

# Automated Synthesis of a Fixed-Length Loaded Symmetric Dipole Antenna Whose Gain Exceeds That of a Commercial Antenna and Matches the Theoretical Maximum

John R. Koza  
Stanford University  
Stanford, California 94305  
PHONE: 650-960-8180  
koza@stanford.edu

Sameer H. Al-Sakran  
Genetic Programming Inc.  
990 Villa Street  
Mountain View, California 94041  
sameer@genetic-  
programming.com

Lee W. Jones  
Genetic Programming Inc.  
990 Villa Street  
Mountain View, California 94041  
lee@genetic-  
programming.com

Greg Manassero  
Electrical Engineering Consultant  
San Jose, California  
greg.manassero@ieee.org

## ABSTRACT

This paper describes the use of genetic programming to automatically synthesize the design for a fixed-length loaded symmetric dipole antenna whose gain at a specific wavelength exceeds that of a commercially-marketed human-designed antenna and that reaches the theoretical maximum value for an antenna of its type. The run of genetic programming started “from scratch”—that is, without starting from a pre-existing human-created design; did not employ any knowledge base of human design techniques or principles from the field of antenna design; and did not benefit from any human intervention during the run. The run produced a human-competitive result.

## Categories and Subject Descriptors

G.1.6—Global Optimization; I.2.2—Automatic Programming Program Synthesis; I.2.8—Control Methods and Search

## General Terms

Design, algorithms

## Keywords

Genetic programming, automated design, antenna, human-competitive result, invention machine

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

GECCO '07, July 7–11, 2007, London, England, United Kingdom.  
Copyright 2007 ACM 978-1-59593-697-4/07/0007...\$5.00.

## 1 INTRODUCTION

An antenna is a device for receiving or transmitting electromagnetic waves. An antenna may receive an electromagnetic wave and transform it into an electrical signal on a transmission line. Alternately, an antenna may transform a signal from a transmission line into an electromagnetic wave that is then propagated in space.

Maxwell’s equations describe the electromagnetic waves generated and received by antennas and the electrical currents in the antenna. The task of synthesizing the design of an antenna with specified behavior and characteristics is difficult. The design process for antennas typically calls for considerable creativity on the part of the antenna engineer (Balanis 1982; Stutzman and Thiele 1998; Linden 1997). The process by which humans design antennas is similar to that of the design process for analog electrical circuits, controllers, and optical lens systems in that synthesis is considered an art.

Antenna performance can be altered—and often significantly improved—by embedding electrical components, such as capacitors and inductors, into the antenna. For example, a conventional non-loaded half-wavelength dipole consisting of only wire typically achieves a gain of 2.2 dBi, whereas a well-designed loaded version may achieve a gain of 12 dBi or more. The insertion of electrical components into an antenna (called “loading”) exacerbates the task of synthesizing the design of an antenna with specified performance.

The behavior and characteristics of many antennas can be analyzed by simulation. The *Numerical Electromagnetics Code* (NEC) is a method-of-moments simulator for wire antennas that was developed at the Lawrence Livermore National Laboratory (Burke 1992). The NEC simulator is widely used in the antenna community and is considered to be reasonably accurate and reliable for a broad range of structures (Linden 1997).

The NEC simulator works from a text input (called the *geometry table*) describing the antenna. NEC produces output in a well-defined text format that is easily parsed. Source code for NEC (originally in FORTRAN) is available. Thus, the NEC simulator can be readily embedded inside an entirely automated run of a genetic algorithm or genetic programming. In these respects, the NEC simulator for antennas is similar to the SPICE simulator for analog electrical circuits and controllers (Koza, Bennett, Andre, and Keane 1999; Koza, Keane, Streeter, Mydlowec, Yu, and Lanza 2003) and the OSLO and KOJAK simulators for optical lens systems (Al-Sakran, Koza, and Jones 2005).

This paper demonstrates the use of genetic programming to automatically synthesize the design for a fixed-length loaded symmetric dipole antenna whose gain at a specific wavelength exceeds that of a commercially-marketed human-designed antenna and that matches the theoretical maximum value for an antenna of its type. The run of genetic programming started “from scratch”—that is, without starting from a pre-existing human-created design; did not employ any knowledge base of human design techniques or principles from the field of antenna design; and did not benefit from any human intervention during the run. The run produced a human-competitive result.

Section 2 provides a statement of the problem. Section 3 mentions previous work involving the use of genetic algorithms and genetic programming to design antennas. Section 4 discusses our approach to the automated design of antennas. Section 5 discusses the preparatory steps used to apply genetic programming to the automated design of antennas. Section 6 presents the results produced by genetic programming. Section 7 is the conclusion.

