quad (8)$$

where  $\mathbf{r}$  is an  $n \times 1$  column vector of correlations between  $\mathbf{x}^*$  and all  $n$  data points. In Equation 4 just substitute  $\mathbf{x}^*$  for  $\mathbf{x}^{(j)}$  for all  $j$ . The DACE predictor interpolates the data, i.e. it gives the actual values of  $\mathbf{y}$  in Equation 1a at the sample data points and at  $\mathbf{x}^*$  it yields a predicted value.

Interestingly, the DACE predictor is a radial basis function approximation. Suppose the problem is one dimensional ( $k=1$ ) and that there are two data points,  $x_1$  and  $x_2$ . The term  $\mathbf{R}^{-1}(\mathbf{y} - \mathbf{1}\hat{\mu})$  is just a two element column vector  $\begin{bmatrix} c_1 \\ c_2 \end{bmatrix}$  and, at  $x = x^*$ , the DACE predictor becomes (for  $p_h = 2$ ):

$$\hat{y}(x^*) = \hat{\mu} + c_1 \exp[-\theta|x_1 - x^*|^2] + c_2 \exp[-\theta|x_2 - x^*|^2] \quad (9)$$

We now introduce a term called the standard error of prediction. The square of the standard error of prediction, as described by Jones, et al [1], is given below:

$$s^2(\mathbf{x}^*) = \sigma^2 \left[ 1 - \mathbf{r}'\mathbf{R}^{-1}\mathbf{r} + \frac{(\mathbf{1} - \mathbf{1}'\mathbf{R}^{-1}\mathbf{r})^2}{\mathbf{1}'\mathbf{R}^{-1}\mathbf{1}} \right], \quad (10)$$

where the prediction error is reduced by the second term in the brackets due to the fact that  $\mathbf{x}^*$  is correlated with the sample (data) points. The increase in prediction error due to the third term in the brackets represents the fact that we are not using an exact value for  $\mu$  but are estimating it from the data. Note that  $s = 0$  (no uncertainty) at the sample data points. The predicted value  $\hat{y}(\mathbf{x}^*)$ , the standard error of prediction and the known value of the function at the sample points,  $\mathbf{x}$ , can be used to cross-validate the model as described in [1] by calculating the number of standard errors that the predicted value of  $\hat{y}(\mathbf{x}^*)$  differs from the actual sample value with  $\mathbf{x}^*$  located at the  $n$  sample data points. In the cross validation process described above, the number of standard errors,  $s(\mathbf{x})$ , that  $|\hat{y}(\mathbf{x}^*)|$  can vary from the actual sample value  $y(\mathbf{x}^*)$  is required to be less than three.

### 3.1.5 Selecting the Next Design Point

The third, and key, step in implementing the EGO algorithm [1] is the selection of the next data point for the evaluation of the objective function. Jones, et al, [1] introduce a figure of merit called expected improvement which automatically balances local and global search and is the heart of the EGO algorithm. A new random variable  $Y(\mathbf{x})$  is defined which is normally distributed with mean  $\hat{y}(\mathbf{x})$  and variance  $s^2(\mathbf{x})$ .

$Y(\mathbf{x}^*)$  models the uncertainty in the function's value at an arbitrary point  $\mathbf{x}^*$ , as illustrated in Figure 2. The model is stochastic since we do not know the outcome of the experiment an arbitrary design point. Let  $f_{\min} = \min[y^{(1)} \ y^{(2)} \ \dots \ y^{(n)}]$  be the current best cost function value. Formally, the improvement at arbitrary point  $\mathbf{x}$  (for simplicity we use  $\mathbf{x}$  instead of  $\mathbf{x}^*$  from now on) is given by [1]:

$$I(\mathbf{x}) = \max\{f_{\min} - Y(\mathbf{x}), 0\} \quad (11)$$

Since  $I(\mathbf{x})$  is a random variable (recall that  $Y(\mathbf{x})$  is a random variable), we take the expected value of  $I(\mathbf{x})$  and call it the expected improvement. The expected improvement can be expressed in closed form [1]:

$$E[I(\mathbf{x})] = (f_{\min} - \hat{y})\Phi\left[\frac{f_{\min} - \hat{y}}{s}\right] + s\phi\left[\frac{f_{\min} - \hat{y}}{s}\right], \quad (12)$$

where  $\phi[z]$  is the normal probability density function and  $\Phi[z]$  is the normal probability distribution function. The expression inside the bracket is called the normalized improvement. Since we have a closed-form expression, we use Equations 8 and 10 for  $\hat{y}(\mathbf{x})$  and  $s(\mathbf{x})$  to evaluate  $E[I(\mathbf{x})]$  at a large number of points (excluding current sample data points since  $s = 0$  there) and find the value of  $\mathbf{x}$  where the expected improvement is maximized. We evaluate the objective function at that  $\mathbf{x}$  to obtain a new sample data point (a new design point).

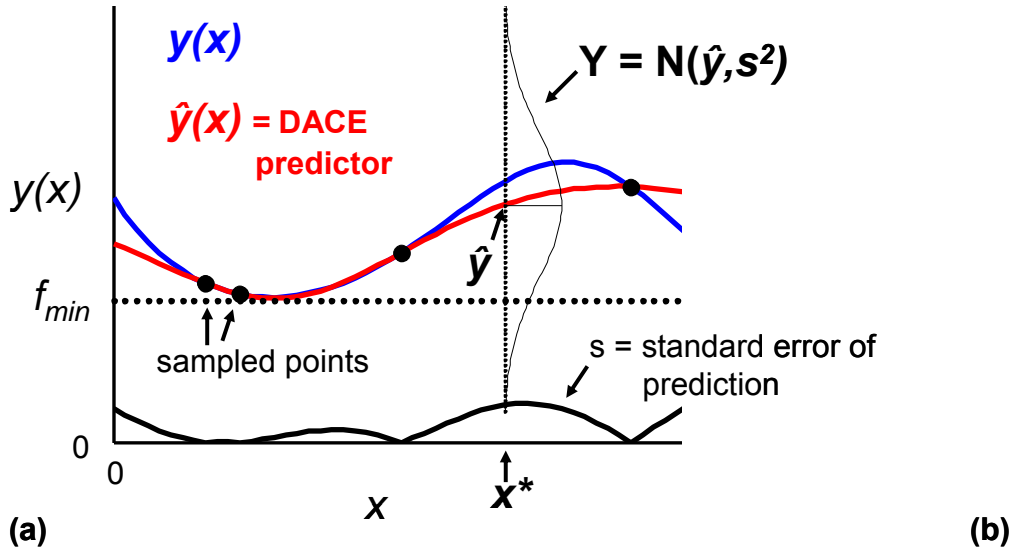


Figure 2. Uncertainty in the value of the cost function  $y(x)$  at the point  $x^*$ : (a) Random variable  $Y(x^*)$  is normally distributed with mean  $\hat{y}(x^*)$  and variance  $s^2$  (b) Detail showing the improvement,  $I(x^*)$ , represented by the shaded area at the bottom where  $Y < f_{\min}$ . Improvement is a random variable since  $Y$  is a random variable which is assumed to be normally distributed with mean  $\hat{y}(x)$  and variance  $s^2(x)$ .

### 3.1.6 Convergence Criteria

The fourth and final step of the EGO algorithm is to implement a stopping rule (or convergence criterion). The expected improvement provides a simple and effective criterion. If the absolute value of the expected improvement,  $|E[I(\mathbf{x})]|$ , at the selected sample data point is less than 1% of  $f_{min}$  (the best current function value), then stop. One could choose a factor less than 1% and we have experimented with decreasing this stopping criterion to 0.005% to obtain better accuracy. However, this requires additional expensive cost function calls.

An alternate stopping criterion is the total number of iterations of the algorithm. Using the total numbers of iterations as a secondary stopping criterion prevents the algorithm from running extensively, while still allowing it to drill deeper into the minima than would otherwise occur with a larger expected improvement stopping criterion.

If the criterion is not met we add the selected point as a new data point and increase  $n$  by one, i.e., now there are  $n+1$  data points. The algorithm proceeds by returning to step one and estimating new values for the correlation parameters using the new data set, which includes all previously evaluated data points plus the new point  $\mathbf{x}$ . The algorithm continues until the convergence criterion is met.

Evolutionary algorithm (EA) techniques such as EGO and GAs can also be terminated based on performance criteria, i.e. where the current value of the cost function satisfies certain design specifications. This is perhaps more realistic and practical and could replace artificial (and often arbitrary) convergence criteria [13]. We address performance-based convergence criteria in Section 4.5.

### 3.1.7 A Simple One-Dimensional Example Using EGO

Perhaps the easiest way to understand the operation of the EGO algorithm is with a simple one-dimensional optimization problem. We seek the global maximum of the function  $\sin(x)$  in the input space  $0 \leq x \leq \pi$ . The single maximum is equal to 1 at  $x = \pi/2$ . Since our algorithms are minimum seeking algorithms we seek the minimum of the function  $-\sin(x)$  in the same input space,  $0 \leq x \leq \pi$ . The single minimum is equal to -1 at  $x = \pi/2$ , i.e.  $x = 1.5708$  radians. The first step is to select an initial data set. The number of initial samples is usually about 10 times the dimensionality; however, for this very simple problem we select only three points in the input space using a random number generator. The three initial sample points selected are: 1.2322, 2.0592 and 0.5378 (diamonds in Figure 3a). Evaluating  $-\sin(x)$  at the sample points we obtain  $f_{min} = -0.9432$ . We can visualize the iterative progression of the algorithm if we plot: (1) the current design points; (2) the DACE predictor obtained from the current set of points; and (3) the expected improvement (EI). All three are plotted over the input space  $0 \leq x \leq \pi$  in Figures 3, 4 and 5 below.

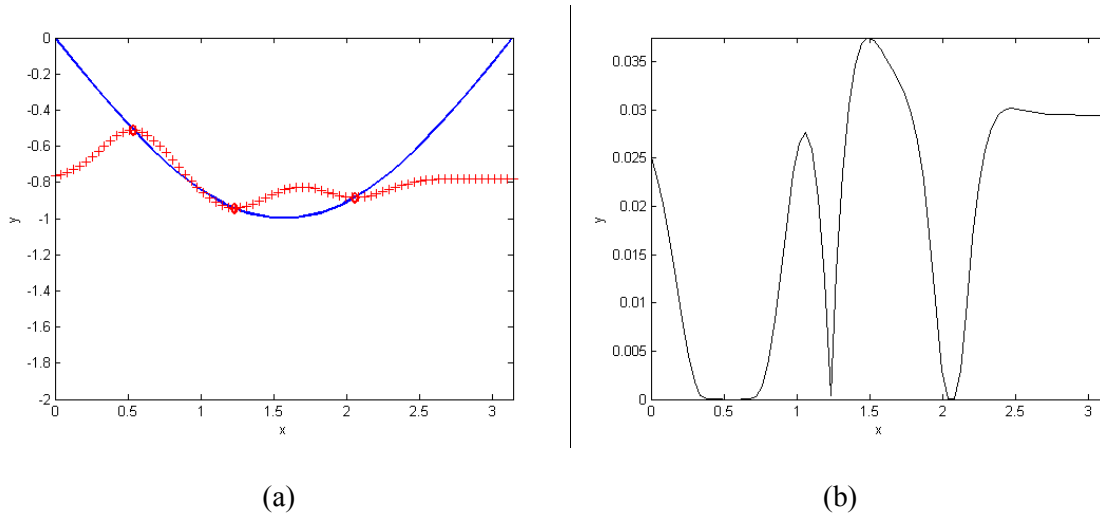


Figure 3. Iteration One (a) The solid curve is the cost function given by:  $-\sin(x)$  over the input space  $0 \leq x \leq \pi$ . The three dark red diamonds are the initial data set. The red curve is the DACE predictor from EGO obtained using the three initial samples. Note that the DACE predictor interpolates the data points. (b) The EI from EGO obtained using the three samples and the DACE predictor in (a). It is maximized at  $x = 1.4859$  therefore the new design point is selected at  $x = 1.4859$ .

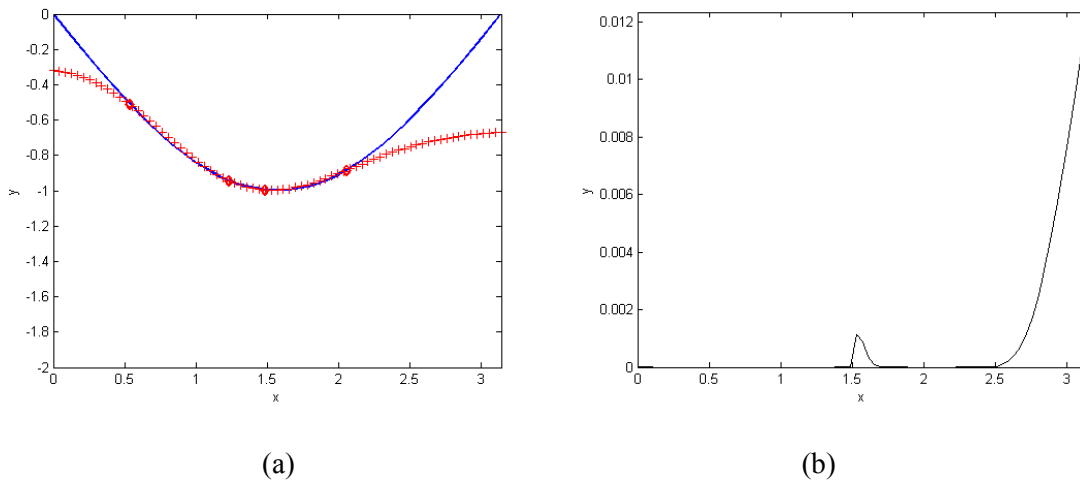


Figure 4. Iteration Two (a) The four dark red diamonds are the initial data set plus the new design point selected from Figure 3b. The red curve is the DACE predictor from EGO obtained using the four data samples. (b) The EI from EGO obtained using the four samples and the DACE predictor in (a). It is maximized at  $x = 3.1416$ , i.e.  $\pi$ , therefore the new design point is selected at  $x = \pi$ .

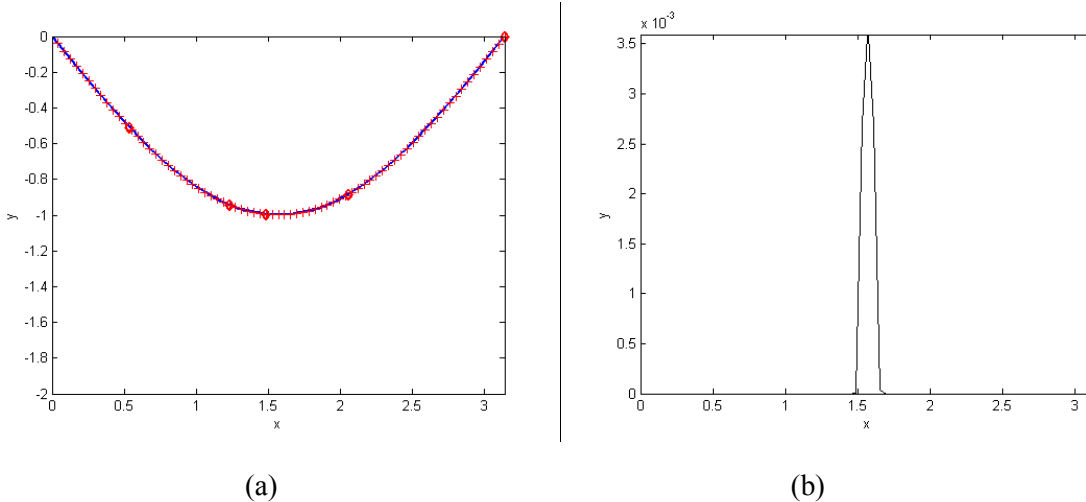


Figure 5. Iteration Three (a) The five dark red diamonds are the current sample points. The red curve is the DACE predictor from EGO obtained using the five data samples. (b) The EI from EGO obtained using the five samples and the DACE predictor in (a). It is maximized at  $x = 1.5708$  therefore the new design point is selected at  $x = 1.5708$ , which is the answer at convergence and is equal to the exact value of the global minimum.

Note that in the second iteration (Figure 4) the algorithm actually selects the next evaluation point at a large distance from the global minimum as part of a global search strategy. At the next iteration (Figure 5) the DACE predictor fits the response surface almost exactly. The fit changes very little if we add the final selected point at  $x = 1.5708$ . Therefore we do not show the curve with six sample points. A summary of the three iterations is shown in Table 1.

Table 1. Summary of the EGO run for the simple 1D problem. Max (EI) is the maximum value of the expected improvement (EI) in the input space. The point where EI is maximized is selected as the next design point. Note that Max (EI) is less than 1% of  $\text{abs}(f_{\min})$ , or 0.01, at the third iteration; therefore convergence is declared at iteration number three.

Iteration	Selected design point	$f_{\min}$	Max (EI)
1	1.4859	-0.9964	0.0375
2	3.1416	-0.9964	0.0123
3	1.5708	-1.0000	0.0036

In Section 3.1.4 we briefly discussed model validation. This procedure is performed once using the initial data set. We can illustrate the concept on our simple 1D problem. The basic idea is to leave out a data point and use the others (two in this case since we have only three initial data points) to predict the value of the cost function at the point that was left out. We show a summary of the results in Table 2. The difference between the actual value  $y(x)$  and the predicted value  $y_p(x)$  divided by the standard error of prediction,  $s(x)$ , should be in the range from -3 to +3, which it is in all three cases. For this very simple case the value of  $y_p(x)$  in all three cases is simply equal to  $\mu$ , the average value calculated from the DACE parameters.

Table 2. Summary of the Model Validation Results

Point Left Out	$y(x)$	$y_p(x)$	$s$	$(y-y_p)/s$
1.2322	-0.9432	-0.7795	0.2334	-0.7014
2.0592	-0.8831	-0.7795	0.2334	-0.4437
0.5378	-0.5122	-0.7795	0.2334	1.1451

### 3.2 Genetic Algorithm (GA) Optimization

The GA is a global search technique which encodes parameter values as chromosomes. The strategy is “survival of the fittest” where more fit chromosomes have lower cost. The cost function is used to determine the cost of each chromosome. The parameters in our problems are continuous valued so we use a continuous-parameter GA (CPGA) [14, 15]. This means that the computer uses its internal precision and associated round-off error to define the accuracy of the parameters rather than using a binary representation. For dimensionality  $k$ , each chromosome contains  $k$  continuous parameters, or genes.

We use the Latin hypercube technique to establish an initial population of  $N_{ipop}$  chromosomes which samples the function space [1, 2, 11 and 40]. There is a trade-off between the number of initial samples and the number of generations required for the algorithm to converge. A larger initial population means that the function space will be sampled better initially and the GA should have a better start at finding a global minimum.

Using the initial population, we rank the  $N_{ipop}$  initial chromosomes from lower cost to higher cost using the expensive cost function. The bottom half (the poorer performers with higher cost) are discarded leaving  $N_{pop}=N_{ipop}/2$  chromosomes for succeeding generations [14]. The rationale for a larger initial population is simple. Once the cost (response) surface has been adequately sampled (explored) the algorithm can work with a subset of the best samples to exploit the response surface. However, a larger  $N_{ipop}$  means more evaluations using the expensive cost function. For each successive generation, we take the top  $N_{pop}/2$  chromosomes (the top half performers with the lower cost) as a mating pool and discard the bottom  $N_{pop}/2$  chromosomes (the bottom half poorer performers with higher cost). The discarded chromosomes are replaced by  $N_{pop}/2$  children which are the offspring of the parents in the mating pool.

#### 3.2.1 The Simple Textbook GA

The first step in the simple, textbook GA [14] pairs the  $N_{pop}/2$  parents in the mating pool. Each pair of parents is mated to yield a pair of children. The parents plus the children are the  $N_{pop}$  chromosomes for the next generation. We use a technique called weighted random pairing to pair up the parents. This procedure is well described by Haupt and Haupt [14, page 38].

Each pair of parents is then mated, i.e. the genes (parameters) of the two parents are swapped and/or combined to form the two offspring (children). Hopefully, some of the children will be better performers (have lower cost) than the parents. The simplest methods randomly choose one or more points within the two chromosomes as crossover points. The parameters within these points are then swapped between the

two parents [14]. For a design problem with two parameters, e.g.  $p_1$  and  $p_2$ , we represent the two parents as:

$$\text{Parent}_1 = [p_{m1} \ p_{m2}] \quad \text{“the Mom”} \quad (13)$$

$$\text{Parent}_2 = [p_{d1} \ p_{d2}] \quad \text{“the Dad”} \quad (14)$$

For two parameters, the crossover point can only be at parameter 1 or 2. For example if the crossover point is randomly selected to be at  $p_1$ , the two children are:

$$\text{Offspring}_1 = [p_{m1} \ p_{d2}] \quad \text{“the first child”} \quad (15)$$

$$\text{Offspring}_2 = [p_{d1} \ p_{m2}] \quad \text{“the second child”} \quad (16)$$

The second parameters have been swapped. Parameters are only swapped vertically to avoid mixing units. There is one critical problem in using the simple point crossover method in continuous parameter GAs: no new information is introduced. The continuous parameter value that was randomly selected in the initial population is propagated to the next generation, only in different combinations, i.e. in different chromosomes. To remedy this problem a *blending method* is used to combine parameter values from the two parents into new parameter values in the offspring. A random crossover point is chosen as in the discussion above. Assuming that the crossover point is at  $p_1$ , the parameters at the crossover point are *blended* to form two new parameter values:

$$p_{\text{new1}} = p_{m1} - \beta [p_{m1} - p_{d1}] \quad (17)$$

$$p_{\text{new2}} = p_{d1} + \beta [p_{m1} - p_{d1}], \quad (18)$$

where  $\beta$  is a random variable between 0 and 1. Finally, we complete the crossover, i.e. the second parameter is swapped:

$$\text{Offspring}_1 = [p_{\text{new1}} \ p_{d2}] \quad \text{“the first child”} \quad (19)$$

$$\text{Offspring}_2 = [p_{\text{new2}} \ p_{m2}] \quad \text{“the second child”} \quad (20)$$

The third (and final) step in the GA is a mutation operation which forces the algorithm to explore new areas of the response surface and is key in assuring that the GA will seek a global minimum rather than a local one. It is an effective way to kick the algorithm out of local minima. After the mating process we

randomly select a fraction of the total number of genes in the parents plus the children and then mutate by changing those genes to a random values within appropriate ranges. We do not mutate the number one ranked chromosome.

For two parameters in each chromosome, the number of genes mutated in each generation is:

$$N_{\text{mut}} = (2\delta/100) (N_{\text{pop}} - 1), \quad (21)$$

where the mutation rate,  $\delta$ , is percentage. In [14], mutation rates between 1 and 20% are recommended; however, for our problem we have found higher mutation rates to be necessary. If both genes are selected for mutation within the chromosome we mutate both by selecting a random value of each parameter within the range of that parameter. For example, for  $p_1$  we would select a new value for that parameter using

$$p_{1\text{mut}} = (\text{High} - \text{Low}) \textit{rand} + \text{Low}, \quad (22)$$

where *rand* is a random number between 0 and 1. This gives a random value between the Low and High values of  $p_1$ .

The GA goes through successive generations producing better and better (lower and lower cost) chromosomes. Unlike EGO, which has a natural stopping criterion, there is no natural stopping criterion for a GA. In practice then, when do you stop and declare convergence? There is no good answer [14]. We discuss a performance-based convergence criterion in Section 4.5.4. For the textbook GA, we implemented four criteria. If any one of the four conditions is satisfied we stop the GA and declare that the number one chromosome is the best design. The four criteria are:

1. Limit of 100 total number of generations.
2. No change in cost associated with the number one chromosome after 1/5 of the limit on the total number of generations, i.e. 20 generations (*stagnation*).
3. 7/8 of the genes in the mating pool are identical (*lack of diversity*).
4. Number of cost function calls exceeds 500 (the GA has become entirely too expensive to use at this point and becomes irrelevant).

### 3.2.2 An Enhanced GA

In addition to the textbook GA, we developed an enhanced robust GA (ERGA) by incorporating a number of additional techniques. These techniques were selected to promote genetic diversity and avoid premature convergence. The algorithm is robust since it consistently finds the global minimum once properly tailored to the function space [16].

In particular, the techniques in ERGA include:

- Normalizing the input range of each parameter to [0 1]
- Varying the mutation rate (increasing or decreasing) as the algorithm progresses

- Varying the mutation type, from the replacement of a gene with a totally random value as described for the textbook GA, to offsetting the current value of the gene by a random amount
- Varying the range of the random amount (the sigma) of the offset mutation values as the algorithm progresses (usually from higher to lower)
- Setting a minimum solution granularity, i.e. a point at which two proposed chromosomes are declared identical
- Maintaining population diversity by never allowing duplicate chromosomes within the population (within the specified solution granularity)
- Employing tournament selection (i.e. comparing two random parents and the best one is selected), rather than the textbook weighted random cost parent selection
- Not mutating and/or re-simulating a fixed percentage of kept parents (not just the number one chromosome), but maintaining them as competitive members for inclusion in future generations (multiple elitism rather than single elitism)

Depending on the characteristics of the function space, we may wish to have more genetic inheritance during portions of the algorithm and more mutation at other times. We may also wish to change the type of mutation from totally random value replacement (to accomplish global exploration) to offset values around the current parameters (to accomplish local search) as the algorithm progresses. For example, in the initial generations, we want to balance the genetic inheritance associated with recombination with a certain amount of random global search (using either gene replacement with new random values or offset mutation with high sigma values) to ensure a wide amount of global exploration. However, unless the function space is very “spiky”, we expect to have one or more chromosomes within the bowl of the global minima after a low number of generations. At this point, we want less global exploration and more local search to find the bottom of the bowl. This can be accomplished by either reducing high mutation rates (if there are good genetic “building blocks” or schema within our chromosomes that can combine effectively) or alternatively by increasing to a higher amount of mutation with a tight sigma offset (if our chromosomes do not have well correlated schema, which turned out to be the case for the parasitic super directive array (PSA) optimization). In either case, for a relatively smooth function space, we want to accomplish a rapid local exploration around the best performers during the algorithm “end-game” to find the global minimum.