## 2 STATEMENT OF THE PROBLEM

Cisco Systems Inc. (2002) offers a 12-dBi high-gain omnidirectional 42-inch antenna operating in the 2,400–2,500 MHz frequency range, as described in the product’s commercial spec sheet (AIR-ANT24120).

The goal in this paper is to automatically synthesize a design for a symmetric dipole antenna whose gain at a specific wavelength exceeds that of the commercially-marketed human-designed antenna and that matches the theoretical maximum value for an antenna of its type.

A good estimate for the maximum practical gain for a dipole at a specific wavelength in dBi (assuming uniform current across the antenna’s length) is  $10\log_{10}2L$ , where  $L$  is the antenna’s electrical length (i.e., the dipole’s physical length divided by the wavelength). A 42-inch antenna (1.0688 meters) is 8.7 times the 0.12245-meter wavelength of a 2,450 MHz wave (the midpoint of the 2,400–2,500 band). The maximum practical gain at a specific wavelength for an  $8.7\lambda$ -long dipole at a single frequency is 12.406 dBi.

## 3 GENETIC METHODS FOR ANTENNA DESIGN

Genetic algorithms have been successfully applied to the design of a variety of antennas, including the design of thinned arrays (Haupt 1994), wire antennas (Linden 1997; Altshuler and Linden 1998, 1999), patch antennas (Johnson and Rahmat-Samii 1999), and linear and planar arrays (Marcano and Duran 1999). The book *Electromagnetic Optimization by Genetic Algorithms*

(Rahmat-Samii and Michielssen 1999) describes numerous applications of the genetic algorithm to antenna design.

Altman, Wiart, and Mittra (1998) used a genetic algorithm to maximize gain for a fixed-length loaded dipole at a specific wavelength.

Jones (1999) applied genetic programming to antenna design.

Comisky, Yu, and Koza (2000) used a developmental representational scheme involving a turtle patterned after the way a draftsman might draw an antenna on paper. The developmental process and turtle was based on earlier work with developmental genetic programming and a turtle (Koza 1993).

Lohn, Hornby, and Linden (2004) used developmental genetic programming to automatically synthesize the design of an X-Band Antenna for NASA’s Space Technology 5 Mission. This work added the additional, and highly advantageous, feature of permitting a wire to bifurcate.

## 4 OUR APPROACH TO AUTOMATIC SYNTHESIS OF ANTENNAS

Our approach to the problem of automatically synthesizing the design of an antenna that satisfies user-specified requirements involves

- (1) establishing a representation for individual antennas that is amenable to a run of genetic programming, and
- (2) defining a fitness measure that measures how well the behavior and characteristics of an individual antenna satisfy the problem’s design requirements.

The implementation of our approach entails working with four different representations for an antenna:

- *Engineering Drawing*: An antenna can be represented as an engineering drawing in two-dimensional or three-dimensional space in which certain lines or curves represent conductive material (e.g., metal).
- *Program Tree*: An antenna can also be represented as a program tree whose internal points (nodes) are functions and external points (leaves) are terminals. This representation enables genetic programming to breed a population of programs in a search for an antenna that satisfies user-specified design requirements.
- *Symbolic Expression*: The program can also be represented as a symbolic expression (S-expression) in the style of the LISP programming language. This representation is used internally by the program code for genetic programming.
- *Geometry Table*: A wire antenna can also be represented as a geometry table that specifies the coordinates of the endpoints of each wire or electrical component and the relevant parameters of each wire (e.g., the radius of each wire) or component (e.g., capacitance, inductance).

The representation and fitness measure are then used in a run of genetic programming. During the run, the evaluation of the fitness of each individual in the population involves

- (1) converting each individual program tree in the population into an antenna,
- (2) converting each antenna into a geometry table of the type accepted by the antenna simulator,
- (3) obtaining the behavior of the individual antenna by simulating its behavior, and
- (4) using the antenna's behavior and characteristics to calculate its fitness.

Developmental genetic programming is especially relevant to work involving the automated design of complex structures. Pioneering work on developmental representations for use with genetic algorithms was done by Wilson (Wilson 1987) and Kitano (Kitano 1990). Pioneering work on developmental genetic programming was done by Gruau (Gruau 1992) for the automatic design of neural networks. In 1993, Koza (1993) used developmental genetic programming to evolve developmental rewrite rules (Lindenmayer system rules) using a turtle to create shapes such as the quadratic Koch island. In 1996, Koza, Bennett, Andre, and Keane (1996) used developmental genetic programming to automatically synthesize a variety of analog electrical circuits, including several previously patented circuits and human-competitive results. This work for automated circuit synthesis entailed the integration of a complex simulator into the runs of genetic programming. Koza, Bennett, Andre, and Keane (1996b) provided for reuse of portions of circuits (by means of subroutines and iterations), parameterized reuse, and hierarchical reuse of substructures in evolving circuits.

## 5 PREPARATORY STEPS

We now describe the repertoire of functions and terminals, a developmental representation for antennas, the fitness measure, control parameters, and termination criterion.

### 5.1 Repertoire of Terminals

The terminal set for each program's single result-producing branch is

$$T = \{\mathfrak{R}, \text{END}\}.$$

$\mathfrak{R}$  denotes a perturbable floating-point number between 0.0 and 1.0. These numbers are interpreted in different ways depending on the function with which they are associated (as explained below).

The END terminal terminates a component-establishing subtree associated with the LOAD function (explained below).

### 5.2 Repertoire of Functions

Our approach to the synthesis of a loaded dipole antenna involves a turtle moving in one dimension. As the turtle moves, it either lays down metal (wire) or inserts electrical components (capacitors or inductors) in various series-parallel arrangements.

The function set for each program's single result-producing branch is

$$F = \{\text{WIRE}, \text{LOAD}, \text{L}, \text{C}, \text{PROGN}\}.$$

A constrained syntactic structure specifies how the functions and terminals may be combined in a program tree. The top-most function of every program tree is a PROGN function.

The connective function PROGN sequentially executes its two arguments. The only allowable arguments for a PROGN function are a WIRE, LOAD, or another PROGN function.

The one-argument WIRE function advances the turtle along the positive  $X$ -axis by an amount specified by its first argument. When the program tree is executed, the turtle starts at the origin of the coordinate system. The origin is the feed point for the antenna's electrical signal. The turtle lays down metal (wire) as it moves. The only allowable argument for a WIRE function is a perturbable numerical value specifying the wire length (after being scaled between  $0.025\lambda$  and  $1.000\lambda$ ).

The one-argument LOAD function advances the turtle along the positive  $X$ -axis by  $0.01\lambda$ . This function inserts a capacitor, an inductor, or a parallel combination consisting of one inductor and one capacitor in the  $0.01\lambda$  space. The components that are inserted are specified by this function's component-establishing subtree. In addition to the electrical properties of the inserted component(s), this portion of the antenna also possesses the electromagnetic properties of a  $0.01\lambda$  wire. The only allowable arguments for a LOAD function are L and C.

The two-argument L function inserts an inductor, and the two-argument C function inserts a capacitor. The first argument of an L or C function is a perturbable numerical value  $\mathfrak{R}$  specifying the component's inductance or capacitance (after scaling). The second argument of an L or C function is a construction-continuing subtree that may consist of another L or C function or an END function.

The evaluation of each individual antenna-creating program tree in the population begins with its execution. The program tree is executed in the usual depth-first order of evaluation (from left to right).

When a WIRE function is encountered, a length of wire is inserted. If consecutive WIRE functions are encountered, the consecutive wires are considered to be one wire (with length equal to the sum of respective lengths).

When a LOAD function is encountered, its component-establishing subtree is executed. If a component-establishing subtree contains only L functions, a single inductor is inserted whose inductance is the sum of the inductances associated with all L functions in the subtree. The behavior is similar if the subtree contains only C functions. If the subtree contains a mixture of L and C functions, then a parallel composition of one inductor and one capacitor is inserted. The inductance of the inductor is equal to the sum of the inductances associated with all L functions in the subtree (and likewise for the capacitor). The incoming and outgoing leads of the single inductor and the single capacitor are connected together.

If consecutive LOAD functions are encountered, then a series composition is created.

Figures 1 and 2 illustrate the formation of a composition of electrical components consisting of an inductor and capacitor in parallel, followed in series by an inductor. Specifically, figure 1 shows the program tree corresponding to the following S-expression:

```
(PROGN
  (LOAD
    (L 0.75 (C 0.25 END) ) )
  (LOAD
    (L 0.50 END) )
)
```

Figure 2 shows the circuit that would be created by the program tree in figure 1. The L and C functions in the left branch of figure 1 create the inductor-capacitor parallel combination shown in the left of figure 2. Note that the numerical values found in the program tree in figure 1 (between 0 and 1) are scaled to yield the 120-nH component value for the inductor in figure 2 and the 40-pF component value for the capacitor. The right branch of figure 1 adds an 80- $\mu$ H inductor in series (right part of figure 2) with the just-created inductor-capacitor parallel combination.