The technique of defining a minimum solution granularity prevents the algorithm from spending expensive function evaluations computing essentially the same result for the same input point. While the CPGA uses its internal precision and associated round-off error to define the accuracy of the parameters, this is a mixed blessing, as this high amount of numerical precision can create chromosomes which are numerically “different” but which essentially represent the same function space point. This high amount of numerical precision can lead to a condition where the population appears to be numerically diverse, but essentially all chromosomes represent the same function space point, as the numerical variations only exist within in the high-precision gene digits.

Besides eliminating wasted function calls, defining a minimum solution granularity allows the algorithm to determine whether a point already exists in the population (within that predetermined solution granularity). In the ERGA, we do not duplicate members of the population (either preexisting parents or newly created children) but rather check to determine that new potential members (after recombination and mutation) are unique prior to their addition into the population. Since a diverse population is automatically maintained by the ERGA algorithm, population diversity is not a useful stopping criterion for terminating the ERGA. Only the following stopping criteria are used: total number of generations; total number of cost function calls; and stagnation.

Just as variations in population size and mutation rate affect the ability of the textbook CPGA to converge to good solutions, variations in the parameters in the ERGA affect its ability to converge to the global minimum. A certain amount of trial-and-error exploration was performed to find a good set of parameters for the function space being optimized.

## 4. RESULTS AND DISCUSSIONS

In this section we discuss antenna design optimization applications using EGO and the GA. We also describe two areas of research with the goal of improving the performance of design optimization algorithms.

The first antenna design is a parasitic super directive array (PSA) where we compare EGO first with a classic optimization technique called Nelder-Mead downhill simplex and then with GA optimization. The second antenna design optimization problem is a wideband antenna element backed by: (1) a metamaterial ground plane and (2) a perfect electric conductor (PEC) ground plane. The antenna with a PEC ground plane was fabricated and we present test results and compare with predicted results. The third antenna design is an infinite, periodic array of fragmented patch antenna elements which is also a wideband antenna.

We also investigate possible improvements to the EGO algorithm. The first is in the “endgame” where we consider techniques to make the algorithm more accurate in its final estimate of the global minimum. Finally, we consider starting the algorithm, i.e. techniques for selecting initial data sets. This can result in more efficient algorithms.

### 4.1 Parasitic Super Directive Array (PSA) Design Optimization

Super directivity in linear periodic arrays is a phenomenon where, as the spacing between elements decreases, the end fire directivity of the array may approach  $N^2$  where  $N$  is the number of elements in the array. However, achieving super directivity requires driving the individual elements with precise magnitudes and phases to achieve a desired relationship between the currents in the elements. Very small two-element super directive arrays have been developed using monopole and electrically-small elements [15, 16]. While achieving the required current relationships has been shown to be possible, it is impractical and expensive to accomplish in practice.

O’Donnell, et al [17] demonstrated an alternative method for obtaining almost equivalent super directivity in two-element arrays by feeding only one element and shorting the second “parasitic” element as shown in Figure 6a. In Figure 6b, note that the super directivity of this *parasitic array* (blue curve) approaches that of the fully-driven super directive array (red curve) for a limited range of element separations. Outside of this range, the directivity drops off significantly. Upon further investigation, it was discovered that increased parasitic super directivity was possible at many other element separations outside this limited range (green curve), but only by shifting the operating frequency of the array. The reasons for this shift stem from a tradeoff between creating equal current distributions throughout both antenna elements while remaining close to the resonant frequency of a single element. A full discussion of this tradeoff can be found in [17]. The deviation of the super directive frequencies to obtain optimal super directive gain was shown to vary as a function of the element separation and other properties, such as the antenna’s

electrical height and quality factor ( $Q$ ). In general, there is no closed-form solution for determining either the best separation or frequency shift. For the optimized results presented in [17, 18], an exhaustive search of the relevant function space of possible element spacing and frequency shifts was performed to determine the best possible super directive end fire gain. This was feasible because the antennas consisted of thin-wire configurations in free space on an infinite ground plane, which could be quickly simulated using the Numerical Electromagnetics Code 4 (NEC4) [19], a method of moments code. Further research into PSAs immersed in various dielectric media has required more sophisticated electromagnetic simulation software, such as the High Frequency Structure Simulator HFSS [20] which employs a 3-D, full-wave, finite element method solver. These computations take significantly longer and an exhaustive search of the parasitic array trade-off space between separation and frequency shift is a non-trivial matter.

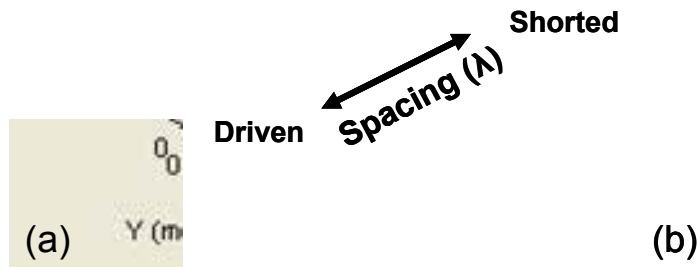


Figure 6. (a) Two electrically small antennas in a parasitic super directive array configuration. (b) A comparison of the driven super directive gain achievable with these elements with the array at element resonance (red curve); the parasitic gain if the array is kept at element resonances (i.e. not frequency shifted) (blue curve); and the parasitic gain possible if the array frequency is optimized for each separation (green curve).

We employed the EGO algorithm described in Section 3 to optimize the directivity of a two-element PSA. This is a design problem with known results, i.e. the optimization of both the separation and the frequency of a thin-wire parasitic array in free-space on an infinite ground plane. This configuration has been extensively studied [17] and we know how well these arrays perform when various element configurations are used. Our goal was to test the EGO algorithm on this limited parameter (two dimensional) problem to determine if the algorithm could find the optimal element separation and frequency shift and whether it could find this optimum each time.

### 4.1.1 The Test131 PSA and Associated Function Space

In Figure 7a we show the **Test131** antenna, which is a moderately small (height =  $0.146\lambda$  at resonance) planar element with an antenna quality factor  $Q = 4$ . Since this element has a wide bandwidth, it can accommodate significant frequency shift, which allows it to be closely spaced in a PSA. The full tradeoff space between element separation and operating frequency is shown in Figure 7b. There are two good operating frequencies and separations for this array. At a separation of 8.9 mm and a frequency of 849 MHz, the array produces a reflector lobe having a directivity of 10.262 dB. This is the best possible directivity and is shown by the red circle in Figure 7b. A less powerful director lobe (in the other end fire direction) occurs at a separation of 14mm and frequency 834 MHz, with a maximum gain of 10.154 dB. The director lobe is indicated by the green circle.



Figure 7 (a) Configuration of a Test131 parasitic array, with elements separated by 20 mm. (b) The full simulated function (input) space showing directivity contours for element separations ranging from 5mm to 60mm and frequencies from 800 to 900MHz.

### 4.1.2 EGO Algorithm Parameters Used for Testing

We ran the EGO algorithm on this two-dimensional design optimization problem to determine the best spacing and frequency to optimize directivity. We followed the  $n=11k-1$  criterion discussed in Section 3.1.2 and chose 21 sample points selected in a Latin hypercube distribution. As indicated in Section 3.1.6, we employed both maximum expected improvement and total number of iterations as stopping criteria. At this point in the research we did not use performance-based convergence.

We did not implement the branch-and-bound method suggested by Jones, et al, in [1] to select the points used to test the DACE predictor. Instead, we used a two-dimensional grid of test points, at 0.5mm separations along the x-axis (element separation), and 1MHz separations along the y-axis (operating frequency) [29]. The DACE predictor was applied at all these grid points, with the expected improvement calculated at each grid point.

A caution here is that, unlike the branch-and-bound method, this method only tests points along an arbitrary test grid. If the function optimum is not on this grid, the nearest point to it will be determined to be the optimum. To mitigate this error, for each iteration we added a random offset to the grid ( $\leq$  one grid step in each dimension), to move the grid to slightly different testing points. For this initial research that appears to be sufficient since we obtained good results.

Our cost function, which is antenna directivity, is usually represented in decibels (dB), which is equal to  $10 \log_{10}$  of the linear directivity. We were initially uncertain whether to use dB for our directivity function or whether the non-transformed linear values would work better. After testing the EGO algorithm on both functions, we determined that either is acceptable for this optimization as both yielded approximately the same accuracy. Our results for directivity presented in the tables below are in dB.

### 4.1.3 Results from the EGO Design Optimization

Table 3 shows a sample of 15 EGO runs on the Test131 parasitic array. Each run is started with a different set of 21 random points in the function space. For these runs, we used directivity in dB, a maximum expected value stopping criterion of 0.1% of  $f_{\min}$ , and a maximum number of iterations of 50. Since our number of initial sample points determined by the Latin hypercube was 21, this meant that up to 71 evaluations of the expensive cost function would be allowed. (If we had evaluated the function directly at each point on our test grid, this would have presented 101 frequency points (from 800–900MHz in 1 MHz increments) times 106 separation points (7.5mm to 60mm at .5mm increments), or 10,706 expensive function evaluations. Our initial 21 sample points generated with the Latin hypercube distribution represent a sample of only 0.19% of the function space (using the coarseness of the test grid increments). Using a stopping criterion of 71 cost function evaluations, we will have only sampled 71/10706 or 0.66% of the function space. In Table 3 below we show the results from EGO after 71 function evaluations. The minimum is negative since we are minimizing the negative of the directivity to find the global maximum in the directivity (or gain).

The data in Table 3 shows that the average predicted value for the maximum directivity of the Test131 parasitic super directive array is 10.233 dB. This compares quite well with the actual 10.262 dB directivity that is possible with this array. In fact, Table 3 shows that the optimum solution was obtained once in run number 9. The average number of expensive function evaluations was 69. Both the maximum expected value and the total number of iterations were used as stopping criteria. The lowest predicted directivity was 10.194 dB. However, even this value was higher than the director lobe maximum of 10.154 dB, indicating that even the poorest of solutions was still in the correct minima “bowl” - but it just didn’t drill down deep enough. This represents a case where we would have wished the algorithm to proceed further, but the maximum expected value and/or the total number of allowed evaluations indicated convergence. One could add a local search method to assist in the endgame, especially if the algorithm shows convergence after a relatively small number of iterations. We discuss such techniques in Section 4.4.

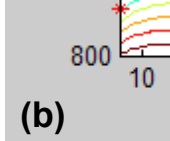
Figure 8 shows one example of the initial sampled points generated by the Latin hypercube, the final distribution of evaluated points, and the function space minima as predicted by the DACE predictor when used on the test grid. Note in Figure 8a that none of the initial points are near the correct minimum (red dot). One test point is, however, very close to the incorrect minima (green dot). Despite these biased starting conditions, the algorithm predicted a final value very close to the correct global minimum.

Table 3. Fifteen test runs of EGO on the Test131 PSA. Initial data set consists of 21 samples. The global minimum is at -10.262 dB.

Run #	Predicted Minimum, dB	% Difference from Global	Separation, mm	Frequency, MHz	# Iterations	# Evaluations
1	-10.256	0.059	9.4	851	51	71
2	-10.237	0.244	12.4	859	32	52
3	-10.249	0.127	11.3	8.56	51	71
4	-10.250	0.117	7.5	844	51	71
5	-10.207	0.536	7.6	847	51	71
6	-10.216	0.448	12.4	861	51	71
7	-10.194	0.663	6.5	867	51	71
8	-10.228	0.331	7.6	846	51	71
9	-10.262	0.000	8.7	848	51	71
10	-10.248	0.136	7.8	845	51	71
11	-10.252	0.097	10.4	854	51	71
12	-10.215	0.458	13.9	863	51	71
13	-10.204	0.565	15.1	864	36	56
14	-10.260	0.020	8.8	849	51	71
15	-10.216	0.448	12.4	861	51	71
<b>Average</b>	-10.233	0.283	10.8	854	49	69



(a)



(c)

Figure 8. (a) A poor initial sample point distribution, as generated by the Latin hypercube. Note the sample point very close to the non-optimal minima; (b) Final distribution of sample points after the algorithm converged, and (c) Location of the predicted minimum. The DACE predictor function space in (c) typically provides a closer match to the exact function space; however, even under these extremely poor starting conditions, the algorithm was able to find the global minimum and approximate the function space reasonably well.

#### 4.1.4 Comparison of EGO with the Nelder-Mead Downhill Simplex Algorithm

We also used the Nelder-Mead downhill simplex method implemented in the MATLAB function *fminsearch*. We assumed no knowledge of the function space and started the algorithm with a random guess in the input space. As expected, this algorithm does well if the initial starting point leads towards the global minima; however, it also converged a number of times into the director lobe minima of 10.154 dB as shown in Table 4. It also got “stuck” several times and converged prematurely in both minima. This could be either a characteristic of this algorithm or an artifact of this particular implementation. The total number of function evaluations between Nelder-Mead and EGO were surprisingly close for this problem. As shown in these tables the average number of expensive function evaluations for the Nelder-Mead technique was 69.

Table 4. Fifteen runs of the Nelder-Mead Simplex Algorithm on the Test131 PSA. The global minimum in the end fire directive gain is -10.262 dB.

Run #	Predicted Minimum,dB	% Difference from global minimum	Separation, mm	Frequency, MHz	# Iterations	# of Function Evaluations	Notes
1	-10.152	1.072	14.8	835	30	61	Wrong min.
2	-10.261	0.01	9.5	851	43	91	
3	-10.133	1.253	9.9	835	19	41	Wrong min.
4	-10.244	0.171	9.8	858	17	40	Stopped early
5	-10.257	0.046	10.2	853	39	73	
6	-10.155	1.041	14	834	29	57	Wrong min.
7	-10.134	1.247	19.1	835	17	39	Stopped early
8	-10.155	1.046	13.4	834	34	72	Wrong min.
9	-10.260	0.023	9.8	852	24	51	
10	-10.254	0.075	10.7	855	44	91	
11	-10.249	0.125	11.4	856	41	82	
12	-10.261	0.008	9.2	850	37	83	
13	-10.262	0.000	8.8	848	37	85	
14	-10.261	0.011	9.6	851	34	85	
15	-10.259	0.031	10.0	853	39	77	
<b>Average</b>	-10.220	0.411	11.3	847	32	69	

#### 4.1.5 Comparison of EGO with a GA for the PSA Design Problem

The textbook GA [14] has two adjustable variables,  $N_{ipop}$  and  $\delta$ . The baseline case is  $N_{ipop}=24$  and  $\delta=50\%$ . We ran the GA for three values of  $N_{ipop}$  (24, 32, and 40) and five values of  $\delta$  (30, 40, 50, 60, and 70%). The average results are shown in Table 5 for 30 individual test runs for each combination of  $N_{ipop}$  and  $\delta$ . Note that positive values of directivity are shown. As discussed in Section 4.1.3, the GA (and EGO) seeks minima so we minimize the negative of the directivity to obtain maximum directivity. That is why the values of directivity are shown as negative numbers in the ERGA results shown in Tables 3 and 4 above and in Table 8 on page 26.

Table 5. Average (over 30 test runs) results for the textbook GA at convergence as a function of the initial population and the mutation rate,  $\delta$ . The average number of calls to the expensive cost function is represented by Calls. The average predicted directivity is represented by Directivity.

$\delta$ (%)	30	40	50	60	70
<b><math>N_{ipop} = 24</math></b>					
Calls	189	280	376	431	389
Directivity	10.214	10.231	10.245	10.246	10.249
<b><math>N_{ipop} = 32</math></b>					
Calls	237	421	474	442	460
Directivity	10.232	10.235	10.246	10.244	10.248
<b><math>N_{ipop} = 40</math></b>					
Calls	358	458	463	493	497
Directivity	10.233	10.243	10.242	10.244	10.245

From Table 5, there are observations that can be made for all three values of  $N_{ipop}$ . As stated earlier, we used higher mutation rates than those discussed in [14]. For the lowest  $\delta$  of 30%, lack of diversity causes the simple GA to stop prematurely with a smaller average value of directivity. There is a very gradual increase in the average value of directivity as  $\delta$  is increased from 30% to 70%; however the increase is small (3%, 0.2%, and 0.1% for  $N_{ipop} = 24, 32,$  and  $40$  respectively) and there is a very large price to be paid, i.e. an increase in the number of expensive function calls (165%, 94%, and 39% for  $N_{ipop} = 24, 32,$  and  $40,$  respectively). There are also some very clear observations that can be made concerning the increase in the initial population. The average number of expensive cost function calls increases significantly with no significant increase in directivity. For the baseline  $\delta$  the directivity actually decreased from 10.245 to 10.242 at the higher value of 40 for  $N_{ipop}$ . For larger values of  $N_{ipop}$ , the stopping criterion of 500 total cost function calls was by far the most predominant stopping criterion as discussed below.

To see how the four different stopping criteria were used in the test runs of the textbook GA we have prepared Table 6. Stopping criterion #1 (Limit of 100 total generations) was never used. For  $N_{ipop} > 24,$  note the obvious transition from stopping criterion #3 (Lack of diversity) to stopping criterion #4 (Limit of 500 function calls) as  $\delta$  is increased from 30% to 70%.

Table 6. Distribution of the number of each stopping criterion used for the textbook GA for the three initial populations and five mutation rates,  $\delta$ . *Stagnation* represents criterion #2 (no change in cost associated with the number one chromosome after 20 generations); *Diversity* represents criterion #3 (7/8 of the genes in the mating pool are identical); and *500 Calls* represents criterion #4 (Number of cost function calls exceeds 500). Criterion #1 (Limit of 100 total generations) was never used.

$\delta$ (%)	30	40	50	60	70
<b><math>N_{ipop} = 24</math></b>					
Stagnation	0	7	11	13	26
Diversity	30	17	7	0	0
500 Calls	0	6	12	17	4
<b><math>N_{ipop} = 32</math></b>					
Stagnation	4	5	10	15	12
Diversity	23	10	0	0	0
500 Calls	3	15	20	15	18
<b><math>N_{ipop} = 40</math></b>					
Stagnation	6	6	7	6	1
Diversity	18	6	0	0	0
500 Calls	6	18	23	24	29

A direct comparison of the GA with EGO can be made by simply stopping the GA after 71 cost function calls and comparing the results. The EGO algorithm had an average predicted directivity of 10.233 dB after 71 cost function evaluations (Calls). As shown in Table 7, the average predicted directivity (over 30 runs) for the GA is less in all cases and is considerably less in most cases.

Table 7. Average predicted directivity for the textbook GA at 71 cost function calls.

$\delta$ (%)	30	40	50	60	70
<b><math>N_{ipop} = 24</math></b>					
Directivity	10.203	10.200	10.198	10.191	10.205
<b><math>N_{ipop} = 32</math></b>					
Directivity	10.209	10.197	10.186	10.204	10.191
<b><math>N_{ipop} = 40</math></b>					
Directivity	10.189	10.189	10.189	10.171	10.177

Another comparison can be made by considering the converged values of predicted directivity for the GA shown in Table 5. Consider the baseline case in Table 5. The GA gives a slightly higher predicted directivity of 10.245 compared with 10.233 for EGO (and both are close to the global minimum of 10.262). This is an increase of about 1.2%; however, the GA requires 376 function calls compared with only 71 for EGO, an increase of 430%. Again, this is a very large price to pay for expensive cost functions.

We now present results for the enhanced GA (ERGA) described in Section 3.2.2 to obtain another comparison with the EGO algorithm. For ERGA, the results, shown in Table 8, are presented differently than the results we presented for the textbook GA in Tables 5, 6, and 7. We allowed the ERGA algorithm to proceed for 100 iterations for all test runs, but recorded the directivity for the other stopping criteria (numbers of “Calls” or expensive function evaluations, and stagnation after 20 iterations) as those criteria were encountered. In Table 8, each row represents the average scores for 15 runs, for those particular parameter conditions. Unlike the textbook GA, ERGA was able to consistently find the global function minimum of 10.262 within 100 iterations once a good set of algorithm parameters was found. There was a range of parameter values for which the ERGA behaved robustly, i.e. it would always converge to the global minimum. The goal then became to see how to best tune the algorithm parameters for speed of convergence, to efficiently reach that minimum using the least numbers of function calls.

Table 8. ERGA Results for the PSA Problem.

GA Parameters								Average Scores (of 15 runs)						
Pop Size	% Mate	% Keep	Initial $\delta$	Rate $\delta$	Incr $\delta$	Final $\delta$	Initial $\delta$ Sigma	25 Iters	50 Iters	100 Iters	71 Calls	300 Calls	500 Calls	Stag
16	50	25	50	1.01	90	0.6	-10.248	-10.261	-10.262	-10.204	-10.247	-10.259	-10.260	
16	50	25	60	1.01	90	0.6	-10.250	-10.260	-10.262	-10.192	-10.248	-10.258	-10.258	
16	50	25	40	1.01	90	0.5	-10.244	-10.259	-10.262	-10.204	-10.241	-10.253	-10.260	
24	25	25	50	1.01	90	0.6	-10.254	-10.261	-10.262	-10.175	-10.247	-10.254	-10.257	
24	25	15	50	1.01	90	0.6	-10.256	-10.261	-10.262	-10.182	-10.247	-10.255	-10.260	
24	20	20	50	1.01	90	0.6	-10.256	-10.261	-10.262	-10.181	-10.238	-10.256	-10.259	
16	50	25	40	1.01	90	0.4	-10.249	-10.260	-10.262	-10.191	-10.247	-10.256	-10.256	
16	50	25	50	1.02	100	0.6	-10.253	-10.259	-10.262	-10.202	-10.249	-10.258	-10.260	
16	50	25	60	1.02	100	0.6	-10.251	-10.259	-10.262	-10.217	-10.249	-10.258	-10.259	
16	50	25	60	1.02	100	0.5	-10.250	-10.260	-10.262	-10.194	-10.250	-10.258	-10.256	

This function space has a very flat “bowl bottom” near the global minimum; there is essentially no single-parameter inheritance between good parents once the algorithm has achieved the bowl. Once in the bowl, both a change in frequency and a change in separation are required to move from one good point to a better point; thus mating two “good” parents while blending only one of the genes does not result in a better design (higher directivity). A lower mutation rate in the early generations of the algorithm allows for more inheritance and a more rapid location of the “bowl”. After that we need to transition to a higher mutation rate with a narrow sigma, to allow the algorithm to randomly bounce around in the bowl. A high mutation rate with a small sigma increases the chances that both parameters will vary simultaneously by a small amount, which is essential for traversing the relatively-flat valley to the global minimum.

In Table 8, we present runs with initial mutation rates varying from 40 to 60% (0.4 to 0.6). Only offset mutation was used in these runs; we did not utilize random value replacement mutation since the results

were not as satisfactory. After each generation is processed, the current mutation rate is multiplied by the “ $\delta$  Increase” until the “Final  $\delta$ ” is reached. Note that for some runs, the final mutation rate is 100%; however, since we are performing offset mutation, even 100% mutation results in only small perturbations around the blended parent values. The amount of allowable offset is determined by the “Initial  $\delta$  Sigma” and the amount that the sigma is decreased per generation. For all the results in Table 8, the “ $\delta$  Sigma” was decreased by 4% (of the current value) each generation, until a final  $\delta$  sigma of 0.05 was reached.

The results demonstrate that ERGA consistently found the global minimum of -10.262 after 100 iterations. The number of function calls, however, exceeds 500 which may make it undesirable for very expensive black-box functions. Comparing the average directivity for ERGA in Table 8 after 71 function calls to the EGO results in Table 3, we see that even the ERGA is not able to match EGO.

## 4.2 The Wideband Folded Triangular Bowtie Antenna (FTBA) Element

The PSA design optimization problem in Section 4.1 required the design optimization algorithm to select the values for two design variables: (1) element separation (mm) and (2) operating frequency (MHz). This is a 2-D problem so  $k = 2$ . The cost function was the end fire directivity. For each proposed array configuration this cost function was calculated using NEC [19]. This calculation is relatively fast and inexpensive and therefore we could compare EGO performance with both Nelder-Mead optimization and GA optimization. For the FTBA element in this section we calculate the cost function (the VSWR over the frequency band) using a full-wave CEM code. In this case we use the High Frequency Structure Simulator (HFSS) [20] discussed later. For this expensive cost function we use only EGO.

The optimization is also more complicated since there are six design variables, i.e. a 6-D problem, therefore  $k = 6$ . This requires us to modify the EGO algorithm in two areas: (1) the calculation of the correlation coefficients and (2) the determination of the point in the input space where the expected improvement is maximized. We discuss the modifications in more detail later in Section 4.2.5. First we discuss the FTBA element and the two configurations for which we perform a design optimization: (1) FTBA element backed by a metamaterial and (2) FTBA element backed by a perfect electric conducting disk.

### 4.2.1 The FTBA Element

A cavity-backed antenna has the desirable radiating properties of high gain, low sidelobes and low backlobes. Reduced radiation in the backward direction, i.e. large front-to-back ratio, makes it attractive for a number of applications. Also, the antenna can be wideband if the radiating element in front of the cavity is wideband. Qu, et al [21] proposed a wideband bowtie dipole as the radiating element [22]. A notional view of the element is shown below in Figure 9. The dipole has two narrow folding arms on either side of the driven bowtie. The bowtie is fed using a lumped port (voltage gap generator) between the two triangular dipole legs. The antenna exhibits good performance over more than an octave bandwidth at UHF (Ultra High Frequency) and SHF (Super High Frequency) frequencies. This antenna is referred to as a cavity-backed FTBA in the literature [21]. The cavity-backed FTBA is more compact than the short backfire antenna structure used for similar applications [21, 23 and 24].

We propose a backing structure which is potentially simpler and more compact than either the cavity-backed FTBA or the short backfire antenna. The concept is shown below in Figures 10 and 11 where we use the FTBA as the radiating element and a flat disk composed of a special metamaterial called an indefinite material [25] as the ground plane. The back surface of the indefinite material is a conducting plate. Design optimization requires a multi-parameter algorithm suitable for expensive cost functions since each proposed design is analyzed using a full-wave CEM simulation in HFSS.