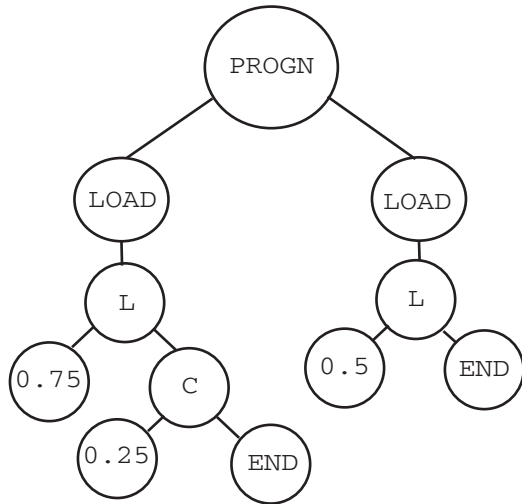


Figure 1 Program tree with two LOAD functions

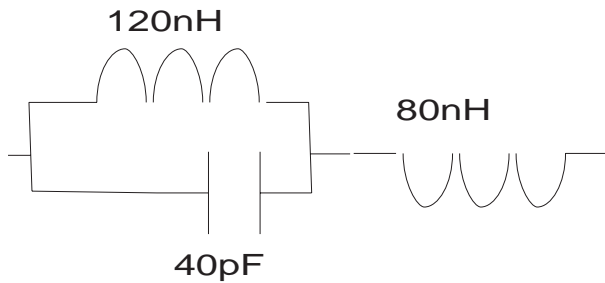


Figure 2 Resulting circuit to be inserted into antenna

### 5.3 Fitness Measure

Fitness is the antenna's gain, in dBi, at a frequency of 300 MHz, measured in the plane perpendicular to the dipole. Obtaining high gain in this direction is generally more difficult than in other directions.

Figure 3 is a flowchart for the execution and evaluation of a program tree during the run of genetic programming.

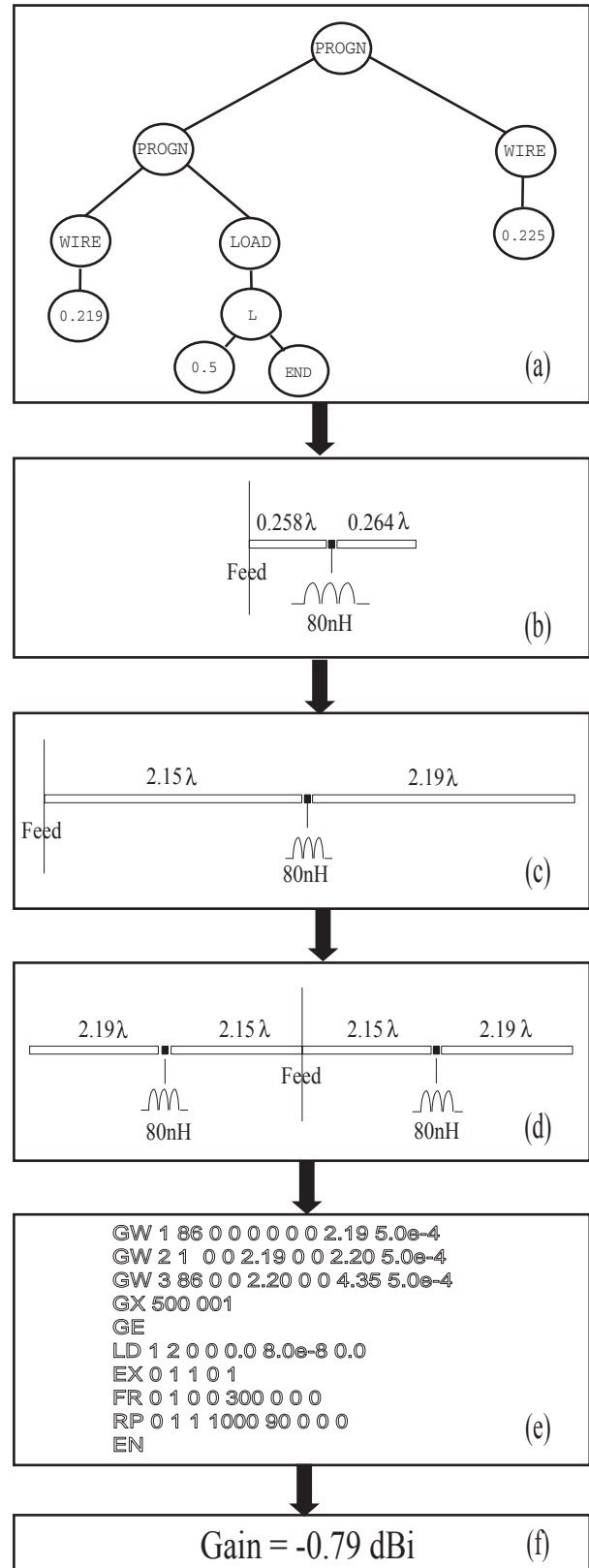


Figure 3 Flowchart for fitness

The program tree (figure 3a) is executed in the usual depth-first (from the left) order of evaluation to yield an unscaled monopole antenna (figure 3b) along the positive  $X$ -axis. The unscaled antenna shown consists of a  $0.258\lambda$  wire, an 80-nH inductor occupying a  $0.01\lambda$  space, and a  $0.264\lambda$  wire.

The unscaled antenna is then scaled to produce the scaled monopole antenna (figure 3c) of length equal to 50% of the desired length of  $8.7\lambda$ .

The scaled monopole antenna (along the positive  $X$ -axis) is then mirrored to yield a symmetric dipole antenna (figure 3d). The result is an antenna whose total length is  $8.7\lambda$ .

We used version 4 of the Numerical Electromagnetics Code (NEC) antenna simulator (Burke 1992) to ascertain the antenna's performance (its gain at a specific wavelength).

The input to the NEC simulator is a geometry table (figure 3e) describing the symmetric dipole antenna. The simulation was conducted with a segment size of  $0.05\lambda$ . This segment size is consistent with that normally employed by antenna designers.

Fitness (figure 3f) is the gain reported by the NEC simulator.

Antennas that cannot be simulated or that begin or end with electrical components (as opposed to wires) are assigned a debilitating value of fitness.

## 5.4 Control Parameters

The population size was approximately 240,000 (750 individuals per node and 320 nodes on a Beowulf-style cluster computer). Crossover, constant mutation, subtree mutation, and reproduction (with elitism for 5 individuals) were performed with a probability of 47%, 40%, 10%, and 3%, respectively. Individuals were created at generation 0 with a minimum tree depth of 4, a maximum depth of 15, and maximum tree size of 1,500.

## 6 RESULTS

The run produced three particularly noteworthy topologies.

The first topology of interest appears in the best-of-generation individual from generation 0 (figure 4). This figure (and subsequent figures in this section) show only the portion of the antenna along the positive  $X$ -axis (i.e., the mirrored portion along the negative  $X$ -axis is not shown). This antenna has a gain of 8.92 dBi. The right half of this best-of-generation individual consists of only four wires and three loads. All three loads are inductive. There are no capacitors present. There are no parallel combinations of electrical components. Thus, "genetic programming starts small" in a manner consistent with the observations contained in the paper entitled "Cross-domain features of runs of genetic programming used to evolve designs for analog circuits, optical lens systems, controllers, antennas, mechanical systems, and quantum computing circuits." (Koza, Al-Sakran, and Jones 2005).

The best-of-generation 6 (figure 5) achieves a gain of 10.06 dBi with the same topology (but different component values) as the best of generation 0.

The second topology of interest appears in the best-of-generation individual from generation 10 (figure 6). This

antenna has considerably more wires and loads than the best individual of generation 0. The best-of-generation individual from generation 10 includes one load consisting of an inductor-capacitor parallel combination, followed in series by a lone inductor. This antenna has a gain of 11.98 dBi.

A pace-setting antenna from generation 25 (figure 7) with this same topology reaches a gain of 12.3 dBi. Its performance is better than that of the commercial product and near the theoretical maximum. This antenna makes use of the parallel-series topological arrangement of the components permitted by genetic programming.

The third topology of interest appears in the best-of-generation individual from generation 25 (figure 8). This antenna achieves a gain of 12.31 dBi. It consists of seven wires, 5 lone inductors, and an LC-L parallel-series combination.

The best-of-run individual from generation 31 (figure 9) achieves a gain of 12.43 dBi. It uses the topology of the best-of-generation individual from generation 25.

Figure 10 shows the radiation pattern in the vertical plane (produced by the simulator) for the best-of-run individual from generation 31. Note that the second largest concentric circle in the figure corresponds to 12 dBi. The figure shows the gain, in the directions of interest ( $-90^\circ$  and  $+90^\circ$ ), to be 12.43 dBi.