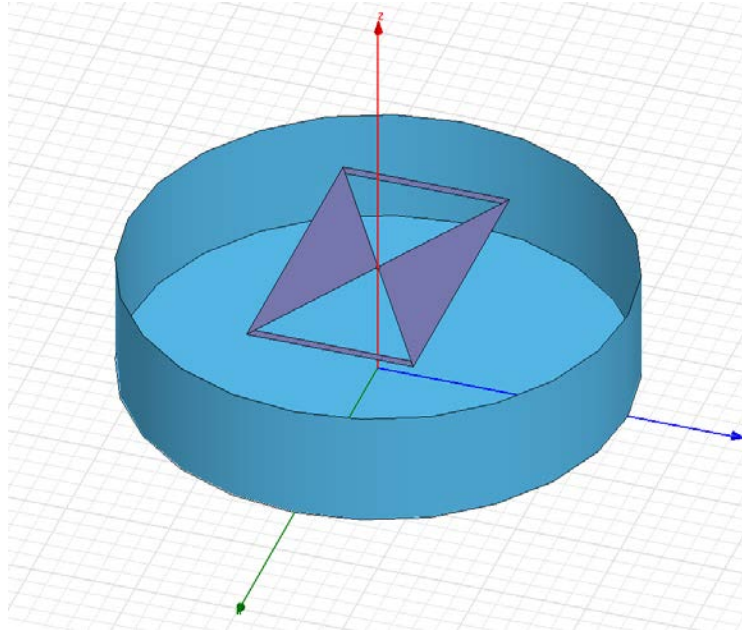


Figure 9. The FTBA element backed by a metal cavity. The coordinates are  $x,y,z$  counterclockwise from the lower left arrow. The two narrow folding arms are at either side of the bowtie element.

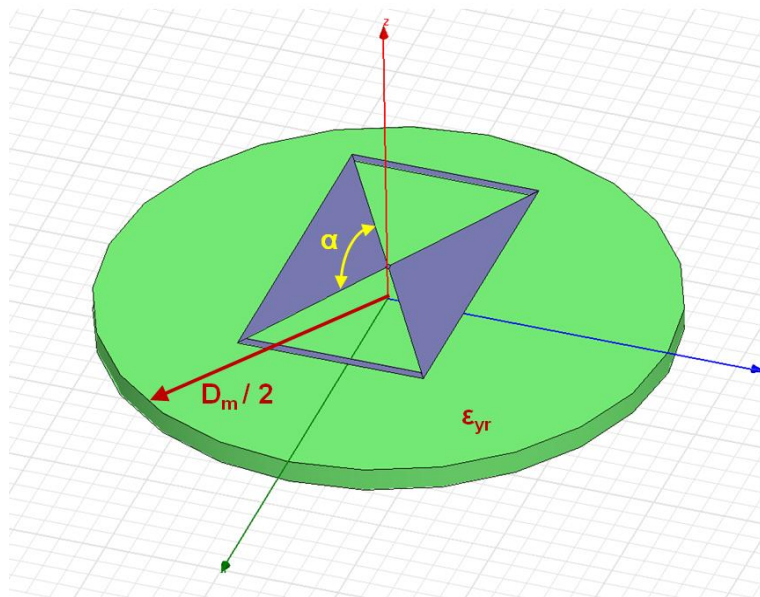


Figure 10. The FTBA element backed by a disk of indefinite material. Principal design parameters are shown. The back surface of the disk is covered by a conducting plate (not visible).

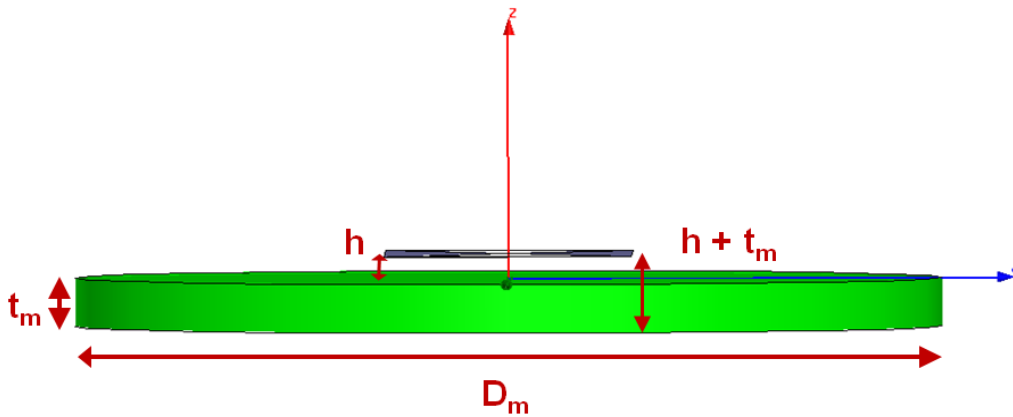


Figure 11. Side view of the FTBA element backed by an indefinite material. Principal design parameters are shown.

#### 4.2.2 The FTBA Element Over a Metamaterial Ground Plane

The polarization vector for the dipole in the above figures is in the  $y$  direction (to the right). The electric field is therefore primarily transverse, or parallel, to the metamaterial surface which is in the  $x$ - $y$  plane. For transverse electric, or  $TE_{xz}$ , plane waves,  $\mathbf{E} = \mathbf{y} \exp \{ j(\omega t - k_x x - k_z z) \} = \mathbf{y} E_y$ .  $E_y$  is directed out of the page in Figure 12 below. Propagation vector,  $\mathbf{k}$ , has components  $k_x$  and  $k_z$  as shown.

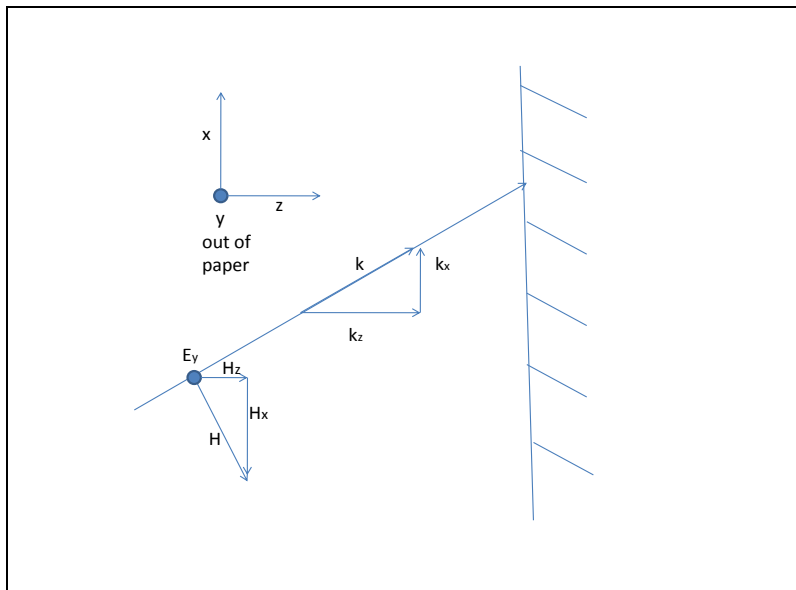


Figure 12. Transverse Electric (TE) wave incident on a metamaterial.

The semi-infinite region to the left is free space. The metamaterial is represented by the semi-infinite cross-hatched region to the right. The metamaterial constitutive parameters are defined by relative permittivity tensor  $\epsilon_r = \text{diag} [\epsilon_{xr} \ \epsilon_{yr} \ \epsilon_{zr}]$  and relative permeability tensor  $\mu_r = \text{diag} [\mu_{xr} \ \mu_{yr} \ \mu_{zr}]$ , where *diag* represents a diagonal matrix. In free space the diagonal elements are equal to unity. For an indefinite material, not all of the diagonal elements have the same sign. Such materials are artificial, structured, composite media with unit cell dimensions much smaller than a wavelength [25]. They are composed of periodically positioned scattering elements (conductors), and have been shown to exhibit simultaneously negative effective permittivity and negative effective permeability [26]. We consider a special kind of indefinite material called a unit magnitude anti-cutoff material which has  $\epsilon_{yr} = -1$  and  $\mu_{zr} = -1$  [25].

In the CEM simulation we assume that the relative permittivity and relative permeability are constant over the frequency band (2 to 5 GHz). In real materials there would be dispersion, or frequency variation. Additionally, we assume that the primary field produced by the dipole is transverse electric. This TE field will encounter a unit magnitude reflection coefficient at the surface of the indefinite material as discussed below. The full-wave CEM code will, of course, model all field components (including non-TE waves) but these will not interact with the indefinite material the same way. To demonstrate the feasibility of the EGO design optimization approach, the emphasis in this report is on coupling the optimization algorithm to the CEM engine; therefore we use a very simple material model.

For plane wave solutions, the dispersion relation for TE wave propagation is given by [25]:

$$k_z^2 = \epsilon_{yr} \frac{\omega^2}{c^2} - \frac{\mu_{xr}}{\mu_{zr}} k_x^2. \quad (23)$$

The surface reflection coefficient is given by [25]:

$$\rho = (\mu_{xr}k_z - q_z) / (\mu_{xr}k_z + q_z), \quad (24)$$

where  $\mathbf{k}$  and  $\mathbf{q}$  are wave vectors in free space and the metamaterial respectively. Using Equations 23 and 24 (and the fact that  $k_x$  represents variation transverse to the media surfaces and is continuous across the interface) Smith and Schurig [25] show that the TE surface reflection coefficient is  $\rho = +j$  independent of the angle of incidence. The reflection coefficient has unit magnitude and a  $90^\circ$  phase shift. The surface impedance is equal to  $+j377 \ \Omega$ . For a perfect electric conductor (PEC) and a perfect magnetic conductor (PMC),  $\rho = -1$  and  $\rho = +1$ , respectively.

We will now investigate the effect of an infinite ground plane with three different reflection coefficients on an antenna element located above that ground plane. For simplicity, consider a horizontal, ideal dipole over an infinite ground plane. We can use image theory and the surface reflection coefficient to calculate the total radiated power as a function of dipole height above an infinite ground plane. In Figure 13 the total radiated power is shown relative to the power of an ideal dipole located at the surface (height=0) of a PMC ground plane. As the dipole height approaches zero the PEC ground plane shorts out the dipole and the total radiated power vanishes. For a PMC ground plane, reinforcement occurs. For the anti-cutoff

indefinite material (which we refer to henceforth as the metamaterial or MM) the effect is intermediate; however, note that a significant amount of power is radiated even when the dipole is closer to the surface.

The above discussion suggests that for certain MM ground planes, radiating elements can be located closer to the surface without inducing field-cancelling image currents which occur in metallic ground planes. This creates the opportunity for more compact, lower profile antenna designs.

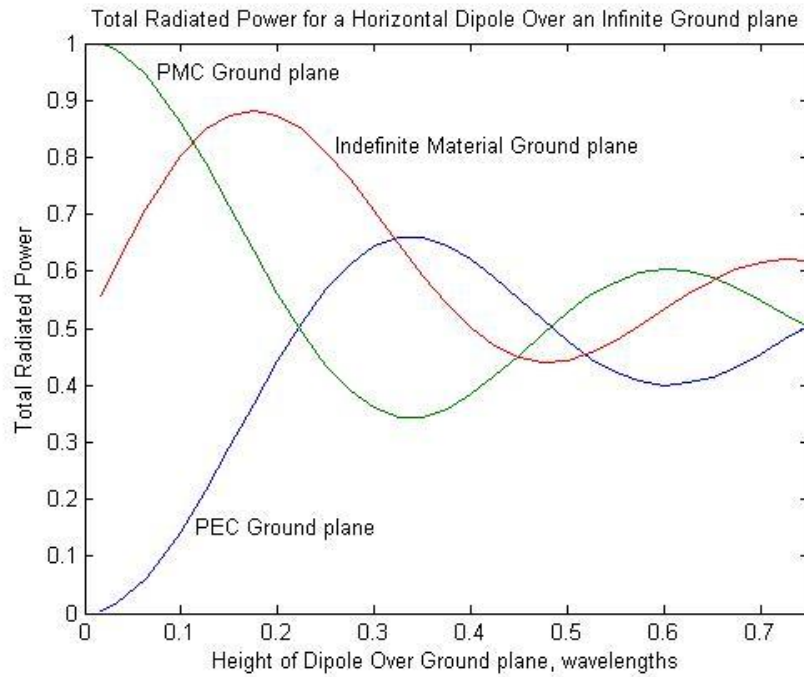


Figure 13. Total radiated power for three cases of an ideal dipole over an infinite ground plane consisting of: (1) a Perfect Magnetic Conductor (PMC); (2) an anti-cutoff indefinite material; or (3) a Perfect Electric Conductor (PEC).

### 4.2.3 The FTBA Element Over a PEC Ground Plane

The indefinite material described above is not currently available. Therefore, to obtain measured data for an optimized antenna design generated using EGO/HFSS design optimization we designed a simpler antenna using a finite PEC ground plane. This antenna structure is the FTBA element over a conducting disk shown in Figure 14. This structure has been fabricated and tested over the design frequency band of 2 to 5 GHz. It is a special case of the short backfire antenna without the front reflector. The short backfire antenna is described in Reference 24 (page 111), where for our case, reflector #1, i.e.  $R_1$ , is absent and on reflector #2, the rim height,  $w$ , is zero.

Since a feed will be required when the antenna is built using a MM, we also use this simpler antenna structure to design and optimize the feed. This will help us learn more about designing the feed. The balun/feed shown in Figure 15 differentially drives the dipole.

The E-plane for this antenna configuration is in the blue-red  $y$ - $z$  plane in Figure 14 which is normal to the feed substrate below the element. The H-plane for this antenna configuration is in the green-red  $x$ - $z$  plane in Figure 14 which is parallel to the feed substrate below the element.

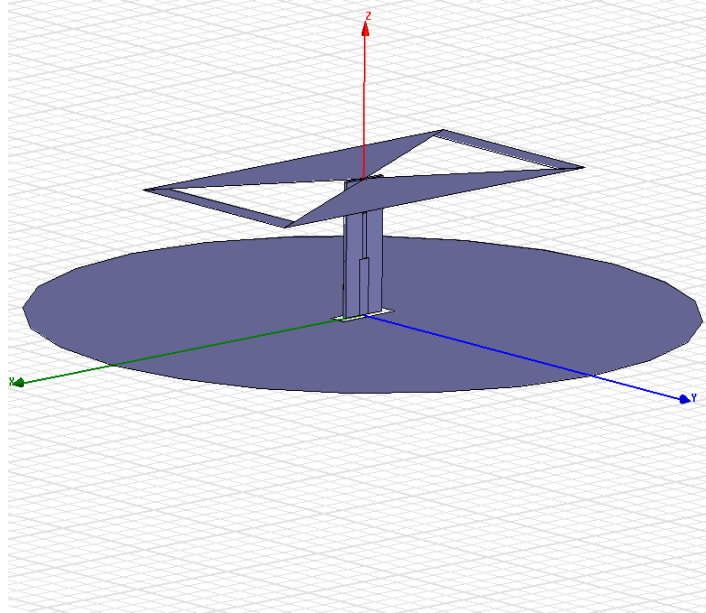


Figure 14. FTBA over a finite PEC ground plane (disk).

#### 4.2.4 The Antenna Balun/Feed for the FTBA Element

A front view of the balun/feed structure is shown in Figure 15 and is similar to the one used by Qu, et al [21]. The structure is a double-sided Rogers 5880 substrate ( $\epsilon_r = 2.2$ ) which is 31 mils (0.787 mm) thick and 10 mm wide. We use a coaxial SMA connector to couple a source to the feed. Since the coax is unbalanced we require a balun to feed the balanced dipole. The first (bottom) trace in Figure 15 is unbalanced 50 ohm microstrip with a ground trace on the back of the substrate. The length of the microstrip section is determined by design optimization. The middle trace is a transition section of balanced double-sided parallel stripline (DSPSL). The length and width of this trace is determined through design optimization. The top trace is 120 ohm DSPSL. The transition and the top traces have identical traces on the back side of the substrate. As discussed at the end of Section 4.2.6, the 120 ohm value is found by optimizing the design of the FTBA over the PEC ground plane using a lumped port (voltage gap) excitation without a feed. Therefore, we effectively separated the design of the feed (having three design variables) from the antenna element design (having six design variables).

The length of the top trace (narrowest trace in Figure 15) depends on the lengths of the first two traces since the height of the antenna above the ground plane is determined in the antenna design optimization. We used EGO/HFSS to find optimum values for: (1) the length of the 50 ohm microstrip trace, (2) the

length of the DSPSL transition trace, and (3) the width of the transition trace. The values are: 11.37 mm, 10.71 mm, and 1.57 mm, respectively. The top DSPSL traces (both sides) extend through the plane of the dipole and form a solder tab which is used to solder the feed to the two legs of the dipole. This can be seen in Figure 17b.

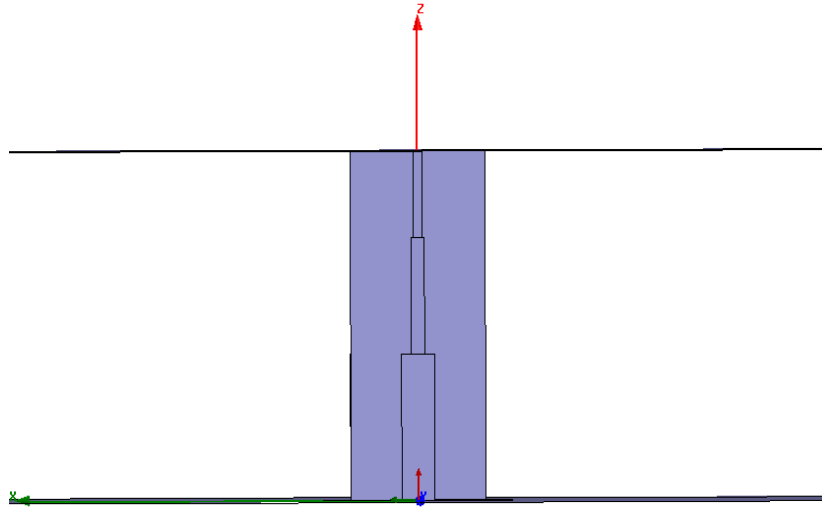


Figure 15. Front view of the balun/feed structure. The Rogers 5880 substrate is in the plane of the paper.

#### 4.2.5 Multi-Parameter Design Optimization Using EGO

There are six design variables for the FTBA design optimization problem and any combination of these six variables constitutes a single antenna design. The variables are used by the CEM simulator to define the geometry of the antenna structure which is then analyzed to yield RF performance characteristics (in particular, the VSWR over the 2 to 5 GHz band). The design parameters and their ranges are shown in Table 9. We also include a nominal design with parameter values near the range centers. The values of the nominal parameters are also reasonably close to those in the paper by Qu, et al [21].

Table 9. Ranges for the six design parameters of the folded triangular bowtie antenna (FTBA) over an indefinite material (MM).

**Note: MM represents metamaterial and mm is millimeters.**

Design Variable		Minimum Value	Maximum Value	Nominal Design
$t_m$	MM Thickness (mm)	5	20	10
$D_m/2$	MM Radius (mm)	50	70	60
H	Dipole height (mm)	2	10	5
L	Dipole length (mm)	35	45	40
$\alpha$	Bowtie angle (degrees)	110	130	120
$W_a$	Folding arm width (mm)	1	5	2

The fundamentals of EGO were described in Section 3 and we have described the application of EGO to antenna design in previous papers [2, 15, and 27]. In these papers, we used two design variables (a 2-D design problem) for demonstrating the feasibility of the technique and for ease in visualizing results. The current FTBA element design optimization problem requires six design variables. In Section 4.3 we describe an antenna problem with 11 design variables. While the EGO algorithm is conceptually scalable, our previous implementations of EGO utilized techniques that were impractical for larger numbers of dimensions. The following techniques required modification before EGO could be successfully applied to problems with larger dimensionality: (1) determination of the correlation parameters [1, 2]; and (2) maximization of the quantity expected improvement.

The first modification involves a technique different from Nelder-Mead for finding the correlation parameters. We follow Welch, et al [28] and use an algorithm which sequentially introduces the parameters. The algorithm is called a screening algorithm since it gives an indication of the importance of each design variable. Variables with larger correlation parameters are more active, or important, and ones with zero value are not important.

The screening algorithm performs a series of less expensive one-dimensional line searches to determine which correlation parameters should receive their own values. The remaining parameters are constrained to have the same common value. The parameter whose line search results in the largest value of the likelihood function is removed from the set of common value parameters. Line searches sequentially remove parameters from the set of common values and lead to good initial guesses for a multi-parameter Nelder-Mead search. The goal is to make this multi-parameter Nelder-Mead search more efficient and reliable. The algorithm terminates when the difference in the increase in the log of the likelihood function becomes less than a threshold value [28]. The algorithm can also terminate when the set of common value parameters has a value of zero. This means that these parameters have no effect on the cost function. Correlation parameters are used to construct the DACE predictor [1, 2] which is an inexpensive surrogate model of the response surface.

The next problem with EGO for higher dimensional design problems is to determine the location in the input space where the expected improvement is maximized. This determines the next sample point. For problems with limited dimensionality, the Cox and John gridding method [29] provides a slow but robust method. A fine  $k$ -dimensional mesh is defined across the function space and the response surface is evaluated at each point within the mesh. While this ensures that the response surface is sampled

completely, it is a time-consuming process which is only feasible with problems of reasonably low dimensionality. A secondary problem is that the precision of the parameters used to generate the solution is limited to the resolution of the grid.

The creators of EGO [1] used a “branch and bound” mechanism for evaluating the DACE predictor for multiple dimensions. However, they indicated that convergence was slow for greater than 6 dimensions and they implemented a “limited memory branch and bound” algorithm for problems with greater dimensionality. We chose instead to implement a continuous-valued GA as described by Haupt [14] to evaluate the DACE predictor and determine the combination of design variables which result in the maximum value of the expected improvement in the input space. This combination of design variables is the next design point. While the GA is not guaranteed to find this point exactly, we know that it will find a good point, i.e., a point where the expected improvement is very high. A sample taken at that point, even if it is not exactly at the point of maximum expected improvement, still provides very useful information for tailoring the response surface to further improve our DACE predictor model.

The GA begins with an initial random population of 1500 chromosomes. Note here that the chromosome consists of  $k$  parameters, where  $k$  is the number of dimensions being optimized, i.e., the dimensionality of the function space. The members of the GA population can not include any points which have already been sampled (the actual value of the cost function is already known at these points). The DACE predictor is evaluated for each point represented by the members of the population and the expected improvement is calculated at those points. A multiple elitist GA is used, with the best 40% of the parents maintained in the population and the remainder consisting of new children. A fixed 20% mutation rate is used and the GA allowed to progress for 1000 generations. Upon completion, the point which generates the maximum expected value is used by EGO as the next point to sample with the expensive cost function.

While the DACE-evaluation GA with 1000 generations of 1500 chromosomes may seem like a time-consuming operation, the individual evaluations of the DACE predictor are computationally very rapid. On a modest personal computer, the entire GA run typically takes less than 20 minutes.

#### **4.2.6 Coupling EGO with a Full-Wave CEM Simulation**

Coupling an optimization engine with an exterior cost function engine, i.e. a different computer code, can be one of the more challenging aspects in setting up a design optimization. For our problem, a full-wave CEM simulator was required. We selected the 3D full-wave electromagnetic field simulator HFSS (High Frequency Structure Simulator) [20]. One attractive feature of this code is the ability to export results directly into MATLAB. We also require external control of the program, in this case from our MATLAB implementation of EGO.

Initially, we submitted requests to HFSS using Ansoft’s Visual Basic Scripting (vbs) language and the HFSS batch job command line interface. The parameters selected by EGO for the next design point were coded into appropriate vbs commands, along with commands to open the project, run the analysis, and export results. This script was then called by a command line execution of HFSS, which launched the program, ran the script (which analyzed the appropriate antenna configuration and exported results) and then terminated HFSS. While this method yielded acceptable results, it suffered from the overhead of repeatedly launching and terminating HFSS. Since our ability to run HFSS also depended on acquiring a floating-license at runtime, a second drawback to this method was the continual acquiring and surrendering of the HFSS license and the potential loss of that license.

Fortunately, a better method was found. Using an Active X Server opened to the HFSS Script Interface we could open and control HFSS directly from MATLAB. This had two major benefits over the batch-file interface. First, the program could be opened once and maintained open throughout the entire optimization. Since most of the computational time of the optimization was spent within this “expensive cost function”, this was a very acceptable tradeoff which eliminated the overhead of opening and closing the program many times. More importantly, HFSS results, such as numbers of meshing cycles before convergence, overall model size, and program errors, could be reported back to MATLAB and then used to guide or terminate the optimization.

The antenna Z parameters were exported from HFSS to MATLAB. Using the Z parameters we can determine the best voltage standing wave ratio (VSWR) of the antenna, when matched to feed line impedance  $Z_0$  in the range from 20 to 250  $\Omega$ . VSWR=1:1 represents a perfect match but VSWR < 2:1 is acceptable. The cost function penalizes the design by summing the squared difference of any VSWR > 1.25:1 across the band. Minimizing this cost function provides the best impedance match.

#### 4.2.7 Antenna Designs Using EGO/HFSS

Since a metamaterial with electric and magnetic properties constituting an indefinite material was not available, we include a design for an FTBA element backed by a PEC ground plane. We fabricated this antenna design and obtained measured results to compare with the theoretical results from our EGO/HFSS optimum antenna design.