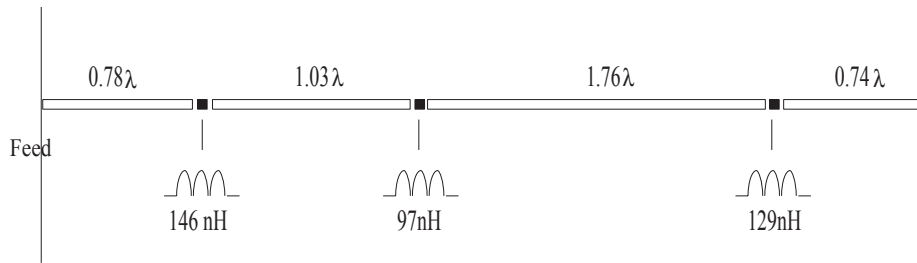
As a reasonableness test, we ascertained the gain of the best-of-run individual from generation 31 over the frequency range (2,400MHz to 2,500MHz) mentioned in the Cisco data sheet. The data sheet specifies only the peak performance over the entire frequency range, so no direct comparison can be made. However, a precipitous drop in gain at near-by frequencies would raise questions about the practicality of the best-of-run antenna. The worst gains for the best-of-run antenna occurred at the two ends of the range. The gain was 11.39 dBi for a frequency of 2,400 MHz and 10.34 dBi for a frequency of 2,500 MHz. The peak gain (12.43dBi) occurred at 2,450MHz. These values lead us to believe that the evolved antenna behaves in a reasonable manner.

Referring to the eight criteria for saying that an automatically created result is human-competitive in *Genetic Programming III: Darwinian Invention and Problem Solving* (Koza, Bennett, Andre, and Keane 1999), the genetically evolved result described in this paper satisfies the following criteria and, therefore, we claim that this result is an instance of a "human-competitive" result produced by genetic programming.

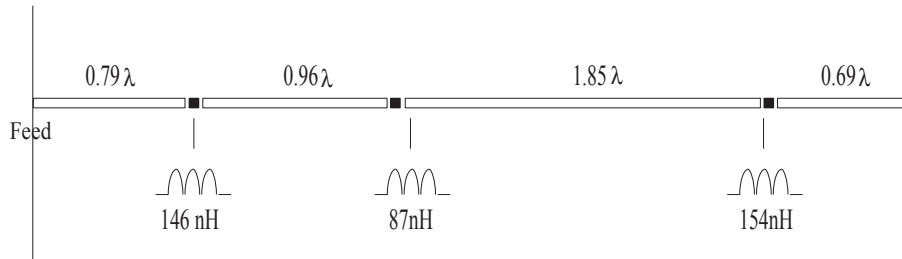
(A) The result was patented as an invention in the past, is an improvement over a patented invention, or would qualify today as a patentable new invention.

(B) The result is equal to or better than a result that was accepted as a new scientific result at the time when it was published in a peer-reviewed scientific journal.

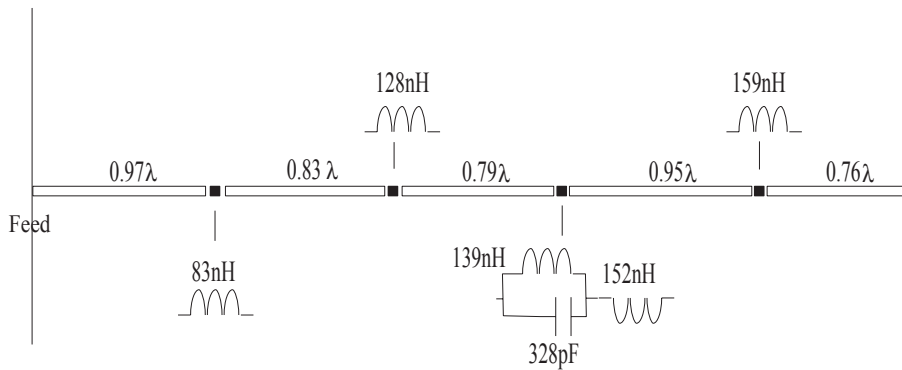
(E) The result is equal to or better than the most recent human-created solution to a long-standing problem for which there has been a succession of increasingly better human-created solutions.



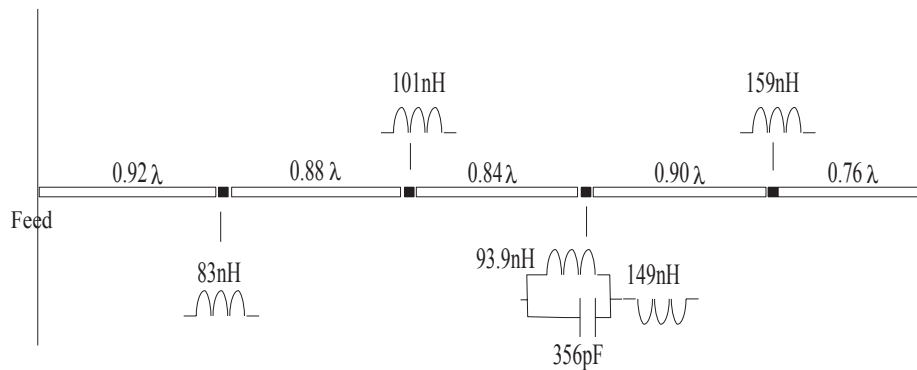
**Figure 4 Best of generation 0—with gain of 8.92 dBi**



**Figure 5 Best of generation 6—with gain of 10.06 dBi**



**Figure 6 Best of generation 10—with gain of 11.98 dBi**



**Figure 7 A pace-setting individual from generation 25—with gain of 12.3 dBi**

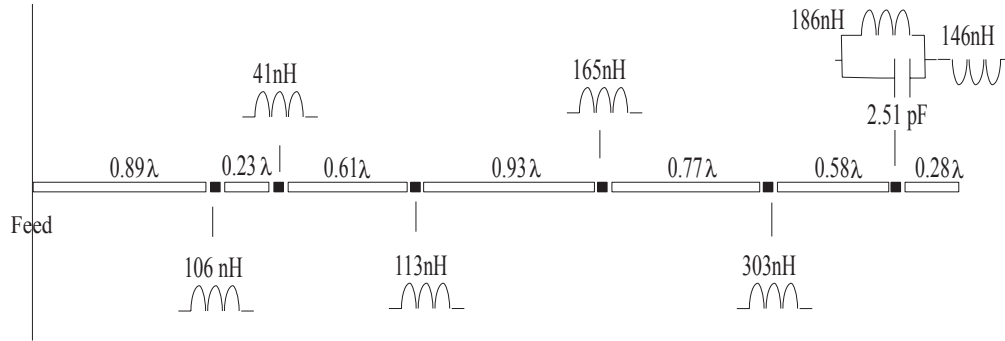


Figure 8 Best of generation 25—with gain of 12.31 dBi

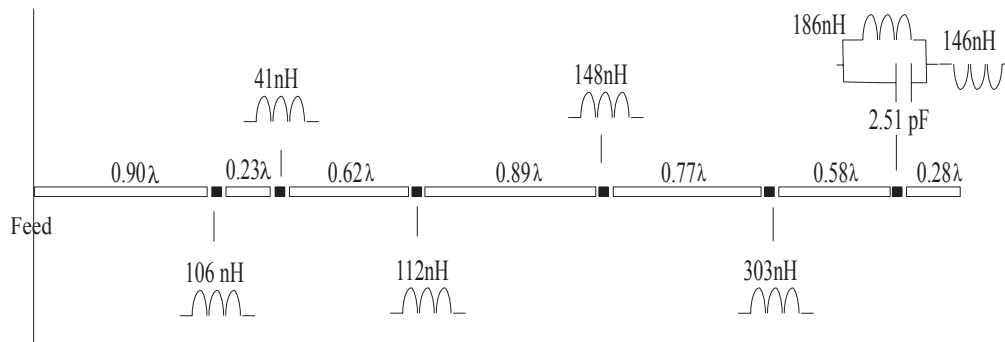


Figure 9 Best-of-run individual from generation 31—with gain of 12.43 dBi

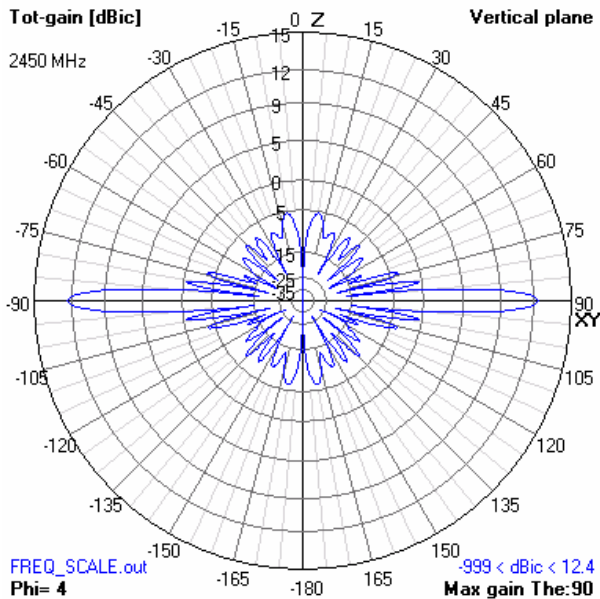


Figure 10 Radiation pattern of best-of-run individual from generation 31—with gain of 12.43 dBi

## 7 CONCLUSION

This paper described the use of genetic programming to automatically synthesize the design for a fixed-length loaded symmetric dipole antenna whose gain at a specific wavelength exceeds that of a commercially-marketed human-designed antenna and that reaches the theoretical maximum value for an antenna of its type. The design is a human-competitive result.