Three FTBA designs were produced using EGO coupled with HFSS. Two designs incorporate a backing ground plane consisting of the metamaterial described in Section 4.2.2 and the final design has a PEC as a backing ground plane. The first design has relative permittivity tensor  $\epsilon_r = \text{diag} [1 \ -1 \ 1]$  and relative permeability tensor  $\mu_r = \text{diag} [1 \ 1 \ -1]$  which has a unit magnitude surface reflection coefficient for TE waves. The second design has relative permittivity tensor  $\epsilon_r = \text{diag} [-1 \ -1 \ 1]$  and relative permeability tensor  $\mu_r = \text{diag} [1 \ 1 \ -1]$  and has a unit magnitude reflection coefficient for both TE and TM waves. The MM thickness for the FTBA over the finite PEC ground plane is zero since there is no MM. The three designs are shown in Table 10. Theoretical performance of the two MM-backed designs is reported in Reference 43.

Table 10. Antenna design variables for the three FTBA element designs.

The Six Design Variables	All dimensions in mm ( $\alpha$ in degrees)	MM-backed (TE Case)	MM-backed (TE/TM Case)	PEC-backed element
$t_m$	MM thickness	15.94	14.10	0.00 (no MM)
$D_m/2$	MM or PEC radius	63.11	63.65	70.00
$h$	Dipole height	8.47	10.00	23.94
$L$	Dipole length	41.30	39.54	43.31
$\alpha$	Bowtie angle	119.53	119.89	125.32
$W_a$	Folding arm width	1.56	1.01	5.00

#### 4.2.8 The Experimental FTBA Over a PEC Ground Plane

The FTBA shown in Figure 14 and characterized by the six design variables in the last column in Table 10 was fabricated and tested. The antenna feed was described in Section 4.2.3 and is shown in Figure 16 below. The feed was fabricated and integrated with the FTBA and the PEC ground plane. The PEC ground plane is a brass disk 140 mm in diameter. The feed and the experimental FTBA element are shown in Figures 16 and 17.

The FTBA over the square ground plane was only used as a transmitting source for gain measurements since the element is so broadband. Measured data were obtained for the circular PEC ground plane only.

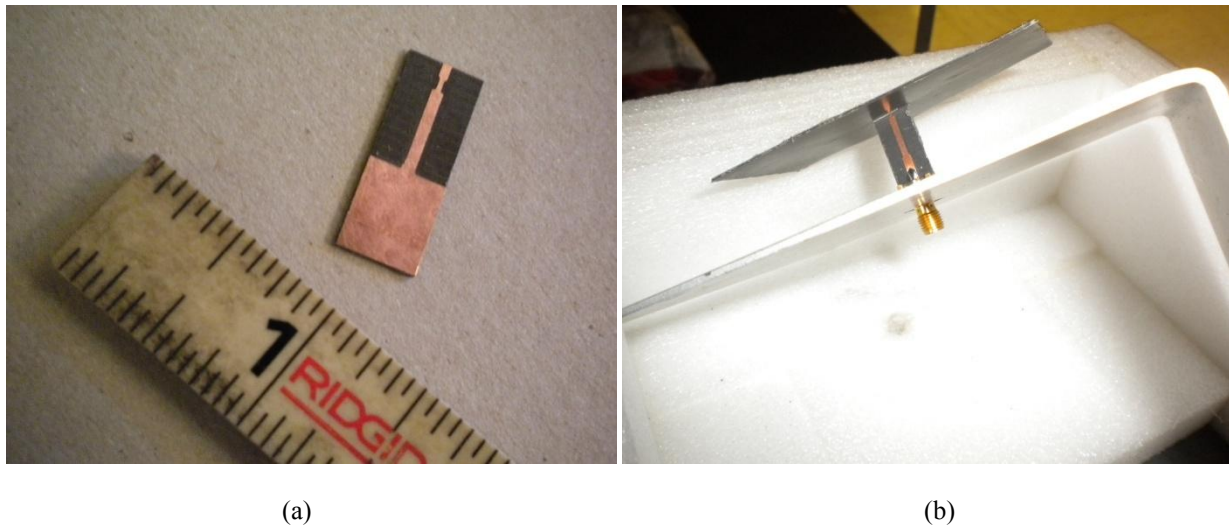


Figure 16. (a) The “underside” of the balun/feed structure on a double-sided Rogers 5880 substrate. (b) The “trace” side of the balun/feed structure showing the coaxial SMA connector on the bottom of the ground plane. The center conductor of the SMA connector is soldered to the top trace of the microstrip portion of the feed/balun.



Figure 17. (a) Top view of the FTBA element over both square and circular ground planes. Only the circular ground plane was used for measurements to compare with predicted results. (b) Side view of the FTBA element showing a solder tab at the top of the feed structure soldered to a leg of the dipole.

**4.2.9 Measured Data for the FTBA over a PEC Ground Plane**

All measurements were made at the AFRL/RHYA Ipswich Antenna Research Facility, Ipswich, MA [30].

Recall that our cost function (basically a figure of merit) for the FTBA design optimization problem was to make the VSWR as close to 1.25 as possible. A VSWR of 1.25 corresponds to a return loss of approximately -19 dB. The predicted and measured return loss for the optimum design of the FTBA element over a PEC ground plane is shown in Figure 18 below. The predicted results for all of the measurements were obtained using HFSS with the dimensions of the optimum design in the last column in Table 10. The predicted results agree very well with the measurements and although the return loss exceeds -19 dB it is less than -10 dB over the frequency band of 2 to 5 GHz except for the extreme low end of the band. Even at the low end of the band it is very close to -10 dB.

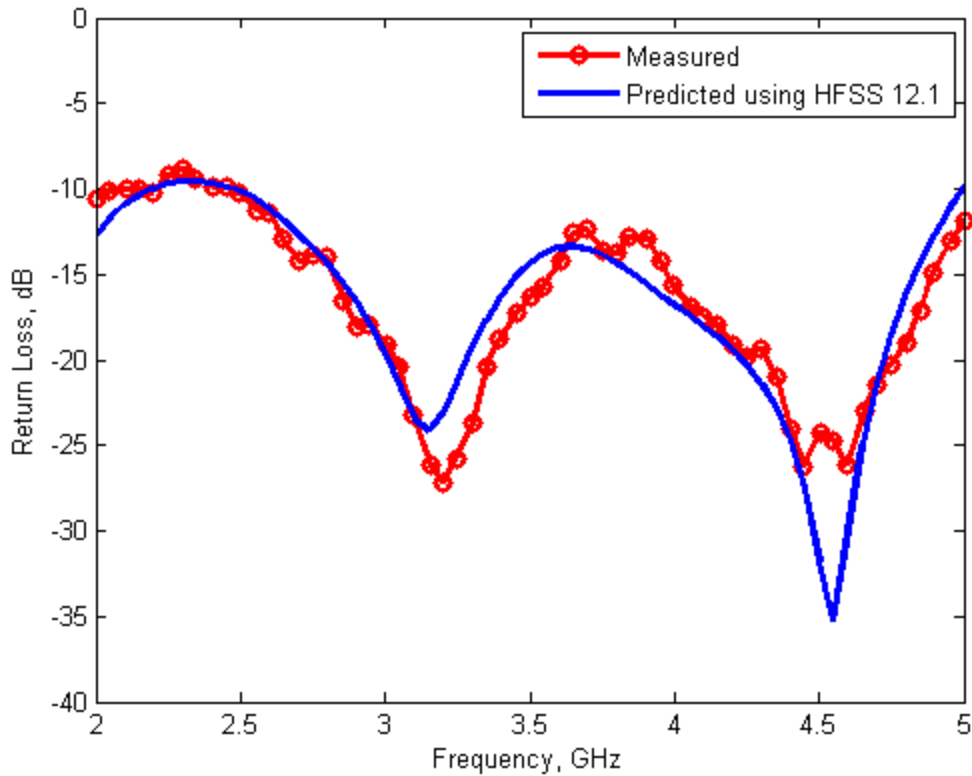


Figure 18. Predicted and measured return loss in dB for the FTBA over a PEC ground plane.

Another very important performance parameter for the antenna element is gain. Although we do not explicitly consider gain in the design optimization it is desirable to know the gain of the resulting design. We also want to compare the gain of our optimized design with the gain obtained by Qu, et al [21] for an FTBA over a circular PEC ground plane using their values for the six design variables. We show the broadside gain in dBi (dB over isotropic gain) for three cases: (1) our predicted results using HFSS, (2) measured results from the experimental FTBA over a circular PEC ground plane and (3) Qu's predicted results from [21]. Using EGO design optimization we were able to increase the broadside gain over more than 75 % of the frequency band compared with Qu's results.

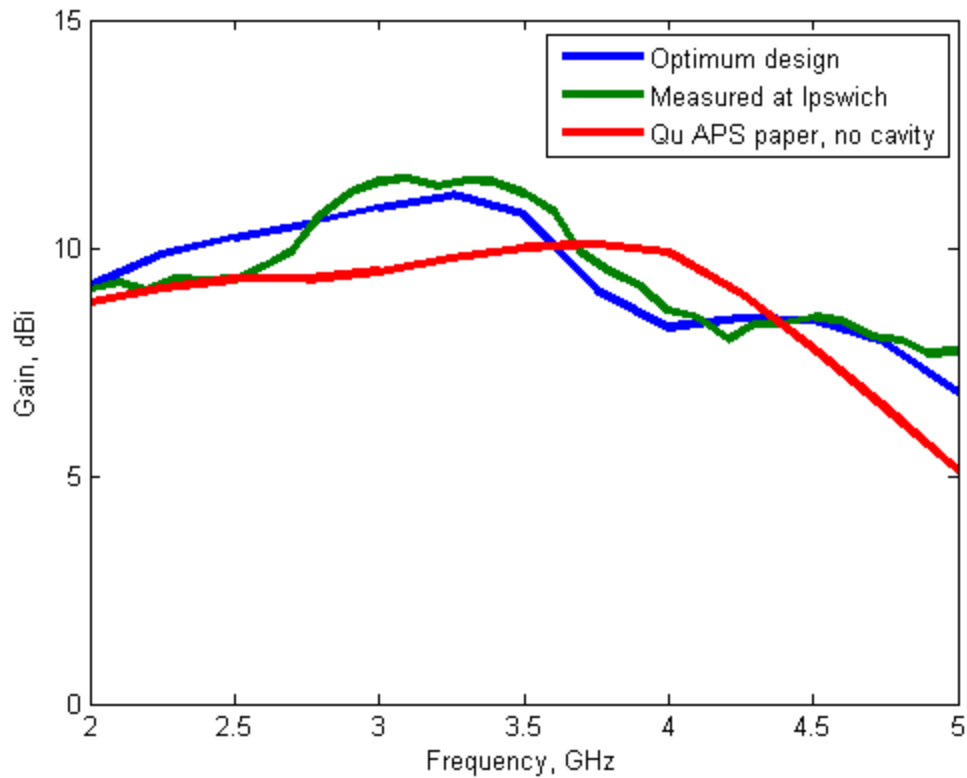


Figure 19. Predicted (blue) and measured (green) results for the broadside gain (in dB over an isotropic radiator) for the FTBA element over a circular PEC ground plane. The results from Qu, et al, [21] for the same configuration, but different dimensions, are shown by the red curve.

The antenna patterns for the FTBA element over a circular PEC ground plane were also measured at the AFRL/Ryha Ipswich Antenna Research Facility, Ipswich, MA [30]. The mounting pedestal is shown in Figure 20. The co-polarized and cross-polarized antenna patterns were measured at the three frequencies 2.0, 3.5 and 5.0 GHz in both the E-plane and the H-plane. For comparison, the predicted results from HFSS using our EGO optimized design variables are presented with the measured results from Ipswich in Figures 21 to 26. Predicted results using HFSS are blue and the Ipswich measurements are red. In general there is excellent agreement between the predicted and measured results.



Figure 20. The FTBA on a mount at the AFRL/Ryha Ipswich Antenna Research Facility, Ipswich, MA. The standard gain horns to the left of the mount were used in the gain measurements. The white frame is foam structural support. In this configuration the antenna is horizontally polarized.

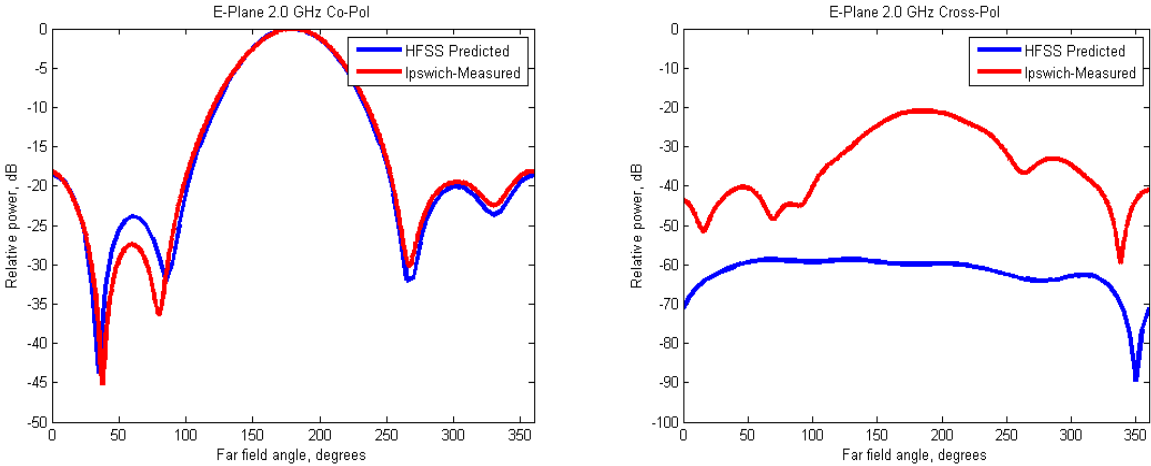


Figure 21. E-plane co-polarized and cross-polarized responses at 2.0 GHz.

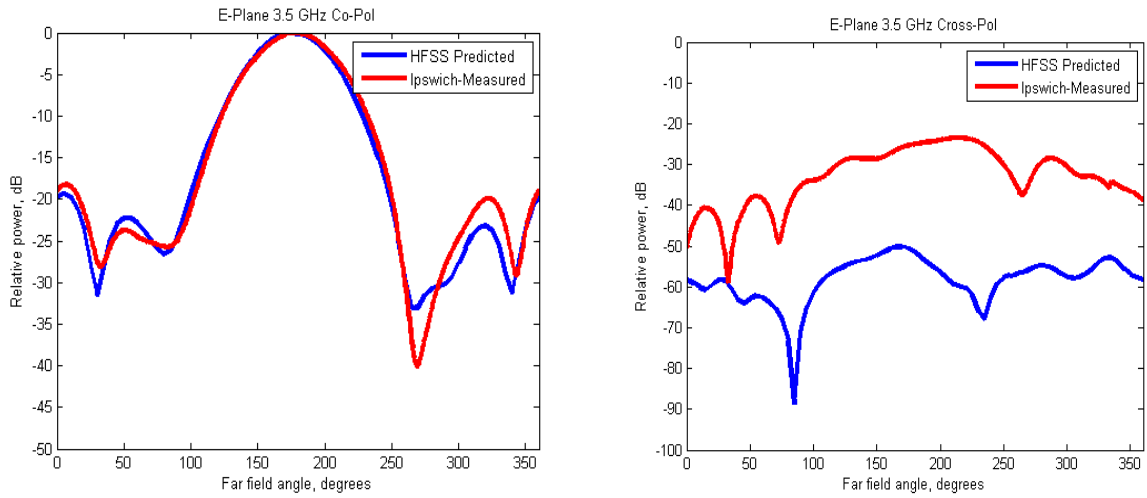


Figure 22. E-plane co-polarized and cross-polarized responses at 3.5 GHz

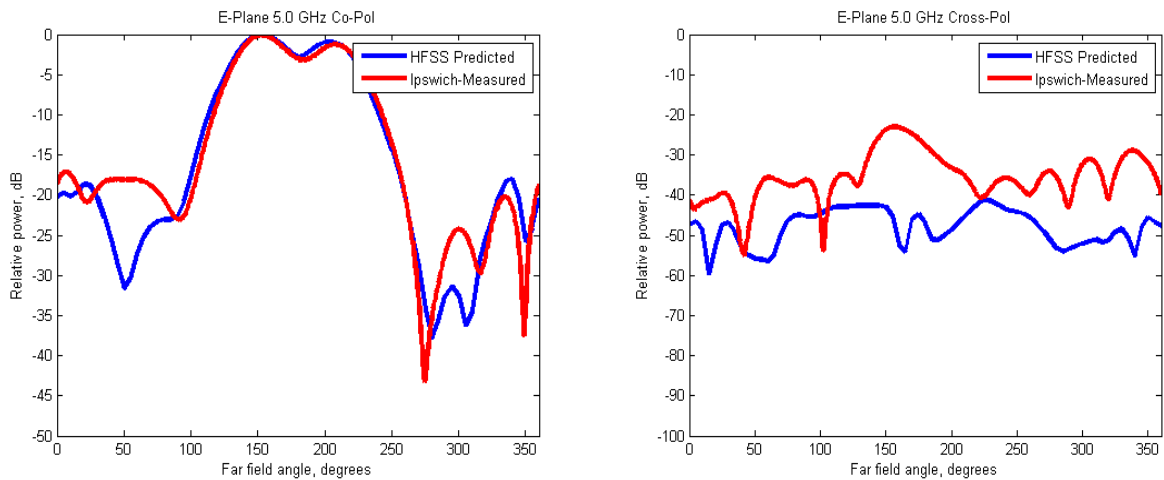


Figure 23. E-plane co-polarized and cross-polarized responses at 5.0 GHz

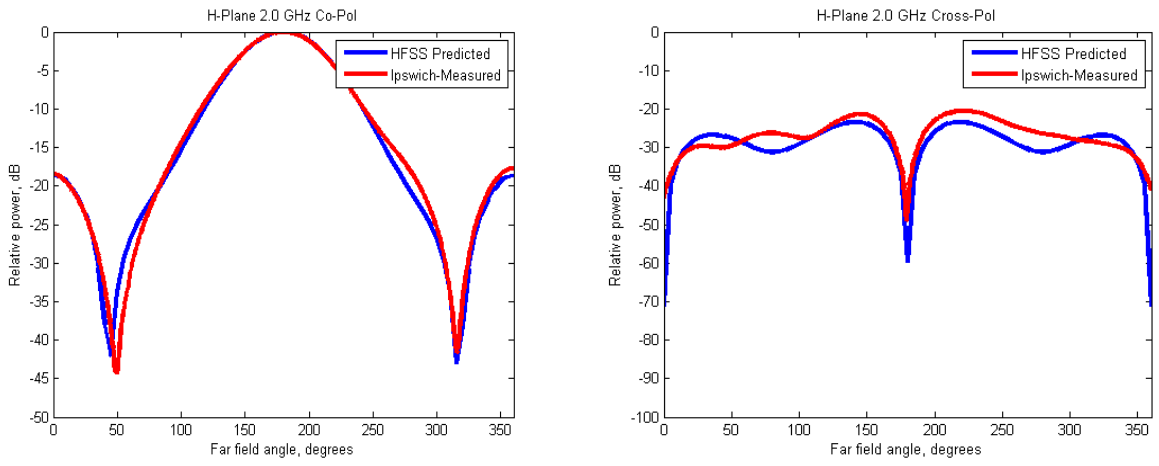


Figure 24. H-plane co-polarized and cross-polarized responses at 2.0 GHz

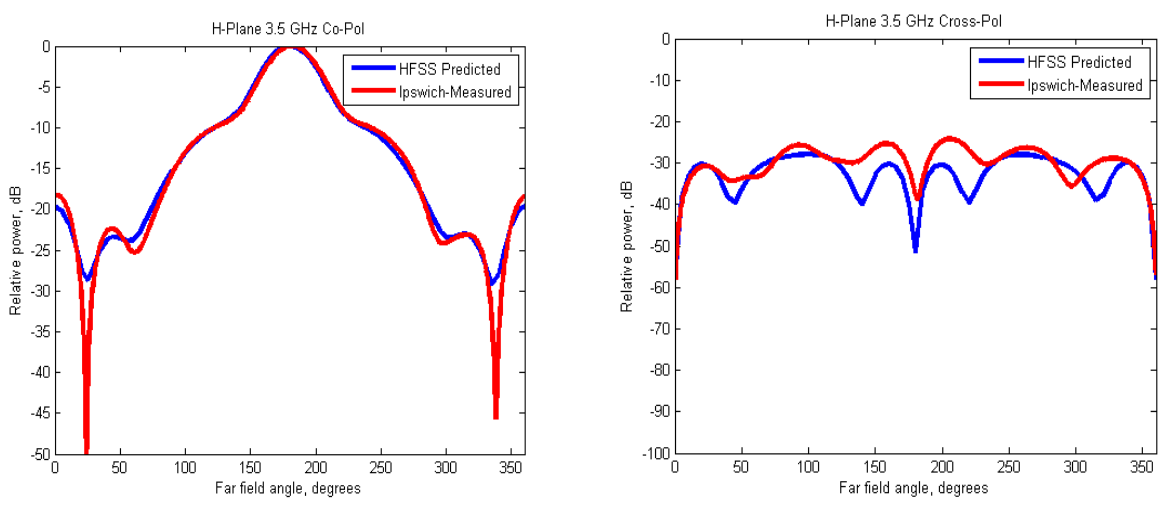


Figure 25. H-plane co-polarized and cross-polarized responses at 3.5 GHz

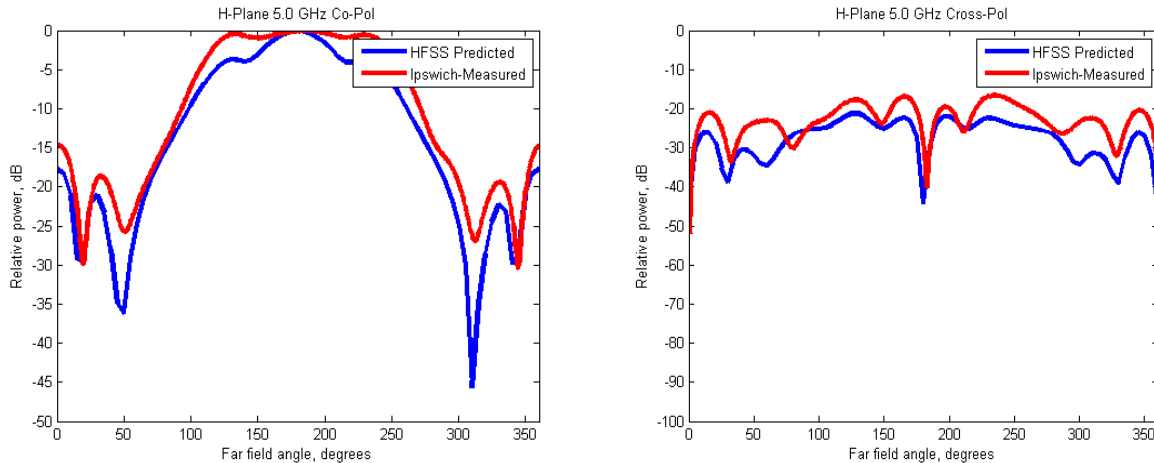


Figure 26. H-plane co-polarized and cross-polarized responses at 5.0 GHz

### 4.3 The Wideband Fragmented Patch Antenna Element

#### 4.3.1 Introduction

Military and commercial applications utilize wideband, wide scan-angle phased arrays to provide wideband operation with multiple functions using a single aperture. However, many wideband array elements, such as the flared notch, “bunny ear,” and TEM horn are three-dimensional structures which are complicated and costly to manufacture. Planar elements such as patches or printed dipoles offer a simplified geometry with potentially reduced cost and conformal applications. The goal is to design these types of elements with the scan and bandwidth characteristics required for the intended array applications.

GA techniques have previously been employed to design and optimize fragmented patch wideband array elements [31, 32 and 33]. However, they required extensive optimization times. Since genetic algorithms typically require many cost function evaluations, and the time required to complete a single configuration evaluation may take many minutes, some of these optimizations have required weeks of computational time. It has also been the case that the simple genetic algorithms previously employed for this effort have not converged reliably without seeding the population, i.e., introducing a known “good” configuration into the initial population.

The efficient global optimization (EGO) technique has been applied to this problem to produce solutions better than the GA in significantly less time [44]. As shown in Sections 4.1 and 4.2 and in References 2 and 34, EGO has been applied to antenna design optimization with limited numbers of design variables. However, this is the first time that EGO has been applied in the antenna design community to a problem of such large dimensionality, in this case 11 design variables.

### 4.3.2 The Fragmented Patch Antenna Element

The goal is to design a patch element in a unit cell of an infinite, periodic phased array which exhibits wideband performance. Figure 27 shows an element and a portion of the infinite array. Top and side views of the unit cell are depicted. The cell includes substrates and superstrates of variable thicknesses and relative permittivities. While an infinite array is not feasible, it is a useful computational tool for analysis and the results are acceptable for large arrays.

The patch is fed at its center by a delta-gap voltage source and the top layer of the patch is fragmented into a set of pixels that are either conducting or non-conducting. The goal is to determine the pixel distribution and the substrate and superstrate characteristics (thickness,  $d$ , and relative permittivity,  $\epsilon$ ) for optimal wideband and wide scan angle performance.

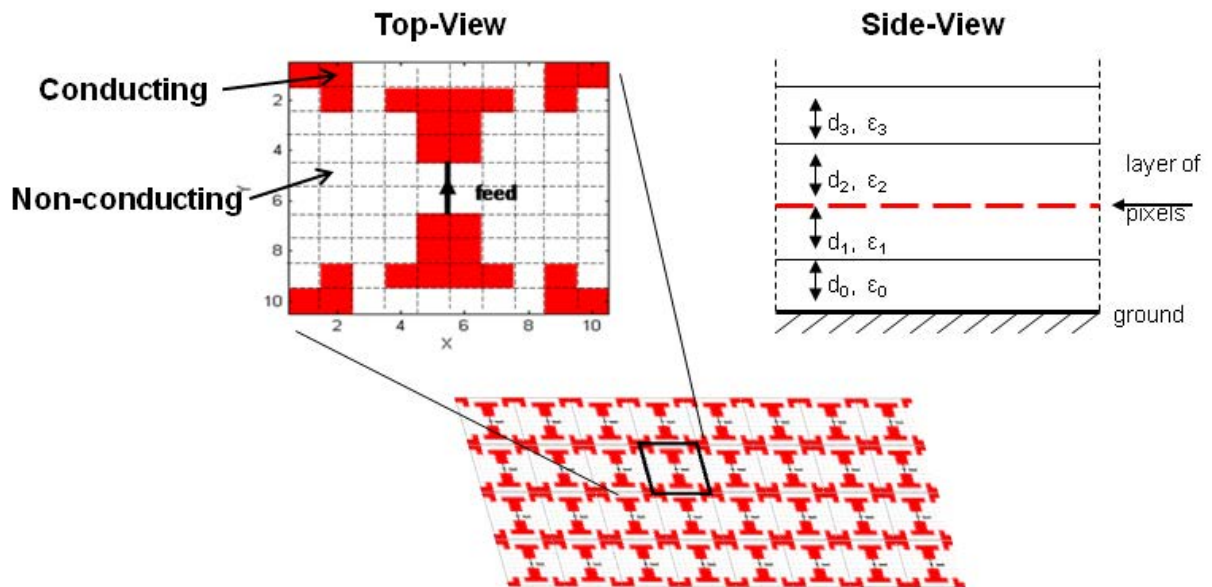


Figure 27. Infinite planar array of fragmented patch unit cells. Horizontal and vertical cross sections of a unit cell are shown. The cell has two substrates and two superstrates. Conducting pixels are shown in red.

### 4.3.3 Patch Modeling

The parameters include conducting or non-conducting regions, or „pixels,“ on the surface of the fragmented patch and the thickness and permittivity of two substrates and two superstrates. We model a patch using a ten-by-ten pixel grid, as shown in Figure 28, and use a Finite Difference Time Domain (FDTD) code [35] to analyze the array. A single five-by-five quadrant of the grid (consisting of 25 pixels) is optimized. The total patch is then constructed by applying the appropriate flipped and/or rotated pixel distribution to the remaining quadrants to enforce symmetry. This not only reduces the number of pixels to be optimized, but also corresponds to the desired symmetric radiation characteristics across the scan region. Within the five-by-five quadrant, the individual pixels have two states - either “metal” (conducting) or “air” (non-conducting). The patch is restricted to have specific metal or non-metal pixels within the fixed feed region (consisting of the four pixels in the green, cross-hatched region in Figure 28). The number of pixels to be optimized by the algorithm is therefore reduced to 21.

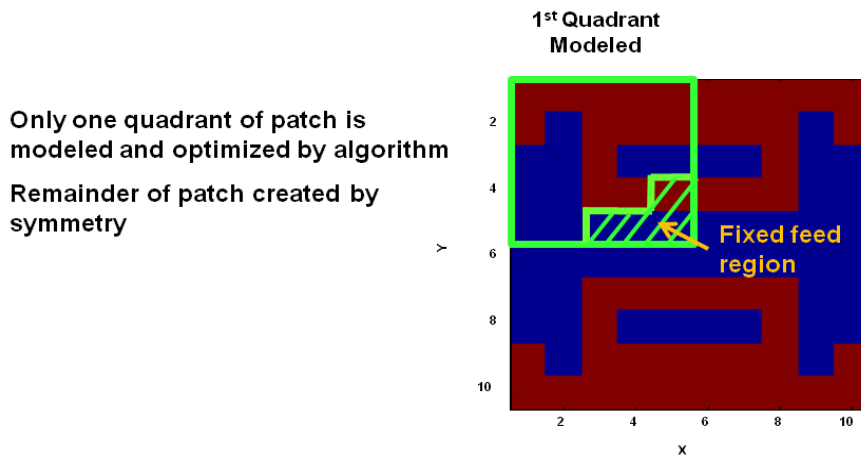


Figure 28. Design optimization utilizes only the upper left quadrant of the patch antenna element as shown above. Conducting pixels (metallization) are red.

The patch element is square and measures 30 mm along each side. We set the FDTD resolution such that each pixel spans four FDTD cells in both  $x$  and  $y$ . Since each pixel is three-by-three millimeters, each FDTD cell is 0.75 x 0.75 mm. We then represent the substrate and superstrate thicknesses in units of FDTD cells. For this experiment, the thickness of the first substrate layer  $d_0$  varied from 1 to 20 cells along the  $z$  axis. Since we were using a commercial substrate material of fixed size and permittivity for the second layer,  $d_1$  was set to a fixed value of two cells to correspond to that material’s thickness. The two superstrates represented by  $d_2$  and  $d_3$  varied in thickness from 1 to 20 and 1 to 6 cells, respectively.

For relative permittivity, we allowed  $\varepsilon_0$  (for the first substrate) to vary from 1 to 1.6 in discrete increments of 0.2. The relative permittivity of the second substrate,  $\varepsilon_1$ , was set to a constant value of 3.38 corresponding to the substrate used in manufacture. We allowed  $\varepsilon_2$  to vary from 1 to 2 in increments of 0.2;  $\varepsilon_3$  varied from 4 to 8 in 0.2 increments for a total of 21 discrete values.

EGO fully supports continuous parameters and there is no requirement to use the discrete values described here. However, for this experiment, we intend to compare EGO solutions with results obtained previously with the simple binary genetic algorithm, which does require discrete parameter representations. Since we intended to compare both convergence speed and best overall solution (i.e., lowest cost), it would have been an ambiguous comparison if we had allowed EGO to use continuous values while the GA could only choose from fixed discrete values. Therefore, we forced the output parameters from EGO to round-off to the nearest discrete value of each variable in the binary GA to allow for fair comparisons.

#### 4.3.4 The Cost Function

The cost function  $F$  computes the reflected power

$$F = \frac{\sqrt{\sum_i^I \sum_j^J |\Gamma(\theta_i, f_j)|^2}}{IJ} , \quad (25)$$

where  $\Gamma(\theta, f)$  denotes the active reflection coefficient at frequency  $f$  and scan angle  $\theta$ . The  $\theta_i$  are the sample scan angles and the  $f_j$  are the sample frequencies over the band. For the evaluation of  $\Gamma(\theta_i, f_j)$  we use a highly efficient finite difference time domain (FDTD) code [35], which produces the wideband response of the array for a specified range of scan angles. We compute the reflection coefficient  $\Gamma$  at three scan angles  $\theta_i$ : broadside;  $45^\circ$  in the  $x$ - $z$  plane; and  $45^\circ$  in the  $y$ - $z$  plane. The frequencies  $f_j$  at which  $\Gamma$  was evaluated consist of 1,001 equally-spaced points from 2 to 4.5 GHz. The source impedance  $Z_0$  was allowed to vary from 30 to 300  $\Omega$  in 10  $\Omega$  increments to find the lowest possible value of the cost function.

#### 4.3.5 Simple GA Optimization

The first optimization of the wideband fragmented patch antenna utilizes a simple GA. Figure 29 illustrates the encoding scheme used in the GA. There are a total of nine variables, including the thickness and permittivity characteristics of two substrates and two superstrates, and the fragmented patch itself. Since the substrate is fixed by commercial manufacturing constraints,  $d_1$  and  $\varepsilon_1$  are constant and are, therefore, not implemented within a chromosome.

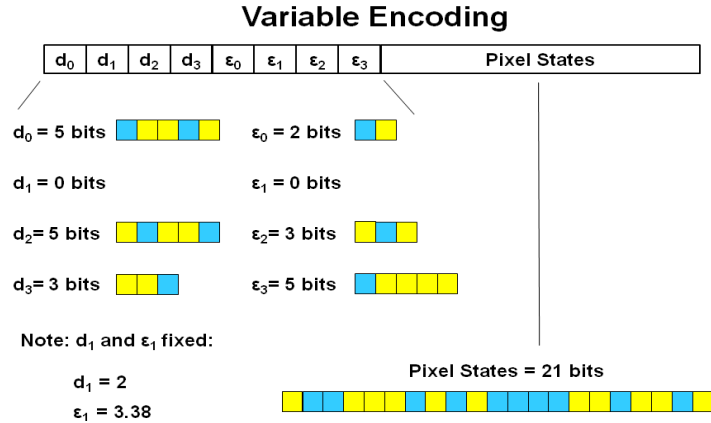


Figure 29. Fragmented patch parameter encoding scheme for the simple GA design optimization.

The number of bits used to represent each variable depends on the variable bounds and granularity that the designer imposes. For example, the thickness of the first substrate layer  $d_0$  varied from 1 to 20 FDTD cells along the  $z$  axis, thus requiring five bits to represent. The two superstrates were allowed to vary from 1 to 20 and 1 to 6 FDTD cells, respectively; thus,  $d_2$  and  $d_3$  utilized five and three bits, respectively. For the substrates and superstrates,  $\epsilon_0$  varied from 1 to 1.6 in discrete increments of 0.2; therefore only taking on four possible discrete values and requiring two bits, as shown. The first superstrate  $\epsilon_2$  varied from 1 to 2 in fixed increments of 0.2, thus having six possible discrete values and requiring three bits. Similarly,  $\epsilon_3$  varied from 4 to 8 in 0.2 increments, yielding 21 discrete values and requiring five bits. While the granularity of the thickness is set by the FDTD cell representation in the CEM code, the granularity of the permittivity is arbitrary. They are really continuous values which must be discretized for optimization using a binary GA.

Figure 30 shows a flow diagram of the GA that we used. First, an initial population of 12 random chromosomes was created. Random members of this population (selected with 10% probability) then had their patch configuration replaced with a pre-defined pixel geometry known as a “butterfly” patch, shown in Figure 31. Since this self-complimentary design worked well for an infinite array of patches in free space, this geometry was added to 10% of the population as an initial good “seed”. The simple GA did not converge well without this seed.

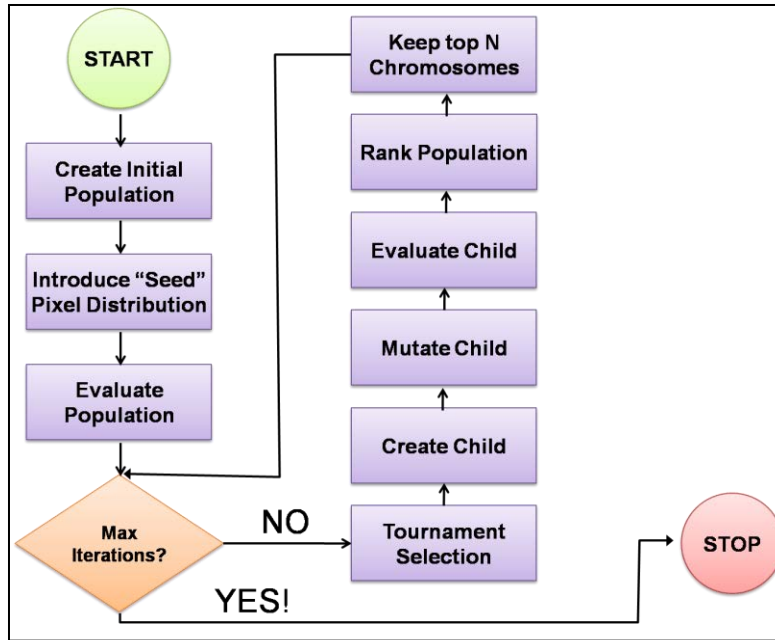


Figure 30. Flowchart for the simple GA design optimization algorithm.

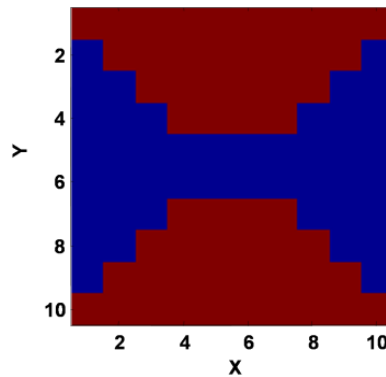


Figure 31. Pre-defined pixel geometry known as a “butterfly” patch. Conducting pixels are in red.

The population was evaluated with tournament selection, where 20% of the members were randomly chosen and the highest ranking individual selected as a parent. The process was then repeated to obtain the second parent, with uniqueness applied to the parent selection. The parents were then recombined with 3-point crossover, and the resulting child mutated. The initial mutation rate was 0.15, which then decreased by 0.995 each generation. This single child was then evaluated and added to the current population. The new population was ranked, with the top 12 chromosomes preserved and the worst solution discarded.

### 4.3.6 EGO Optimization

Figure 32 shows the encoding scheme developed for using EGO. There are two notable differences from the simple GA encoding. First, substrate and superstrate thickness and permittivity are encoded as real-valued continuous parameters. Note that only six parameters are shown; the parameters for the fixed manufacturing substrate are not included.

Since the parameters are represented as continuous values, post-processing is necessary in some cases when translating back into the physical antenna model. For example, the thickness of the sub- and superstrates can only have an integer number of FDTD cells; therefore, the real-valued parameters resulting from the EGO optimization must be rounded.

However, since there is a wide range of material permittivity available it is not necessary to round these parameters. Since we will compare this EGO implementation to the simple GA results, we post-process the permittivities to force them to the closest discrete value used in the binary GA. Note that this constraint will be removed for future optimizations since it is possible that the optimum permittivity value may lie between the discrete points.

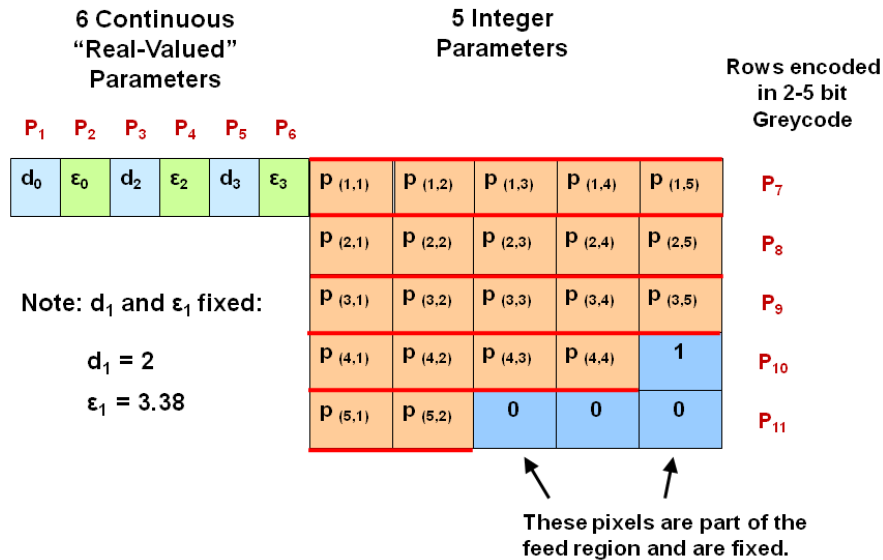


Figure 32. Encoding scheme for the EGO design optimization algorithm.

The second major difference is that the pixels in the patch have been grouped into rows and encoded into integers via a binary Gray code [36]. This reduces the number of patch “conducting vs non-conducting” pixel parameters from 21 (the # of pixels) to only five (the # of rows) and also reduces the problem

dimensionality without sacrificing accuracy. The Gray code is used to reduce the binary Hamming distance between consecutive integer representations, which provides for a smoother response surface.

This encoding scheme resulted in an 11-D function space. While EGO has been reportedly used with success on problems up to a dimensionality of 21, this 11-D function space was significantly larger than our earlier EGO optimizations, thus requiring the modifications to our EGO implementation discussed earlier in Section 4.2.5.

The initial population for EGO is created via a random Latin Hypercube distribution. Unlike simple GA optimization, there is no need to seed the initial population with a butterfly patch. Various population sizes were tested, with populations of 30 and 50 chromosomes yielding excellent solutions.

#### **4.3.7 Optimization Results for the Fragmented Patch Antenna**

While a statistical analysis would provide the best comparison between the simple GA and EGO, this has proven to be difficult to construct. The simple GA has been used to optimize this antenna configuration for several years with runtimes taking several days to weeks. However, only the very best solutions have been saved and many non-converging optimizations have been discarded. For EGO, more analysis is required to determine optimal initial population size. The long computation time required by the expensive cost function used to generate the electromagnetic solutions for the sample points makes it difficult to generate sufficient numbers of optimization runs for a meaningful statistical comparison of the techniques.

However, the EGO solution is better than the best solution achieved by the GA. The EGO solution also required significantly less computation time; EGO used only a third of the function evaluations required by the best GA runs.

In Figure 33 we compare the convergence characteristics of this EGO run to two “best” GA runs. There are two GA runs presented: one exhibits very good early convergence, while the other takes longer to develop a good solution but ultimately achieves a slightly better result. EGO achieves a superior solution than either GA run. The GA runs take up to three times more function evaluations.

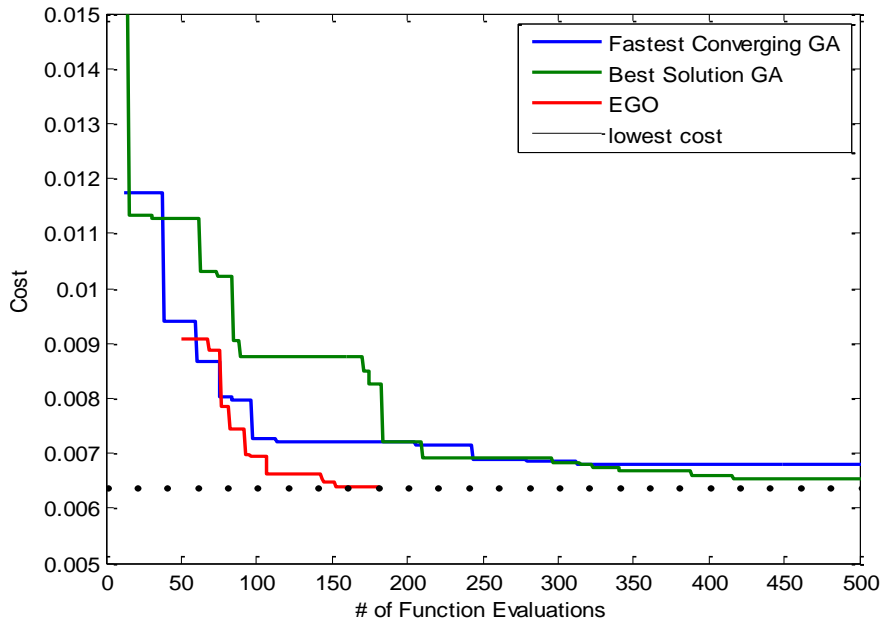


Figure 33. Convergence comparison of EGO with the best GA results.

Even more impressive is the fact that EGO discovered this solution without requiring seeding of the initial population with a known good solution, i.e., the butterfly patch. In Figure 34 we compare the patch configuration of the best EGO solution to the best GA solution and the butterfly patch. Even without seeding EGO converged to a very similar patch configuration.

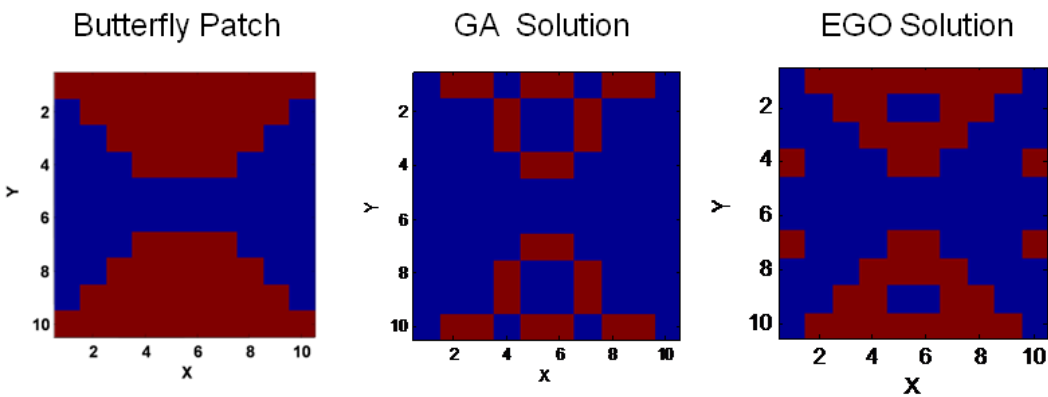


Figure 34. Comparison of GA and EGO patch solutions to the butterfly patch. Conducting pixels are shown in red.

In Table 11 we show the substrate and superstrate thicknesses and relative permittivity values resulting from the best GA and EGO runs. The results appear to indicate that the substrate characteristics are more important than the superstrate. The permittivity  $\epsilon_0$  and thickness  $d_0$  of the modifiable substrate are identical for the two best solutions. The two superstrate characteristics are close and show similar trends, but are not identical.

Table 11. Comparison of GA and EGO best results (\* Indicates a constant value).

Design Variable	GA	EGO
$\epsilon_0$	1.2	1.2
$\epsilon_1$	3.38*	3.38*
$\epsilon_2$	1.6	1.4
$\epsilon_3$	4.2	5.0
$d_0$ [FDTD cells]	20	20
$d_1$	2*	2*
$d_2$	10	13
$d_3$	4	3

In Figure 35 we compare the bandwidth characteristics of the two solutions. Since the cost functions for the two solutions were very close, it is not surprising that the bandwidth characteristics are also similar. Each has the widest bandwidth for the broadside scan (“bs”, blue curve) and the least bandwidth for the 45<sup>0</sup> degree y-z scan (“yz45”, red curve).

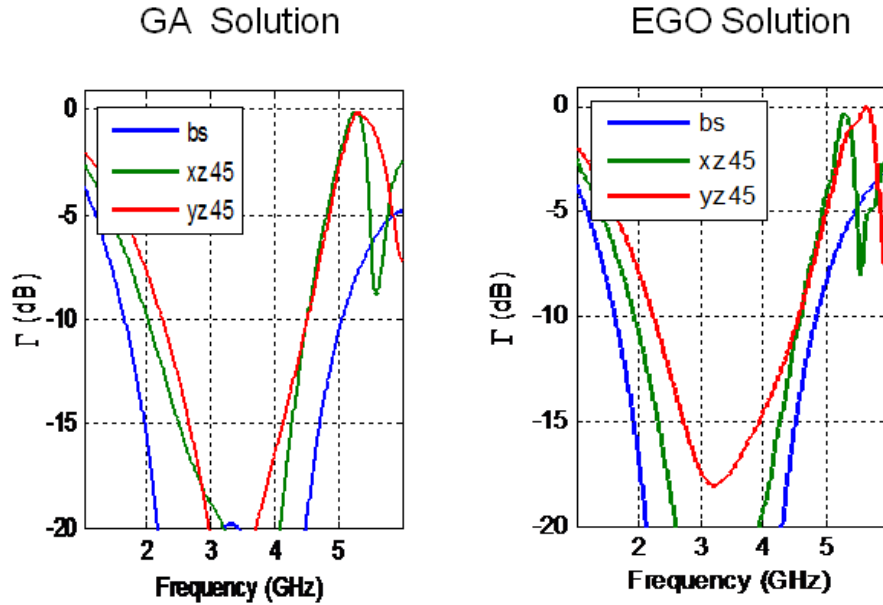


Figure 35 Wideband scan characteristics for the GA and EGO solutions.

#### 4.4 Endgame Techniques for EGO

The goal of EGO is to find the global minimum in the response surface using as few function evaluations as possible. As indicated in the last section, EGO typically requires far fewer cost function evaluations than GAs. However, both algorithms do not always drill down to the absolute minimum; therefore the addition of a final local search technique is indicated. In this section we introduce three “endgame” techniques which can improve optimization efficiency (fewer cost function evaluations) and, if required, can provide very accurate estimates of the global minimum. We also show results using a different cost function than the one previously used [2, 15].

Our effort in this section is to investigate strategies which force the algorithm into a local search mode in the endgame. Endgame techniques can keep the algorithm from searching wide areas of the function space (global search) and force the algorithm to focus on a smaller area (local search) [34]. The goal is to provide a more accurate estimate of the location of the global minimum in the response surface.

#### 4.4.1 Generalized Expected Improvement

Schonlau, et al, discuss an endgame technique which uses an integer-valued parameter  $g$  which can provide a systematic way of controlling the search mode [37]. The larger the value of  $g$  the more globally the algorithm will tend to search [37]. In this section we use  $g = 0, 1$ , and  $2$ . We used  $g = 1$  in our previous papers [2, 15]. Here we require both the  $g = 2$  case (a very global search) and the  $g = 0$  case (a very local search) in the optimization. We need the former to get us in the “basket,” i.e. near the global minimum, and we need  $g = 0$  to force local search in the endgame to provide a very accurate estimate of the value of the global minimum.

In EGO, the selection of the next point for evaluating the cost function requires a figure of merit called expected improvement [1]. The idea is to select the next sample point where the expected improvement is greatest [1]. As in Section 3.1.5, we introduced a new random variable  $Y(\mathbf{x})$  which models the uncertainty in the function’s value at the next sample point, i.e. at an untried  $\mathbf{x}$  [1, 37].  $Y(\mathbf{x})$  is assumed to be normally distributed with mean  $\hat{y}(\mathbf{x})$  and variance  $s^2(\mathbf{x})$ . The term  $[Y - \hat{y}]/s$  is a standard normal random variable.

Let  $f_{\min} = \min[y^{(1)} y^{(2)} \dots y^{(n)}]$  be the current function minimum. Improvement can be regarded as the improvement in  $f_{\min}$  at the next function evaluation, i.e. will  $Y$  be smaller than  $f_{\min}$ ? Improvement is defined to be equal to  $f_{\min} - Y(\mathbf{x})$  if  $Y(\mathbf{x}) < f_{\min}$  and 0 otherwise. The improvement is zero if  $Y$  is greater than the current minimum function value. The expected improvement is the expected value of the improvement and the generalized improvement is simply the improvement raised to the power  $g$ , i.e.:

$$I^g(\mathbf{x}) = [f_{\min} - Y(\mathbf{x})]^g \quad \text{if } Y(\mathbf{x}) < f_{\min} \quad (26)$$

$$I^g(\mathbf{x}) = 0 \quad \text{otherwise.}$$

$Y(\mathbf{x})$  is a Gaussian random process therefore  $I^g(\mathbf{x})$  is also a Gaussian random process. We are interested in the expected value of  $I^g(\mathbf{x})$ , the generalized expected improvement. For notational simplicity, we now suppress the  $\mathbf{x}$  dependence of  $Y(\mathbf{x})$ ,  $\hat{y}(\mathbf{x})$ ,  $I^g(\mathbf{x})$  and  $s(\mathbf{x})$  and introduce a variable,  $u$ , called normalized improvement. Normalized improvement is based on the DACE predictor  $\hat{y}$ ,  $s$  and  $f_{\min}$  and is given by:

$$u = (f_{\min} - \hat{y})/s. \quad (27)$$

If  $g = 0$ , the expected value of  $I^0$  yields the probability of improvement in  $f_{\min}$ , i.e. it is the probability that  $Y < f_{\min}$ , i.e.  $P(Y < f_{\min})$  [37] as shown below:

$$E[I^0] = P( Y < f_{\min} ) = P( (Y - \hat{y})/s < u ) = \Phi(u), \quad (g = 0) \quad (28)$$

where  $\Phi(u)$  is the cumulative distribution function of the standard normal probability density function denoted below by  $\varphi(u)$ . Recall that  $[Y - \hat{y}]/s$  is a standard normal random variable and  $u$  is the normalized improvement as defined in Equation 27.

Schonlau, et al [37] also give closed form expressions for the generalized expected improvement for  $g = 1$  and  $g = 2$  as follows:

$$E[I] = s[ u\Phi(u) + \varphi(u) ] \quad (g = 1) \quad (29)$$

$$E[I^2] = s^2[ (u^2+1) \Phi(u) + u\varphi(u) ]. \quad (g = 2) \quad (30)$$

The factor  $s^g$  in Equations 29 and 30 gives  $s$  more weight, i.e. it increases the value of the expected improvement for larger  $s$ . Since we seek the largest value for the expected improvement, the algorithm will search areas where  $s$  is larger. Since  $s$  is zero at the sample points and larger in regions away from the sample points, larger values of  $g$  tend to force the algorithm into a more global search [37]. Another way of looking at this is that there is a tradeoff between small improvements with high probability (local search) and large improvements with lower probability (global search) [37]. For larger values of the integer  $g$ , larger improvements become more important, even if they have a lower probability of occurring, and the search tends to be more global.

Since we have closed form expressions for  $E[I^g]$ , we can use  $\hat{y}$ ,  $s$ , and  $u$  to evaluate  $E[I^g]$  over  $\mathbf{x}$  (recall that  $\hat{y}$ ,  $s$ ,  $u$ , and  $E[I^g]$  are all functions of  $\mathbf{x}$ ) to find the value of  $\mathbf{x}$  where the expected improvement is maximized. We then evaluate the cost function at  $\mathbf{x}$ , i.e. we run the complex, expensive computer code to obtain a new sample point. The calculation of  $\hat{y}$ ,  $s$ , and  $u$  are easy and very fast, therefore we can evaluate  $E[I^g]$  at a large number of  $\mathbf{x}$  vectors and find the maximum.

Expected improvement also provides a simple and effective stopping criterion. For  $g = 0$  and 1, we base the stopping criterion on  $E[I^0]$  or  $E[I]$ , i.e. if the *maximum* of  $E[I^0]$ , or  $E[I]$  is smaller than a pre-specified tolerance, we stop the algorithm [37]. For  $g > 1$ , we compare the *maximum* of  $\{E[I^g]\}^{1/g}$  with the tolerance. Schonlau, et. al. show that  $\{E[I^g]\}^{1/g} > E[I]$ , therefore for the same tolerance, the stopping criterion based on  $\{E[I^g]\}^{1/g}$  will tend to sample more points, be more conservative and search more globally [37]. For  $g > 1$  the search tends to be very global and for  $g < 1$  the search tends to be very local. If the criterion is not met, the selected point,  $\mathbf{x}$ , is added as a new data point and  $n$  is increased by one. The algorithm proceeds by estimating new values for the correlation parameters using the new data set, which includes all of the previous data points plus the new one. The algorithm iterates until the stopping criterion is met.

#### 4.4.2 An Ad Hoc Approach

We can also control the nature of the search in a more ad hoc way [37, 38] by artificially increasing the MSE of prediction ( $s^2$ ) for a more global search or decreasing it for a more local search. Recall that for larger  $g$ , the algorithm tends to be more conservative and sample in a more global manner. Conversely, if we decrease the standard error of prediction we should be able to force the algorithm into a more local search. We show in Section 4.4.4.1 that multiplying  $s^2$  by a factor which decreases linearly as a function of the iteration number gives good results.

#### 4.4.3 Engineer-in-the-Loop Approach

Within a few iterations the DACE predictor can produce a good approximation to the true cost function surface. At this point it could be useful to have an engineer examine the results and suggest a subset of the function space for more fruitful search. The visualization aspect of EGO is discussed in [1] and is another way to implement an endgame technique, i.e., a narrowing of the search space based on engineering judgment. We used this technique to optimize the design of a different antenna array. In this case the elements of the PSA are different from the elements in Figure 7a.

#### 4.4.4 Results Using Endgame Techniques

The upper valley region near the global minimum (red dot) in Figure 7b has an extremely flat floor. For example, along the valley for separations from 8 to 10 mm the directivity changes by less than 0.05% while the separation changes by 25% and the frequency changes by 1%. This makes it *extremely* difficult for optimization algorithms to refine estimates in that region and obtain values for separation and frequency which give the absolute maximum directivity. This is also motivation for implementing an endgame technique for EGO.

A Latin hypercube technique [11] is used to select 21 initial data points. We limit the total number of iterations (with one cost function evaluation per iteration) to 101, i.e. 80 iterations after the initial data set. Since the actual cost function is very expensive, we are not interested in performing more than about 100 function evaluations. This is done by setting the stopping criterion to 0.000001 % of  $f_{\min}$  which ensures that the algorithm will never stop before we end at 80 iterations. We then select  $f_{\min}$  after 101 iterations as the best estimate of the global minimum.

We present results for the two endgame techniques which involve modification of the EGO algorithm. The third technique does not involve modification of the EGO algorithm but utilizes the output of the DACE predictor for visualization of the response surface. This enables engineering judgments to be made.

#### 4.4.4.1 Results Using Modifications to the EGO Algorithm

The first technique is an extension of the generalized expected improvement theory presented by Schonlau, et.al. [37], therefore we refer to it as the “g parameter technique.” For  $g = 2$  the search is very global; for  $g=1$  the search is balanced; and for  $g = 0$  the search is very local. We experimented by changing  $g$  from 2 to 1 to 0 as the algorithm iterates to show that this is the case. Our baseline case used  $g = 2$  for iterations 1 to 20;  $g = 1$  for iterations 21 to 70 and  $g = 0$  for iterations 71 to 80. We discovered a problem: the algorithm can go into local search mode ( $g = 0$ ) in the valley a short distance from the true global minimum and never leave the area. As a result, a less accurate estimate of the global minimum is obtained.

The second technique is an ad hoc method described by Mockus, et.al. [38]. The mean squared error is multiplied by a factor we call “ratio” (the notation  $\alpha$  is used in [38]). Initially, “ratio” is larger to make the algorithm search more globally and then “ratio” becomes smaller as the algorithm iterates to make the algorithm search more locally. We call this technique “linearly decreasing ratio” since our experiments used a linearly decreasing value for “ratio” from 1 at iteration number one to 0.01 at iteration 80.

Our goal is to estimate the true global minimum with an accuracy of better than 0.1 % in a reasonable number of cost function evaluations. In Table 12 we show the accuracy (in % difference from the true global minimum) for both endgame techniques. EGO with both techniques found an estimate of the global minimum of -10.263 dB, which is slightly better than the one found using “exhaustive” search. Our “exhaustive” search was unable to find a minimum better than -10.262 dB.

For the “linearly decreasing ratio” endgame technique the average value of the difference between the estimate and the true global minimum is 0.066% with a standard deviation of .05%. The goal of less than 0.1 % accuracy has therefore been achieved. The standard deviation is extremely small which we discuss when we compare the EGO results with GAs in Section 4.2. We do not present comparable data for the “g parameter” technique since there are larger inaccuracies in the estimates (basically outliers at 0.816% and 0.35%) when the algorithm gets trapped out in the valley.

Table 12. Estimates of the directivity (dB) and the % difference are shown for both EGO endgame techniques for 10 typical runs. For the linearly decreasing ratio technique the average difference is 0.066% and the standard deviation is only 0.05%.

Directivity (dB)	(% Difference)	Directivity (dB)	(% Difference)
10.263	0.000	10.258	0.110
10.259	0.080	10.263	0.000
10.258	0.110	10.258	0.110
10.255	0.180	10.228	0.816
10.261	0.047	10.261	0.047
10.259	0.080	10.258	0.110
10.262	0.028	10.248	0.350
10.260	0.057	10.261	0.047
10.262	0.019	10.263	0.000
10.260	0.056	10.258	0.110

**Linearly decreasing ratio technique**

**g parameter technique**

To illustrate the convergence of these two endgame techniques we show in Figures 36, 37, and 38 the following: (a) all 101 data points on the Test131 function space (with the final estimate shown as a black dot); (b) the DACE predictor after 101 iterations; (c) the standard error (standard error of prediction) after 101 iterations; and (d) the generalized expected improvement after 101 iterations. In Figure 36a we show how the EGO algorithm with the “g parameter” endgame technique can get trapped away from the global minimum. Note that the DACE predictor in Figure 36b is a poor approximation of the true cost function shown by the contour lines in Figure 36a. This is related to the fact that the correlation coefficients become very large when the algorithm goes into a strong local search mode (note the concentration of data points out in the valley). In Figure 37 we show the case where the “g parameter” endgame technique found the “exact” global minimum. Again, the DACE predictor is a poor approximation to the true response surface after the algorithm has gone into a local search mode.

In Figure 38 we show results using the “linearly decreasing ratio” endgame technique where the algorithm has also gone into a local search mode resulting in an estimation accuracy of 0.08%. In Figure 39 we show the correlation parameters  $\theta_1$  and  $\theta_2$  as a function of the iteration number for the case in Figure 38. There is a clear indication of local search as the values for  $\theta_1$  and  $\theta_2$  become large and the

corresponding radial basis function approximation to the response surface [15] consists of very narrow Gaussian functions. Again, the result is a large loss in prediction fidelity elsewhere in the function space. This is not a concern if one is interested in the design point (separation and operating frequency) which yields the best directivity. In practice, one is typically interested in the “best” design point, i.e. the optimum design.

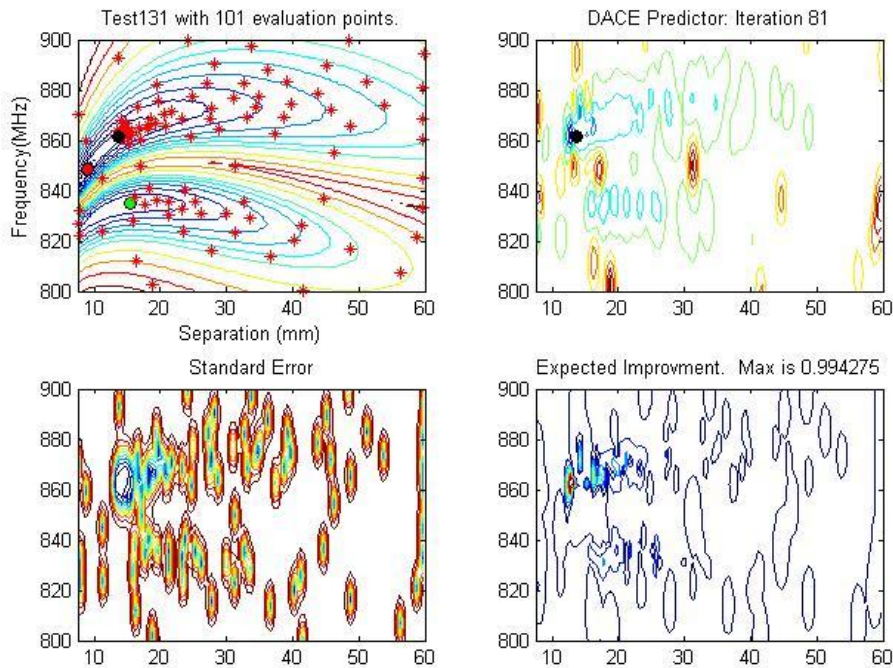


Figure 36. Results for the “g parameter” endgame technique where the algorithm is trapped in the flat valley. (a) Test131 function space with 101 data points (b) DACE predictor (c) Standard error (d) Expected improvement.

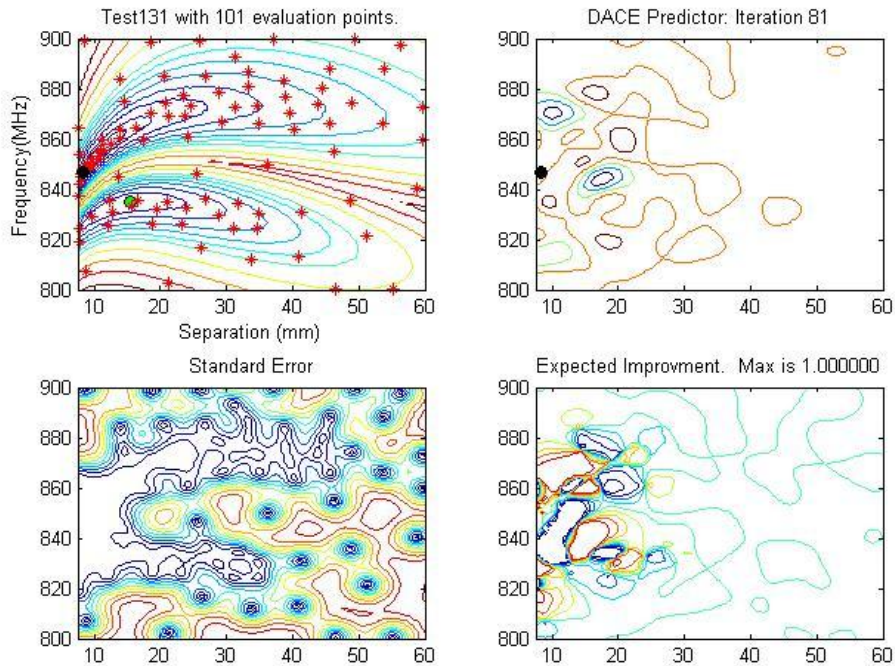


Figure 37. Results for the “g parameter” endgame technique where the “exact” global minimum was found to be -10.263 dB at  $\mathbf{x} = [8.37\text{mm} \ 847 \text{MHz}]$ .

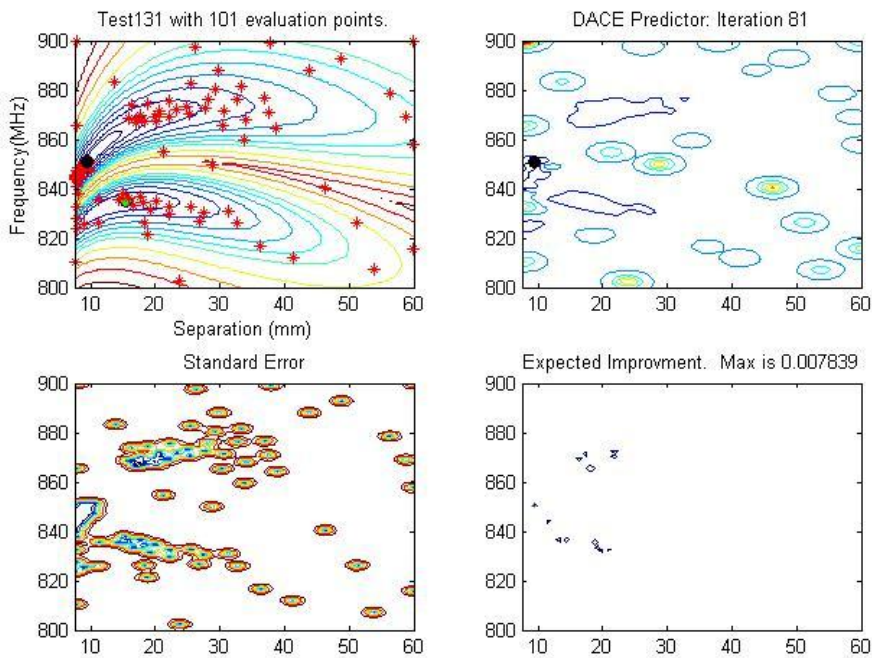


Figure 38. Results for the “linearly decreasing ratio” endgame technique which resulted in a very local search. The final difference from the true global minimum was 0.08% at  $\mathbf{x} = [9.4\text{mm} \ 851 \text{MHz}]$ .

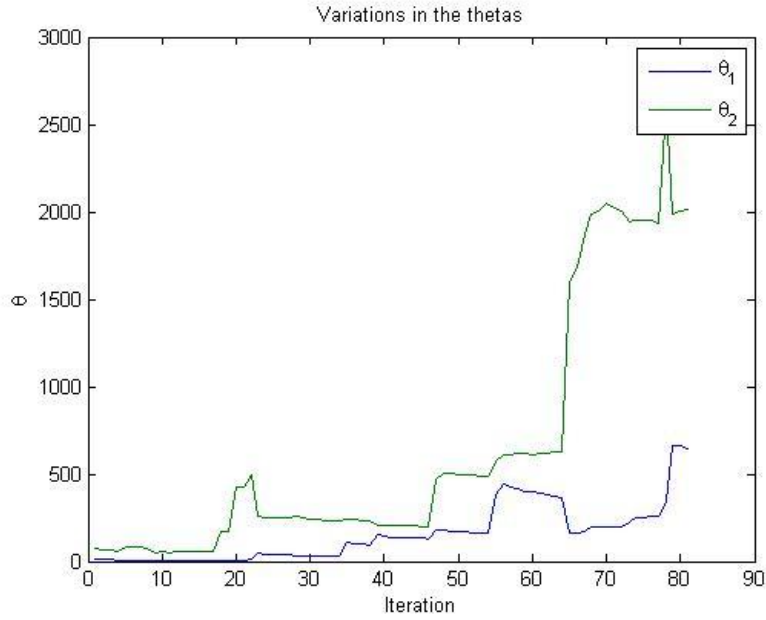


Figure 39. Correlation coefficients  $\theta_1$  and  $\theta_2$  as a function of the iteration number.

#### 4.4.4.2 Results Using the Engineer-in-the-Loop Technique

The **Test55** array geometry is shown in Figure 40a. A subset of the function space is shown in Figure 40b where the global minimum (red dot) is located in a very flat, steep-sided, long, narrow valley.

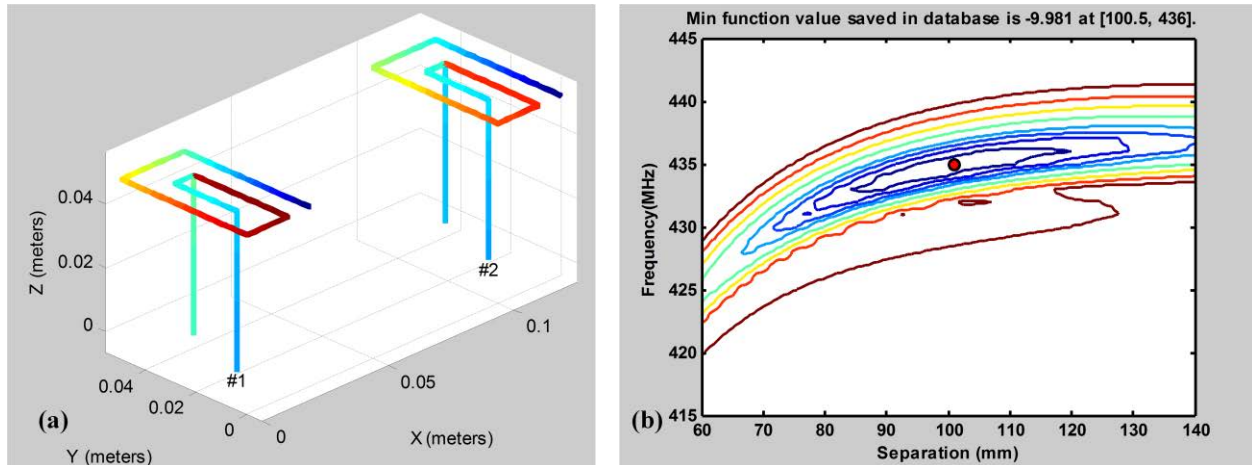


Figure 40. (a) Test55 PSA geometry. The colors represent current density magnitude. (b) The response surface represented as directivity contours. The red dot is the global minimum where the negative of the directivity is -9.981 dB at  $x = [100.5\text{mm } 435\text{ MHz}]$  found using exhaustive search.

The search was much larger than the portion shown in Figure 40b and extended from 350 to 500 MHz and from 50 to 140 mm. For this initial search space, we ran EGO for 15 iterations after the initial set of 21 data samples. The antenna engineer then reviewed the DACE approximation to the response surface and, using a cautious approach, narrowed the search space in two stages. The first stage narrowed the search to the box 430 to 450 MHz and 90 to 120 mm. Again, using the DACE predictor, the engineer selected the box 432 to 437 MHz and 95 to 105 mm. In the first stage 21 initial points and 5 iterations were used and in the final (second) stage 11 initial points and 5 iterations were used. The total number of function evaluations was 78. EGO estimated a global minimum of -9.991 dB at  $\mathbf{x} = [99.45\text{mm } 434.74\text{ MHz}]$ . An exhaustive search estimated -9.981 dB at  $\mathbf{x} = [100.5\text{mm } 435\text{ MHz}]$ . Again, EGO found a slightly better directivity than an exhaustive search.

#### 4.4.5 Observations on Endgame Techniques

We implemented three endgame techniques which tend to make the EGO algorithm perform local search in the endgame and can result in very accurate estimates of the global minimum. In fact, the techniques found a value which was slightly better than the one found using an “exhaustive” search. However, one technique, the “g parameter” technique, can get trapped in a strong local search mode in flat valleys and never leave the area. The “linearly decreasing ratio” endgame technique never encountered this difficulty and is therefore the preferred technique. The remarkable thing about EGO with endgame techniques (and also our original version of EGO [2, 15]) is that it never gets stuck in the local minimum which corresponds to the director lobe. EGO always finds an estimate for the global minimum for our problem, i.e. the reflector lobe of the Test131 PSA.

In Section 4.4.6 we compare EGO results using the “linearly decreasing ratio” endgame technique with the performance of GAs and with EGO with no endgame technique. The results are encouraging, especially regarding the very small deviation in performance for different initial sample sets. This means that EGO is more likely to find a very accurate estimate of the global minimum given an arbitrary set of initial sample points.

It is evident in Figure 39 that the correlation parameters give an indication of local search. The initial values of the parameters are determined by the initial set of sample points; however, as new samples are selected and added, the correlation parameters change. The approximating radial basis functions [2] become very narrow (spiked) as the algorithm gets close to the answer and  $\theta_1$  and  $\theta_2$  become very large. In the end, we lose DACE “prediction accuracy” and the Nelder-Mead downhill simplex method for finding maximum likelihood estimates of  $\theta_1$  and  $\theta_2$  can collapse. The sample points become so closely spaced that the correlation matrix is severely ill-conditioned. This problem is inherent in stochastic modeling techniques as discussed by Mockus, [38, page 122]. At this point, they state that “... it is quite reasonable to use the quadratic approximation approach which is widely used in some very efficient methods of local optimization.” The good news is that the EGO estimate of the global minimum is already very good at this point. We have not investigated a switch to another class of local optimization since EGO (modified with an endgame technique) works so well; however, it could be warranted if extremely accurate estimates of the global minimum are required.

#### 4.4.6 Comparison of Results Using EGO (with Endgame) and the GA

In Figure 41 we compare EGO using the “linearly decreasing ratio” endgame technique with our previous implementation of EGO (without endgame) and with two versions of a continuous parameter GA which were described in detail in [15]. EGO outperforms the GAs, i.e. the EGO algorithm gives more accurate estimates of the true global minimum with fewer function evaluations. The enhanced GA, however, performs well (even for less than 100 function evaluations) and is comparable to EGO (without an endgame technique). The “linearly decreasing ratio” endgame technique EGO exceeded the goal of 0.1% deviation from the true global minimum and the variation over 10 typical runs (with a different set of initial samples for each run) is significantly smaller than for either the enhanced GA or the EGO algorithm without an endgame technique. This is shown by the size of the error bar on the black circle representing the results for EGO with the “linearly decreasing ratio” endgame technique.

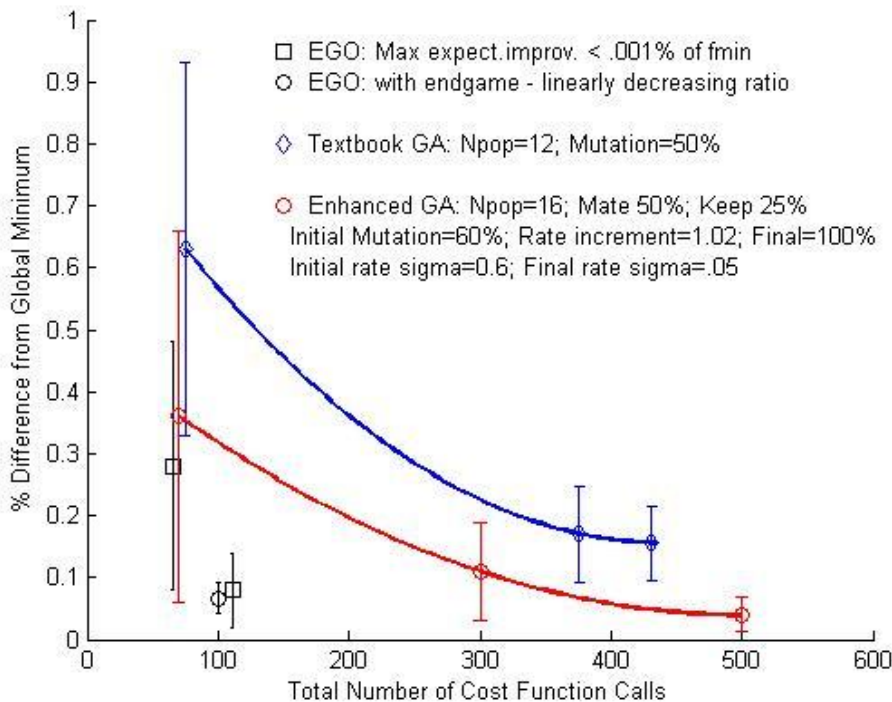


Figure 41. Comparison of results obtained for the Test131 PSA geometry using the different optimization algorithms [15]. The error bars represent slightly different estimates of the global minimum for different sets of initial data samples.

## 4.5 Orthogonal-Maximin Latin Hypercube Design (OMLHD) Initial Data Sets

### 4.5.1 Introduction

As discussed in Section 4.1, antenna design optimization problems using complex computational electromagnetic (CEM) codes can require considerable computational resources [2, 15]. The cost function can therefore be very expensive. Additionally, the output of a CEM code is deterministic and not subject to random variations. For such problems, we have successfully used a technique called EGO [1, 2 and 15].

The output of the CEM code for any combination of design variables (input parameters) is simply the response surface (cost function) of the optimization problem. EGO uses a stochastic model to fit the surface and to select the next design point in the input space in the search for the global optimum [1, 2 and 15]. Such techniques are also referred to as kriging in the literature [1, 39]. Joseph and Hung [39] have suggested that the performance of such techniques may be improved by using a modified Latin Hypercube technique to establish an initial data sampling of the response surface over the input space of design variables.

Design optimization using evolutionary computation (EC) techniques such as GAs or EGO require an initial set of data samples (design points) on the order of ten times the dimensionality. These are obtained by evaluating the cost function at selected sites in the input space. A 2-D input space can be sampled using a Latin square, a statistical sampling technique which samples a square grid such that there is only a single sample in any given row and column. The Latin hypercube is a generalization to an arbitrary number of dimensions. However, a standard random Latin hypercube can result in initial data sets which may be highly correlated and may not have good space-filling properties. There are techniques which address this issue and we apply one of these techniques to EGO and to a GA.

We have used standard Latin hypercube data sampling in EGO for six and 11 design variables and found acceptable designs using far fewer cost function evaluations than using a GA (see Section 4.4.6). We will now investigate some simple design optimization problems and show where OMLHD may be helpful and where it is not particularly useful [46].

### 4.5.2 Generating OMLHD Data Sets

Consider a 2-D design optimization problem with dimensionality  $k = 2$ . It has been suggested that  $n = 11k-1 = 21$  initial computer experiments be performed to sample the 2-D input space [1]. Each experiment produces a single value of the cost function, i.e. a single, scalar value of the response surface. In our previous research we have used a standard, random Latin hypercube design (LHD) which we designate as LHD (21, 2) for  $n = 21$  and  $k = 2$ . The algorithm for generating a standard, random LHD ( $n, k$ ) is well known and relatively easy to implement [11, 40]. We developed a MATLAB version to generate standard, random LHD (21, 2) designs. For example, consider an LHD (5, 2) design matrix,  $\mathbf{D}$ , shown below:

$$\mathbf{D} = \begin{bmatrix} 4 & 1 \\ 2 & 2 \\ 1 & 5 \\ 5 & 3 \\ 3 & 4 \end{bmatrix}.$$

Each of the two factors is divided into five possible values with each value given a “bin number” from 1 to 5. Each row represents a 2-D vector (design point or experiment) in the input space. Each column contains the  $n$  bins for one of the factors (design variable). In the exchange rule below Equation 37, we denote an element in a design matrix as  $x_{ij}$ .

The number of possible LHDs is  $(n!)^k$ , which is 14,400 for this very simple example. As discussed in [39] a random LHD is not necessarily a “good” design. In general, we want the distribution of one factor in the input space to be as uncorrelated as possible with the distribution of the other factor(s). This means that the pair wise correlation between columns should be small. If they were highly correlated we may not be able to distinguish between the effects of the two factors [39]. Also, we want to spread the points across the experimental region (input space) as much as possible. Minimizing pair wise correlation and maximizing inter-site distances are good criteria but there is not a one-to-one relationship between the two [39]. Joseph and Hung [39] propose a multi-objective criterion that minimizes pair wise correlations and also maximizes the minimum inter-site distance.

We now discuss the performance measures used to quantitatively implement the joint correlation/distance objective function. Following [41], Joseph and Hung [39] introduce the following correlation performance measure:

$$\rho^2 = \frac{1}{\frac{k(k-1)}{2} \sum_{i=2}^k \sum_{j=1}^{i-1} \rho_{ij}^2}, \quad (31)$$

where  $\rho_{ij}$  is the linear correlation between columns  $i$  and  $j$  and is calculated using the MATLAB function *corrcoef*. Suppose that each factor takes values in  $\{1, 2, 3, \dots, n\}$  as in  $\mathbf{D}$  above. The total number of inter-site distances between the  $n$  vectors (rows) is  $m = n!/\{2!(n-2)!\}$ , since we are taking  $n$  distinct objects two at a time. Let  $\mathbf{s}$  and  $\mathbf{t}$  be vectors at two sites (design points). The rectangular distance is defined as  $d(\mathbf{s}, \mathbf{t}) = \sum_{j=1}^k |s_j - t_j|$  and the rectangular distances are  $\{d_1, d_2, d_3, \dots, d_m\}$ . The inter-site distance performance measure is:

$$\phi_p = \left( \sum_{i=1}^m \frac{1}{d_i^p} \right)^{\frac{1}{p}}, \quad (32)$$

where  $p$  is a large positive integer. Following Joseph and Hung [39] we use  $p = 15$ . We now combine the two measures, therefore, we must scale  $\phi_p$  using upper and lower bounds for  $\phi_p$  such that the scaled variable,  $\Phi_p$ , is in the range  $0 \leq \Phi_p \leq 1$  (since  $0 \leq \rho^2 \leq 1$ ). Details for calculating upper and lower bounds are described in [39]. The combined, or multi-objective, function is:

$$\Psi_p = w\rho^2 + (1-w)\Phi_p, \quad (33)$$

where  $0 \leq w \leq 1$  and  $w = 0.5$  weights both measures equally. An LHD that minimizes  $\Psi_p$  is called an orthogonal-maximin LHD or OMLHD. Maximin refers to the fact that the distance measure maximizes the minimum inter-site distance.

Joseph and Hung [39] developed an interesting new algorithm which can minimize  $\Psi_p$  and therefore find an OMLHD (n, k) which can be used to select a “good” initial data set for design optimization techniques. It uses a version of the Metropolis, or simulated annealing, algorithm which is described well in [42]. The algorithm starts with an initial design  $\mathbf{D}$  found from a standard LHD. The algorithm then iterates through a sequence of new designs where each new design,  $\mathbf{D}_{try}$ , is a perturbation of the previous design. The perturbation is an element exchange, i.e. one element changes places with another element within  $\mathbf{D}$ . We replace  $\mathbf{D}$  with  $\mathbf{D}_{try}$  if  $\Psi_p$  is reduced. Otherwise  $\mathbf{D}$  is replaced with  $\mathbf{D}_{try}$  with probability:

$$\pi = \exp \left[ -\frac{\{\Psi_p(\mathbf{D}_{try}) - \Psi_p(\mathbf{D})\}}{t} \right], \quad (34)$$

where  $t$  is a parameter called “temperature” which is initialized at a starting value  $t_0$ . The probability of replacement is larger for higher temperatures. If  $\mathbf{D}_{try}$  is not better than the best design found so far,  $\mathbf{D}_{best}$ , after a large number of iterations the temperature is decreased. Typically the large number is chosen to be equal to  $I_{max} = 10 \times m \times k$ , where  $m$  is the total number of inter-site distances and  $k$  is the dimensionality [42]. The temperature is reduced by a constant factor  $FACT$  (typically 0.9 to 0.95) and the algorithm continues to iterate. The temperature is higher at the beginning of the search so that the algorithm can kick out of local minima.

The key to the new algorithm is in the exchange technique for producing a perturbed design. The element to be exchanged is chosen judiciously in order to improve (reduce)  $\Psi_p$  [39]. First define a correlation performance measure **for each column (i.e. factor)**  $l = 1, 2, \dots, k$ :

$$\rho_l^2 = \frac{1}{k-1} \sum_{j \neq l} \rho_{jl}^2 \quad (35)$$

where  $\rho_{jl}$  is the correlation between column  $j$  and column  $l$ . Next, define a distance measure **for each row (i.e. data point)**  $i = 1, 2, \dots, n$ :

$$\phi_{pi} = \left( \sum_{j \neq i} \frac{1}{d_{ij}^p} \right)^{\frac{1}{p}}, \quad (36)$$

where  $d_{ij}$  is the rectangular distance between rows  $i$  and  $j$ . The two measures can be used to select a column (factor) which is highly correlated with the other columns (factors) and a row (data point) which is close to the other rows (data points). The row and column are selected as follows [39]:

$$i^* = \arg \max_i \{\phi_{pi}\} \quad \text{and} \quad l^* = \arg \max_l \{\rho_l^2\} \quad (37)$$

This pair of indices  $(i^*, l^*)$  locates a single element,  $x_{i^*l^*}$ , within  $\mathbf{D}$ . The exchange rule is then [39]:

*Exchange  $x_{i^*l^*}$  with a randomly chosen element in column  $l^*$ .*

This gives us a good opportunity for improving (reducing) the multi-objective cost function  $\Psi_p$  [39]. We implemented the OMLHD algorithm (including the simulated annealing part) in MATLAB to produce OMLHD  $(n, k)$  design matrices. We were able to reproduce the results in Table 1 in Reference 39 using the parameters  $t_0 = 100$ ,  $p = 15$ ,  $\text{FAcT} = .90$  and  $w = 0.5$ . The results are shown in the Appendix. We also show a similar OMLHD  $(5, 3)$  design which has identical performance measures. This is not that surprising since there are 1,728,000 possible LHDs for  $n = 5$  and  $k = 3$ .

### 4.5.3 A Two-Dimensional Problem Using EGO

We consider an optimization problem with two variables to show the effect of selecting an initial data set using OMLHD  $(21, 2)$  designs versus standard LHD designs. The cost function or response surface shown in Figure 42 is obtained from the following function [14] which we call the “egg crate function”:

$$f(x, y) = x \sin(4x) + 1.1y \sin(2y). \quad (38)$$

The input space is:  $0 \leq x \leq 10$  and  $0 \leq y \leq 10$ . We seek the single global minimum of -18.55 at  $(x, y) = (9.035, 8.66)$  using the EGO algorithm [1, 2 and 15]. The response surface is complicated and has 17 local minima in the input space.

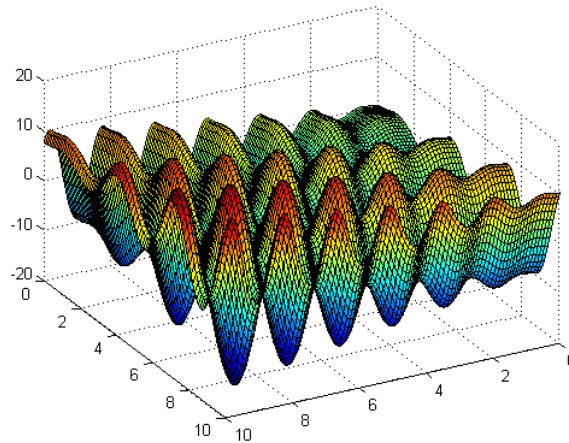


Figure 42. The “egg crate function” from Haupt and Haupt [14]

Although EGO is most useful in design optimization with expensive cost functions and the function in Equation (38) is not expensive to evaluate, we introduce EGO here since we will use this algorithm in Section 4.5.5 for a 4-D design optimization coupled with a CEM code [43].

Again, very briefly, EGO performs both global and local searches simultaneously in order to fully explore the function space and avoid becoming trapped in local minima [1, 2]. Unlike a GA, EGO creates a model of the response surface. The model is refined throughout the search and is used to predict areas of the function space which warrant further exploration, either because they are close to known good areas (i.e. local search) or because they have been insufficiently explored and therefore exhibit high uncertainty (i.e. global search). In a single iteration only a single design point is evaluated. In a GA, most of the time is spent evaluating numerous proposed solutions; the EGO algorithm spends computation time refining the response surface model. This reduces the number of expensive cost function evaluations.

Previously [1, 2, 43, and 44] we have used the response surface model and a parameter called expected improvement [1, 2] to select the next point in the input space as the next design to evaluate. The next design point is selected where the value of a quantity called the expected improvement is maximized. Algorithm convergence is declared if the expected improvement at that point is less than a percentage (typically 0.1 %) of the current best function value,  $f_{\min}$  [1, 2 and 15]. One can also simply terminate the algorithm after a set number of iterations and select the design which has the lowest cost. Both approaches are somewhat arbitrary and ad hoc and can lead to problems with local minima. However, we believe

that with evolutionary optimization techniques in engineering design it is more realistic, and practical, to terminate algorithms based on a performance criterion, i.e. where the current value of the cost function satisfies certain design specifications. This replaces artificial, and often arbitrary, criteria and you get an acceptable design [13].

For the analytic egg crate function we have no design specification (such as keeping the voltage standing wave ratio below a certain value over the frequency band) but we do know the exact value of the global minimum. We therefore impose a “performance spec” by declaring convergence if any cost function for any proposed design is within 7.5 % of the global minimum of -18.55, i.e. if any value is below -17.16 we declare convergence and take that design point as the optimum design. Since the value of the deepest local minimum is -16.97 we will never select a local minimum. This is important since we will perform many EGO runs to get statistical data to compare the performance of an OMLHD initial data set with standard LHD data sets.

For  $n=21$  and  $k=2$ , there are  $2.61028437 \times 10^{39}$  possible LHDs. Using our MATLAB version of the Joseph and Hung algorithm [39] for generating OMLHD (21, 2) designs, we produced a design  $\mathbf{D}_{\text{test2}}$  with performance measures  $\rho = 0.0091$ ,  $\phi_p = 0.1318$  and  $\Psi_p = 0.2333$ . The correlation parameter is very low and the design points are also spread quite well over the input space as shown in Figure 43 where this OMLHD (21, 2) design is compared with a typical LHD (21, 2) design.

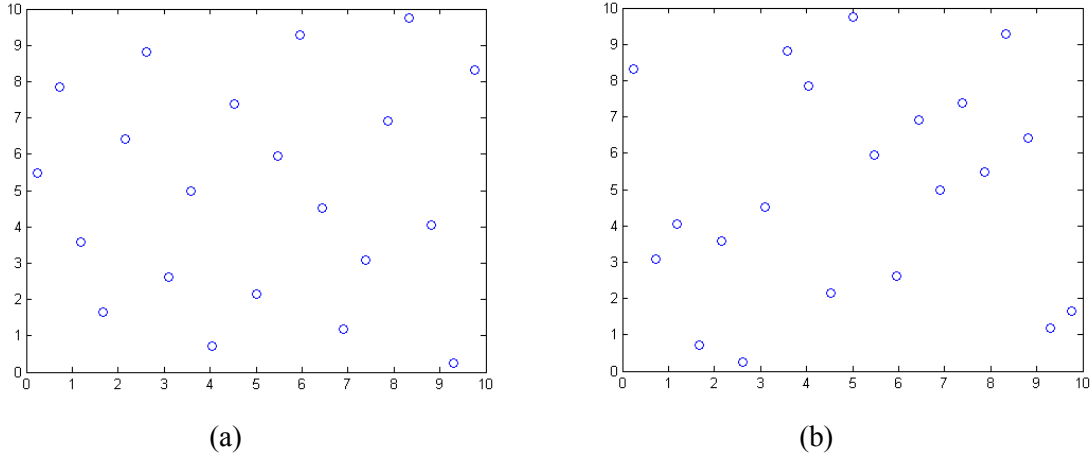


Figure 43. (a) OMLHD (21, 2) design  $\mathbf{D}_{\text{test}2}$  (b) Typical LHD (21, 2) design.

We compare the performance of the OMLHD design with an LHD design by plotting the frequency of the number of cost function evaluations to reach convergence versus the number of cost function evaluations. The EGO algorithm was run 54 times for each type of initial data set. Recall that convergence is declared whenever the current best value of the cost function,  $f_{\min}$ , is within 7.5 % of the actual global minimum. Results are shown in Figure 44 below [46].

The OMLHD design uses the same initial data set ( $\mathbf{D}_{\text{test}2}$ ) for every run. The only variation from run to run is due to the fact that we randomize (by a very small amount) the spacing of the search grid used to find the point of maximum expected improvement. Otherwise, we would get the same answer every time. In the iterative process for EGO, the point of maximum expected improvement becomes the next design point. For the LHD designs, in addition to the randomized search grid spacing, we use a different LHD design for every run [46].

Convergence for the OMLHD design takes slightly more than 60 function evaluations on a consistent basis (see Figure 44). In fact there are many more cases of rapid convergence (fewer cost function evaluations) for the LHD initial data sets. In these cases, the LHD initial data sets contain sample points close to the global minimum resulting in faster convergence (fewer cost function evaluations). Therefore, there appears to be no significant advantage in using the OMLHD initial data set to start the EGO algorithm for this 2-D design problem. In the next section, we investigate a 4-D design optimization problem.

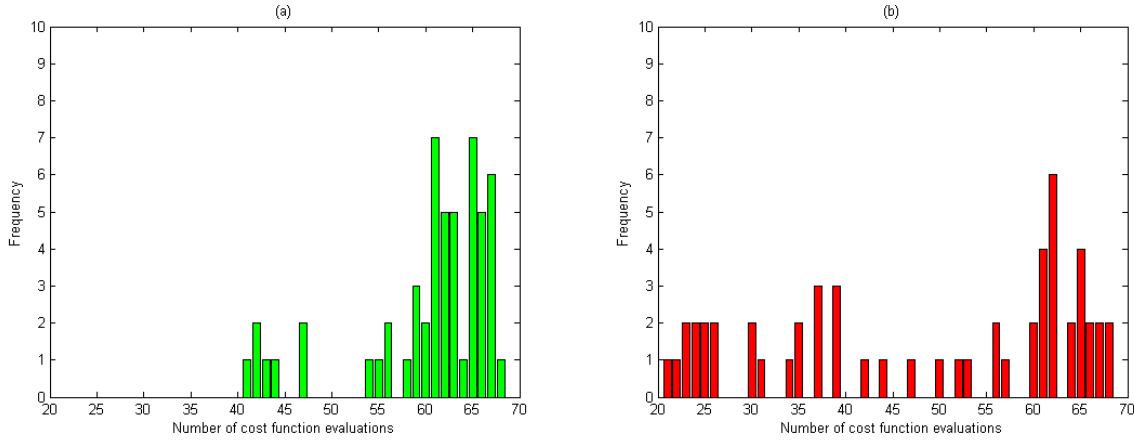


Figure 44. EGO performance with (a) OMLHD initial data set. (b) LHD initial data set. We show the number of occurrences (frequency) of the number of function evaluations required for convergence, e.g. in (a) there are seven occurrences where convergence required 61 cost function evaluations.

#### 4.5.4 A Four-Dimensional Design Problem Using a GA

The following optimization problem is the design of a unit cell for a special material called a metamaterial [26]. This material has interesting applications in optics and microwave engineering resulting from properties not found in natural materials, specifically, negative values for permittivity,  $\epsilon$ , and permeability,  $\mu$ . This artificial material consists of planar arrays of unit cells consisting of conductors called split ring resonators (SRRs) producing a desired magnetic response (associated with the permeability) and straight, conducting wires producing a desired electric response (associated with the permittivity) [26]. The desired response is that  $\epsilon$  and  $\mu$  each be equal to negative one over the operating frequency band of 9 to 11 GHz.

Since we are using a GA for our design optimization algorithm we use simple analytic expressions for  $\mu$  and  $\epsilon$  found in References 26 and 45 respectively. The GA requires many cost function evaluations and analytic expressions allow very fast evaluation. The cost function is simply the average over the frequency band of the squared deviations of  $\epsilon$  and  $\mu$  from -1. The goal of the GA algorithm is to minimize the cost function by adjusting the values of the four design variables within appropriate ranges. The four design variables are: (1) a scaling parameter related to the overall size of the unit cell and the SRR within the unit cell; (2) the width of the strip which forms the SRR; (3) the thickness of the gap in the SRR; and (4) the effective radius of the straight wire. The scaling parameter is dimensionless and has values in the range {1 4}. The other variables have dimensions of mm and are in the ranges: {0.05 1.5}; {0.02 1}; and {0.00025 0.625} respectively.

The GA is a search technique which encodes parameter values as genes within a chromosome [3, 14 and 15]. The strategy is “survival of the fittest,” where more fit chromosomes have lower cost and have a better chance of surviving for one generation to the next. The cost function is used to evaluate the cost of each chromosome. The parameters in this problem are continuous valued so we use a continuous-parameter GA (CPGA) where the computer uses its internal precision and associated round-off error to define the accuracy of the parameters rather than using a binary representation [3, 14 and 15]. Each

chromosome contains values of the four design variables, i.e. the four genes, as described above. The 4-D input space of the problem is defined by the ranges of the variables given above. Each generation of the GA algorithm drives the cost of the best chromosome lower and lower resulting in better and better designs.

We also use a performance-based convergence criterion for GA design optimization. The physics of the problem (captured in the analytic model of  $\varepsilon$  and  $\mu$ ) do not allow an absolute minimum to be obtained, i.e. we cannot make  $\varepsilon$  and  $\mu$  both exactly equal to -1 over the frequency band of 9 to 11 GHz. We therefore determine a reasonable value for the cost function by running the GA for many generations until there is no change. We use a slightly larger value of this “converged” cost function as the performance-based convergence criterion. We have separate “performance specs” for the two parameters  $\varepsilon$  and  $\mu$ . The performance-based cost criterion for  $\varepsilon$  is 0.06 and for  $\mu$  it is 0.11. The cost associated with each parameter must be below their individual performance-based criterion before we declare convergence. We run the GA design optimization multiple times and record the number of generations and a figure of merit (FOM) when the value of the cost function is driven below the performance-based convergence criterion. The FOM is an equally weighted sum of the costs recorded for  $\varepsilon$  and for  $\mu$ . Therefore, at the performance-based criteria levels, we have  $FOM = \frac{1}{2}(0.06 + 0.11) = 0.085$ .

The size of the initial data set is 24, i.e. 24 initial chromosomes. This is somewhat less than ten times the dimensionality but we have had success using smaller sets. We compare results for an initial data set generated by using an OMLHD (24, 4) design and an LHD (24, 4) design. Using our MATLAB version of the Joseph and Hung algorithm [4] for producing an OMLHD (24, 4) we obtained a design  $\mathbf{D}_{G2}$  with performance measures  $\rho = 0.0146$ ,  $\phi_p = 0.0716$ , and  $\Psi_p = 0.0529$ . The frequency of occurrence versus the number of generations required for convergence is shown in Figure 45 for 40 total runs of the GA, i.e. 40 runs using the OMLHD initial data set and 40 runs using an LHD initial data set [46].

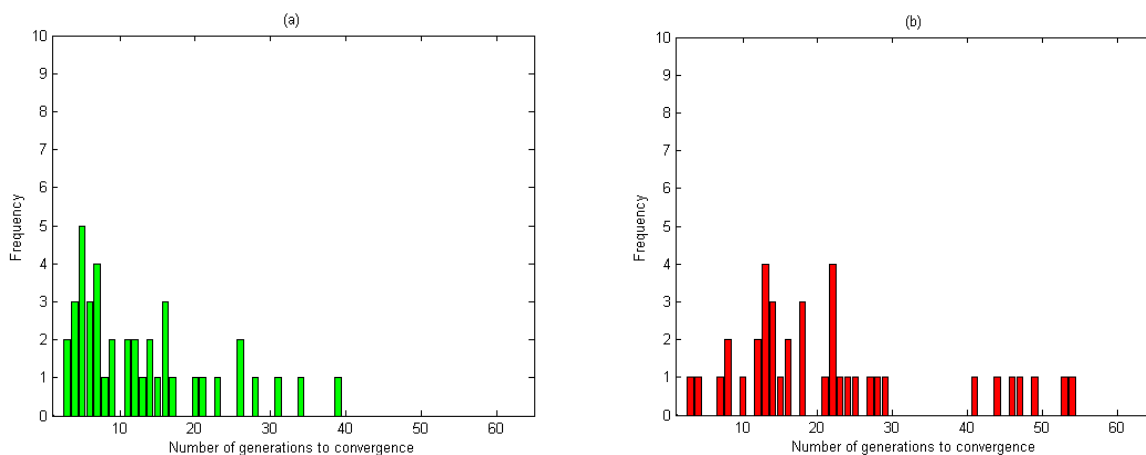


Figure 45. GA performance with (a) OMLHD (24, 4) and (b) LHD (24, 4) initial data sets. We show the number of occurrences (frequency) of the number of function evaluations required for convergence.

Note in Figure 45a that the efficiency of the design optimization algorithm (in terms of the number of generations required to drive the costs below the performance-based criteria) is much better for the OMLHD initial data set than for LHD data sets, shown in Figure 45b. In two instances the GA optimization using LHD did not converge after the maximum number of 200 generations. This is not indicated in Figure 45b.

#### 4.5.5 A Four-Dimensional Design Problem Using EGO

The EGO algorithm was also used in design optimization for the same problem as the GA, i.e. a unit cell of an indefinite material. In this case EGO was coupled with the 3-D, full wave CEM code HFSS [14]. The cost function was therefore expensive. We compare one design using an initial data set of 24 designs found using an OMLHD (24,4) with a single optimized design found using a standard LHD (24,4). The convergence results and performance are shown in Figures 46 and 47 for an OMLHD initial data set and in Figures 48 and 49 for an LHD initial data set.

The number of initial samples was 24 for both designs (as was the case for the GA optimization) and we allowed 60 additional iterations of the EGO algorithm without imposing a performance-based convergence criterion. Therefore the total number of cost function evaluations was 84. The OMLHD case resulted in a best design at sample 70, i.e. at the 46<sup>th</sup> iteration. The LHD case resulted in a best design at sample 33, i.e. at the 9<sup>th</sup> iteration. Therefore the efficiency of the LHD case was much better. In terms of performance, the figures of merit for the two electric responses (upper left curve in Figures 47 and 49) were within 2% of each other with the LHD case being slightly better. However, the figure of merit for the magnetic response of the OMLHD case (upper right curve in Figure 47) is better by more than a factor of two for the LHD case (upper right curve in Figure 49). The figure of merit is the squared deviation from -1 of  $\text{Re}\{\epsilon\}$  and  $\text{Re}\{\mu\}$ . The two optimum designs from EGO are quite similar as shown in Table 13 below. The performance in terms of permeability (upper right curve) in Figure 47 for the OMLHD initial data set is remarkable [46].

Table 13. List of parameters for the indefinite material unit cell optimum designs: OMLHD and LHD initial data sets.

Design Variable	OMLHD data set	LHD data set
Scaling factor (Dimensionless)	2.20	2.18
Wire width ( $\mu\text{m}$ )	17.56	51.73
SRR Line width (mm)	1.25	1.25
SRR Gap width (mm)	2.00	1.90

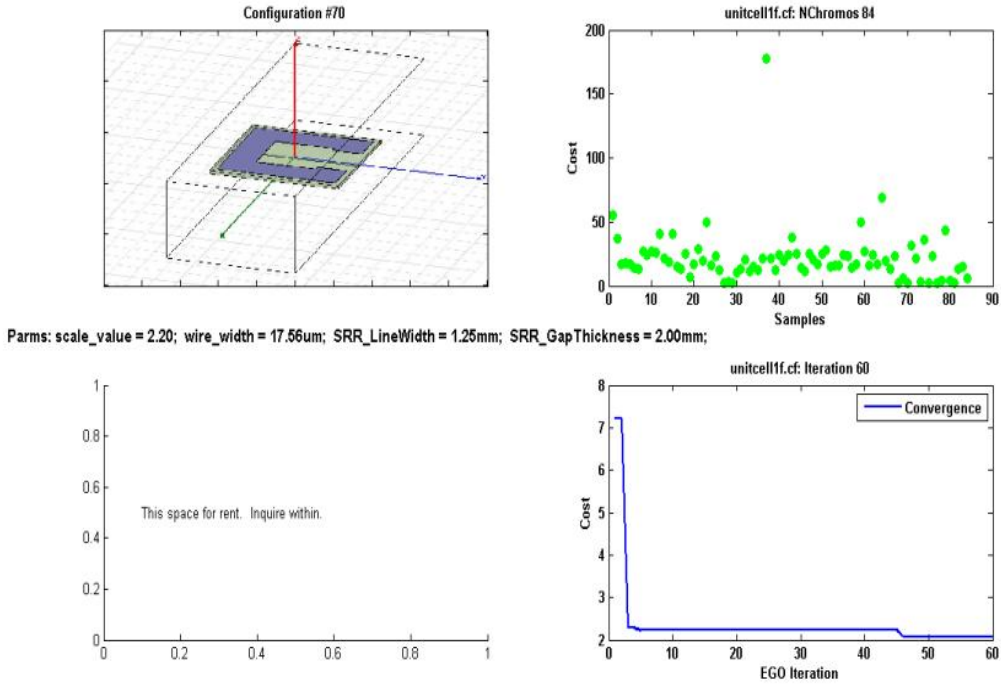


Figure 46. Convergence for the OMLHD initial data set and EGO design optimization of the indefinite material unit cell. The optimum values found for the four design variables are shown below the picture of the unit cell configuration.

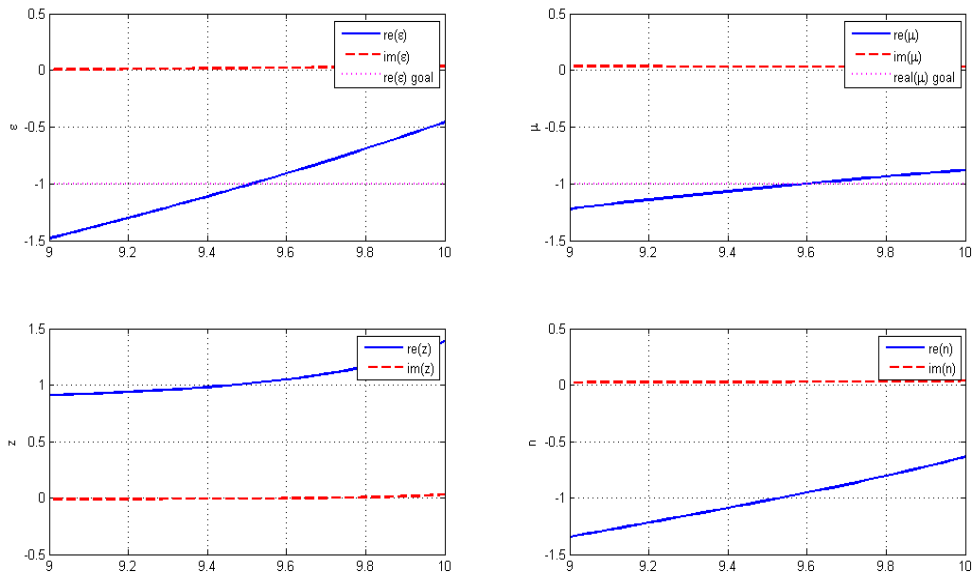


Figure 47. Performance curves for the OMLHD initial data set and EGO design optimization of the indefinite material unit cell. Real and imaginary parts of the permittivity; permeability; refractive index; and normalized impedance are shown clockwise from upper left. The performance goals are negative one for both the real part of the permittivity and the real part of the permeability.

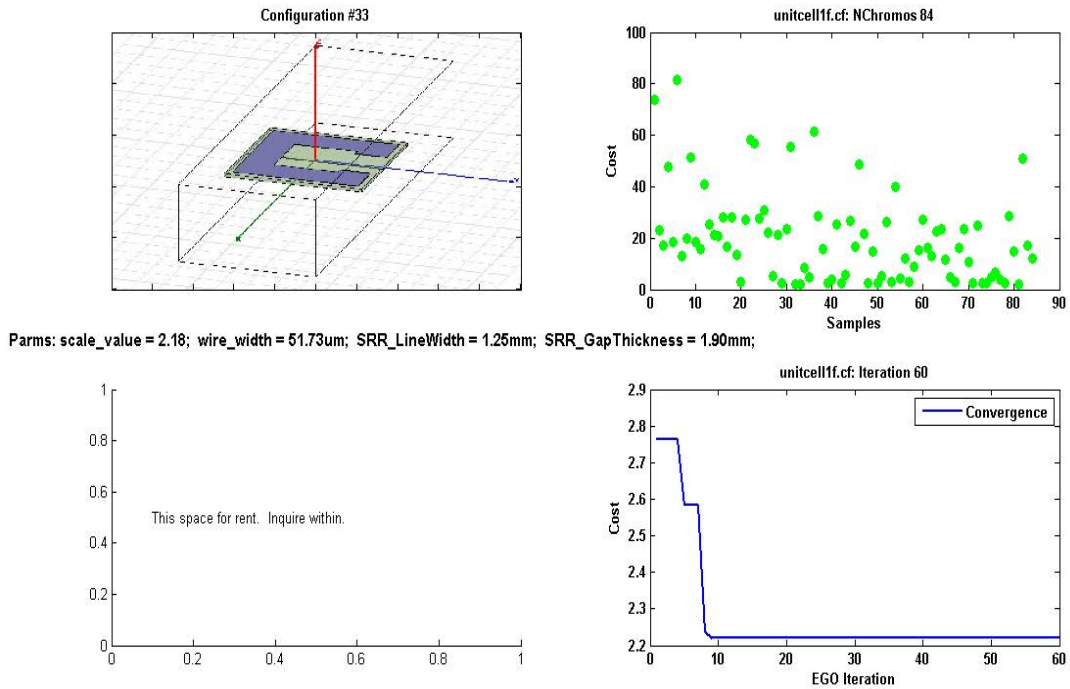


Figure 48. Convergence for the LHD initial data set and EGO design optimization of the indefinite material unit cell. The optimum values found for the four design variables are shown below the picture of the configuration.

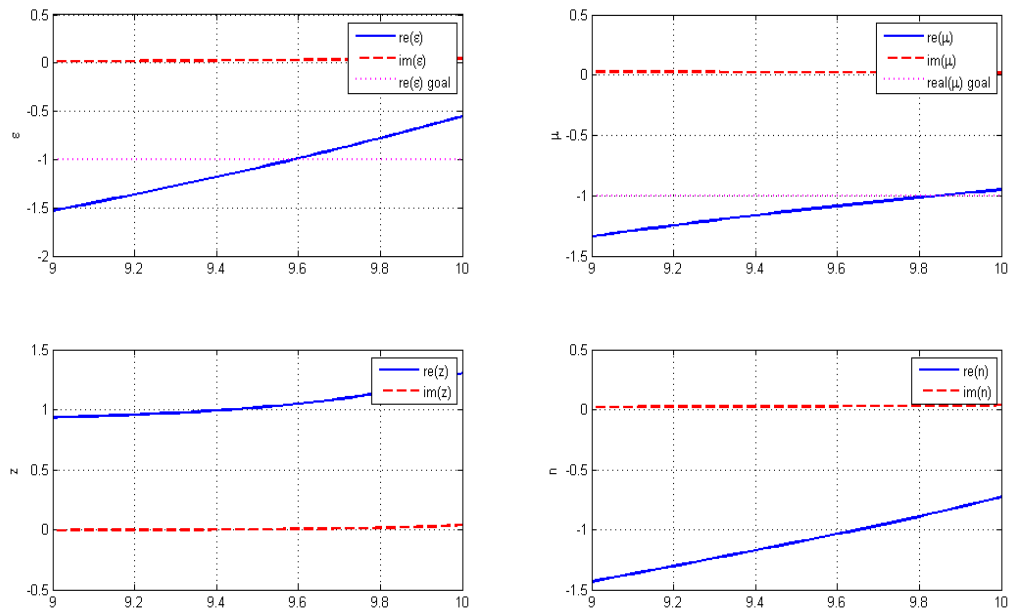


Figure 49. Performance curves for the LHD initial data set and EGO design optimization of the indefinite material unit cell. Real and imaginary parts of the permittivity; permeability; refractive index; and normalized impedance are shown clockwise from upper left. The performance goals are negative one for both the real part of the permittivity and the real part of the permeability.

## 5. CONCLUSIONS

The EGO algorithm was shown to be useful in several simple, theoretical optimization problems and also for practical antenna design problems. It is primarily applicable for a limited number of design variables; although, the wideband fragmented patch antenna (Section 4.3) was optimized for 11 variables. We compared the performance of the EGO algorithm for a relatively simple 2-D problem with two well-known optimization techniques: Nelder-Mead downhill simplex and the GA. The EGO algorithm performed far better (fewer cost function evaluations and more accurate estimates of the global minimum) than either Nelder-Mead or the GA.

Using the EGO algorithm for antenna design optimization required a full-wave CEM simulator. We selected the 3D, full-wave electromagnetic field simulator HFSS (High Frequency Structure Simulator) [20]. One attractive feature of this code is its ability to export results directly into MATLAB. We also require external control of the program, in this case from our MATLAB implementation of EGO. Coupling the design optimization engine (EGO in MATLAB) and the external CEM engine (HFSS) was described in Section 4.2.6. Our coupled design optimization code was used to successfully design a wideband antenna element, the folded triangular bowtie antenna (FTBA) over both indefinite material ground planes and perfect electric conductor ground planes. The FTBA over the PEC ground plane was fabricated and tested with excellent results (Section 4.2.9).

For the wideband fragmented patch array, the EGO design solution is better than the best solution achieved by a GA. The EGO solution was also achieved in significantly less computation time, requiring only about a third of the function evaluations required by the best GA runs. EGO achieves a superior solution earlier than the GA runs, which take two to three times more function evaluations. Even more impressive is the fact that EGO discovered this solution without requiring seeding of the initial population with a known good solution, i.e., the butterfly patch.

Implementations of EGO using a GA to find the next design point for larger dimensionality required very large amounts of computation time. We address this in the next section.

We also implemented endgame techniques which tend to make the EGO algorithm perform local search in the endgame and can result in very accurate estimates of the global minimum. In fact, the techniques found a value of the global minimum which was slightly better than the one found using an exhaustive search.

Finally, we showed that a different technique, the orthogonal-maximin Latin hypercube design (OMLHD), for selecting an initial data set (initial set of measurements, i.e. computer designs) can be advantageous for both EGO and GA optimization techniques when the number of design variables is moderately large (larger than two).

## 6. RECOMMENDATIONS

We recommend the use of EGO for antenna design optimization when the number of design variables is not too large (less than approximately 10). For larger numbers of variables, a GA or some other design optimization technique is more appropriate.

For larger dimensionality, we used a GA in the search for the next design point in EGO algorithm. This proved very time consuming and other, more efficient techniques could be useful for reducing the time required for an optimization run.

The use of an OMLHD for selecting initial data sets for design optimization algorithms is recommended for relatively high dimensional design optimization problems (greater than two dimensions). The computation time required to generate these sets is currently prohibitive for very large sets (large numbers of initial experiments and/or large dimensionality) and we recommend that the algorithm for generating OMLHD data sets be parallelized.

As discussed in Section 3.1.6, EGO has a built-in basis for determining algorithm convergence. However, blindly using arbitrary, ad hoc convergence criteria can give poor results. We believe that for evolutionary computation (EC) optimization techniques in engineering design it is more realistic, and practical, to terminate algorithms based on a performance-based criterion, i.e. where the current value of the cost function meets the design specifications. This replaces artificial, and often arbitrary, convergence criteria. It may seem obvious, but at this point in the iterative process you have an acceptable design; which is the best reason to stop.

## APPENDIX

The design matrix that we obtained for OMLHD (5, 3) using our MATLAB version of the Joseph and Hung algorithm [39] is given by:

$$\begin{bmatrix} 1 & 2 & 3 \\ 2 & 4 & 5 \\ 3 & 5 & 1 \\ 4 & 1 & 2 \\ 5 & 3 & 4 \end{bmatrix}.$$

This matrix is identical to the one in Table 1 in Reference 39. The correlation matrix with elements  $\rho_{mn}$  representing the correlation between columns  $m$  and  $n$  can be found using the MATLAB function *corrcoef* and is given by:

$$\begin{bmatrix} 1 & -0.1 & -0.1 \\ -0.1 & 1 & 0 \\ -0.1 & 0 & 1 \end{bmatrix}.$$

The converged performance measures are  $\rho = 0.0816$  and  $\phi_p = 0.2201$ , the same as in Table 1 [39]. We note that using the same values for the algorithm parameters (starting temperature,  $p$ , etc.) we also obtained a slightly different design matrix:

$$\begin{bmatrix} 1 & 3 & 5 \\ 2 & 4 & 1 \\ 3 & 1 & 2 \\ 4 & 5 & 3 \\ 5 & 2 & 4 \end{bmatrix},$$

with performance measures identical to the ones for the design matrix listed above. The correlation matrix shown below is slightly different:

$$\begin{bmatrix} 1 & -0.1 & 0 \\ -0.1 & 1 & -0.1 \\ 0 & -0.1 & 1 \end{bmatrix}.$$

## REFERENCES

1. Jones, D.R., Schonlau, M. and Welch, W.J. (1998), "Efficient Global Optimization of Expensive Black-Box Functions," *Journal of Global Optimization*, **13**, pp. 455-492.
2. O'Donnell, T.H., Southall, H.L. and Kaanta, B., "Efficient Global Optimization of a Limited Parameter Antenna Design," *Proceedings of SPIE, Evolutionary and Bio-Inspired Computation: Theory and Applications II*, 17-18 Mar 2008, Orlando, FL.
3. Haupt, R.L., "An Introduction to Genetic Algorithms for Electromagnetics," *IEEE Antennas and Propagation Magazine*, Vol. 37, Issue 2, April 1995, pp 7-15.
4. Altshuler, E., and Linden, D., "Wire-Antenna Designs Using Genetic Algorithms", *IEEE Antennas and Propagation Magazine*, Vol. 39, No. 2, April 1997.
5. Boat, A., Michielssen, E., and Mittra, R., "Design of Electrically Loaded Wire Antennas Using Genetic Algorithms," *IEEE Trans on Antenna. & Propagation*, Vol. 44, No 5, May 1996, pp. 687 – 695.
6. Lohn, J.D., Linden, D.S., Hornby, G.S., Kraus, W.F., Rodriguez-Arroyo, A., and Seufert, S.E., "Evolutionary Design of an X-band Antenna for NASA's Space Technology Mission," *Proceedings of the 2003 NASA/DoD Conference on Evolvable Hardware (EH'03)*, 9-11 July 2003, pp. 155-163.
7. O'Donnell, T.H., "Genetic Programming Techniques for Thin Wire Antennas," *2007 Evolutionary and Bio-Inspired Computation: Theory and Applications, SPIE Defense and Security Symposium*, Orlando, FL, April 2007.
8. Santarelli, S., Yu, T., Goldberg, D., Altshuler, E., O'Donnell, T., Southall, H., Mailloux, R. (2006), "Military Antenna Design Using Simple and Competent Genetic Algorithms," *Journal of Mathematical and Computer Modeling*, 43 (2006) 990 - 1022.
9. O'Donnell, T.H., Santarelli, S., Altshuler, E., "Genetic Design and Optimization of Military Antennas," *2006 Modeling and Simulation Conference, SPIE Defense and Security Symposium*, Orlando, FL, April 2006.
10. Sacks, J., Welch, W.J., Mitchell, T.J. and Wynn, H.P. (1989), "Design and Analysis of Computer Experiments (with discussion)," *Statistical Science*, **4**, pp. 409-435.
11. Press, W.H., et al, *Numerical Recipes in C, Second Edition* (1994), p 315.
12. Parzen, E. (1963), "A New Approach to the Synthesis of Optimal Smoothing and Prediction Systems," in R. Bellman, (ed.), *Mathematical Optimization Techniques*, pp. 75-108, University of California Press, Berkeley, CA.
13. Gross, F.B. (Editor), *Frontiers in Antennas: Next Generation Design and Engineering*, The McGraw Hill Companies, 2011, Chapter 6, Section 6.6.4, page 264.
14. Haupt, R.L. and Haupt, S.E. *Practical Genetic Algorithms*, John Wiley and Sons, Incorporated, 1998.
15. Southall, H.L., O'Donnell, T.H., and Kaanta, B., "Efficient Global Optimization for Antenna Design," *Proceedings of the 2008 Antenna Applications Symposium at Robert Allerton Park, IL* (2008).
16. D. E. Goldberg, *The Design of Innovation: Lessons from and for Competent Genetic Algorithms*, Kluwer Academic Publishers, Boston, MA. 2002.
17. O'Donnell, T., Yaghjian, A., Altshuler, E., "Frequency Optimization of Parasitic Superdirective Two Element Arrays," *Proceedings of the 2007 IEEE APS International Symposium*, Honolulu, Hawaii, June 2007.
18. Yaghjian, A.D., O'Donnell, T.H., Altshuler, E.E., and Best, S.R., "Electrically Small Superdirective Linear Arrays," *Digest, URSI Radio Science*, Wash DC, July 2005.
19. Burke, G.J., Poggio, A.J., "Numerical Electromagnetics Code (NEC) – Method of Moments," Rep. UCID18834, Lawrence Livermore Lab., Jan. 1981.

20. Ansoft Corporation, "Addressing High Performance Design," *Microwave Journal*, <http://www.mwjjournal.com/Journal/>, Northbrook, IL, 15 Sep, 2005.
21. Qu, S-W., Li, J-L., Chan, C.H. and Li, S., "Wideband and Unidirectional Cavity-Backed Folded Triangular Bowtie Antenna," *IEEE Transactions on Antennas and Propagation*, 57(4), 1259-1263 (2009).
22. Balanis, C.A., [Antenna Theory: Analysis and Design], John Wiley and Sons, Inc., New York, 449 (1997).
23. Li, R., Thompson, D., Tentzeris, M.M., Laskar, J. and Papapolymerou, J., "Development of a Wideband Short Back Fire Antenna Excited by an Unbalance-fed H-Shaped Slot," *IEEE Transactions on Antennas and Propagation*, 53(2), 662-671 (2005).
24. Kirov, G.S., "Design of Short Back Fire Antennas," *IEEE Transactions on Antennas and Propagation Magazine*, 51(6), 110-120 (2009).
25. Smith, D.R. and Schurig, D., "Electromagnetic Wave Propagation in Media with Indefinite Permittivity and Permeability Tensors," *Physical Review Letters*, 90(7), 077405-1 - 077405-4 (2003).
26. Smith, D.R., Padilla, W.J., Vier, D.C., Nemat-Nasser, S.C. and Schultz, S., "Composite Medium with Simultaneously Negative Permeability and Permittivity," *Physical Review Letters*, (84)18, 4184-4187 (2000).
27. Southall, H.L., O'Donnell, T.H. and Derov, J.S., "Optimum Design of Antennas Using the Efficient Global Optimization (EGO) Algorithm," *Evolutionary and Bio-Inspired Computation: Theory and Applications Conference IV Proceedings of SPIE Defense Security and Sensing Symposium*, 7704, 770408 (2010).
28. Welch, W.J., Buck, R.J., Sacks, J., Wynn, H.P., Mitchell, T.J. and Morris, M.D., and Jones, D.R., "Screening, Predicting, and Computer Experiments," *Technometrics*, 34, 15-25 (1992).
29. Cox, D.D. and John. S. "SDO: A statistical method for global optimization," in N.Alexandrov and M.Y. Hussaini, eds., *Multidisciplinary Design Optimization: State of the Art*, pp. 315–329, SIAM, Philadelphia, (1997).
30. Kenney, J. and Martin, E., Ipswich Antenna Research Facility, AFRL/RHYA, Ipswich, MA.
31. Friedrich, P., et al, "A new class of broadband planar apertures," *Proceedings of the 2001 Antenna Applications Symposium at Robert Allerton Park, IL*, (2001).
32. Thors, B., Steyskal, H., and Holter, H., "Broadband fragmented aperture phased array element using genetic algorithms," *IEEE Transactions on Antennas and Propagation*, 53(10), 3280-3287 (2005).
33. O'Donnell, T.H., Santarelli, S., Steyskal, H., and Southall H., "Hybrid chromosome design for genetic optimization of a fragmented patch array antenna", *Evolutionary and Bio-Inspired Computation: Theory and Applications Conference III; Proceedings of SPIE Defense Security and Sensing Symposium*, 7347, 73470R (2009).
34. Southall, H.L., O'Donnell, T.H., and Kaanta, B., "Endgame implementations for the efficient global optimization (EGO) algorithm", *Evolutionary and Bio-Inspired Computation: Theory and Applications Conference III Proceedings of SPIE Defense Security and Sensing Symposium*, 7347, 73470Q (2009).
35. Holter,, H., and Steyskal, H., "Infinite phased array analysis using FDTD periodic boundary conditions – pulse scanning in oblique directions, " *IEEE Transactions on Antennas and Propagation*, 47(10), 1508-1514 (1999).
36. Wikipedia, "Gray code," [http://en.wikipedia.org/wiki/Gray\\_code#Genetic\\_algorithms](http://en.wikipedia.org/wiki/Gray_code#Genetic_algorithms), (2010).
37. Schonlau, M., Welch, W.J. and Jones, D.R., "Global versus Local Search in Constrained Optimization of Computer Models", *New Developments and Applications in Experimental Design*, IMS Lecture Notes – Monograph Series (1998) Volume 34, pages 11-25.
38. Mockus, J., Tiesis, V. and Zilinskas, A., (1998), "The Application of Bayesian Methods for Seeking the Extremum", *Towards Global Optimization 2*, L.C.W. Dixon and G.P. Szego, editors, pp. 117-129, North Holland, Amsterdam.

39. Joseph, V. R. and Hung, Y., "Orthogonal-Maximin Latin Hypercube Designs," *Statistica Sinica*, **18**, 171-186 (2008).
40. McKay, M.D., Beckman, R.J. and Conover, W.J., "A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code," *Technometrics*, **21**(2), 239-245, (1979).
41. Owen, A., "Controlling Correlations in Latin Hypercube Samples," *Journal of the American Statistical Association*, **89**, 381-402 (1995).
42. Morris, M.D. and Mitchell, T.J., "Exploratory Designs for Computer Experiments," *Journal of Statistical Planning and Inference*, **43**, 1517-1522 (1994).
43. Southall, H.L., O'Donnell, T.H., Derov, J.S. and Allen, J.W., "Antenna Design Using a Metamaterial Ground Plane," *Proceedings of the 2010 Antenna Applications Symposium at Robert Allerton Park, IL* (2010).
44. O'Donnell, T.H., Southall, H.L., Santarelli, S., and Steyskal, H., "Applying EGO to Large Dimensional Optimizations: A Wideband Fragmented Patch Example," *Evolutionary and Bio-Inspired Computation: Theory and Applications Conference IV Proceedings of SPIE Defense Security and Sensing Symposium*, 7707, 77040E (2010).
45. Pendry, J.B., Holden, A.J., Stewart, W.J and Youngs, I. "Extremely Low Frequency Plasmons in Metallic Mesostructures," *Physical Review Letters*, (76)25, 4773-4776 (1996).
46. Southall, H.L. and O'Donnell, T.H., "Initial Data Sampling in Design Optimization," *Evolutionary and Bio-Inspired Computation: Theory and Applications Conference V Proceedings of SPIE Defense Security and Sensing Symposium*, 8059, (2011).