## 8 REFERENCES

- 1 Al-Sakran, Sameer H., Koza, John R., and Jones, Lee W. 2005. Automated re-invention of a previously patented optical lens system using genetic programming. In Keijzer, Maarten, Tettamanzi, Andrea, Collet, Pierre, van Hemert, Jano, Tomassini, Marco (editor). *Genetic Programming: 8th European Conference, EuroGP 2005, Lausanne, Switzerland, March 30-April 1, 2005, Proceedings*. Lecture Notes in Computer Science 3447. Heidelberg: Springer-Verlag. Pages 25-37.
- 2 Altman, Zwi; Wiart, Joe; Mittra, Raj. 1998. Design of high gain dipole antennas using the genetic algorithm. *IEEE Antennas and Propagation Society International Symposium. 1998*. Volume 1. Pages 30-33.
- 3 Altshuler, Edward E. and Linden, Derek S. 1998. *Process for the Design of Antennas Using Genetic Algorithm*. U.S. patent 5,719,794. Applied for on July 19, 1995. Issued on February 17, 1998.

- 4 Altshuler, Edward E. and Linden, Derek S. 1999. Design of wire antennas using genetic algorithms. In Rahmat-Samii, Yahya and Michielssen, Eric (editors). *Electromagnetic Optimization by Genetic Algorithms*. New York, NY: John Wiley and Sons. Chapter 8. Pages 211–248.
- 5 Balanis, Constantine A. 1982. *Antenna Theory: Analysis and Design*. New York, NY: John Wiley and Sons.
- 6 Burke, Gerald J. 1992. *Numerical Electromagnetics Code—NEC-4: Method of Moments—User’s Manual*. Lawrence Livermore National Laboratory report UCRL-MA-109338. Livermore, CA: Lawrence Livermore National Laboratory.
- 7 Cisco Systems Inc. 2002. *Cisco Aironet 12 dBi High gain Omnidirectional Antenna. (AIR-ANT24120)*. Data sheet. San Jose, CA: Cisco Systems Inc.
- 8 Comisky, William, Yu, Jessen, and Koza, John. 2000. Automatic synthesis of a wire antenna using genetic programming. *Late Breaking Papers at the 2000 Genetic and Evolutionary Computation Conference, Las Vegas, Nevada*. Pages 179–186.
- 9 Gruau, Frederic. 1992. *Cellular Encoding of Genetic Neural Networks*. Technical report 92-21. Laboratoire de l’Informatique du Parallélisme. Ecole Normale Supérieure de Lyon. May 1992.
- 10 Haupt, Randy L. 1994. Thinned arrays using genetic algorithms. *IEEE Transactions on Antennas and Propagation*. Volume 42: Pages 993–999.
- 11 Johnson, J. Michael and Rahmat-Samii, Yahya. 1999. Genetic algorithms and method of moments (GA/MOM) for the design of integrated antennas. *IEEE Transactions on Antennas and Propagation*. 47(10)1606–1614. October 1999.
- 12 Jones, Eric A. 1999. *Genetic Design of Antennas and Electronic Circuits*. Ph.D. Thesis. Department of Electrical and Computer Engineering. Duke University.
- 13 Kitano, Hiroaki. (1990). Designing neural networks using genetic algorithms with graph generation system. *Complex Systems*. 4(1990) 461–476.
- 14 Koza, John R. 1993. Discovery of rewrite rules in Lindenmayer systems and state transition rules in cellular automata via genetic programming. *Symposium on Pattern Formation (SPF-93)*, Claremont, California. February 13, 1993.
- 15 Koza, John R., Al-Sakran, Sameer H., and Jones, Lee W. 2005. Cross-domain features of runs of genetic programming used to evolve designs for analog circuits, optical lens systems, controllers, antennas, mechanical systems, and quantum computing circuits. In Lohn, Jason; Gwaltney, David; Hornby, Gregory; Zebulun, Ricardo; Keymeulen, Didier; and Stoica, Adrian (editors). *Proceedings of the 2005 NASA/DoD Conference on Evolvable Hardware*. Los Alamitos, CA: IEEE Computer Society Press. Pages 205–212.
- 16 Koza, John R., Bennett III, Forrest H, Andre, David, and Keane, Martin A. 1996a. Automated design of both the topology and sizing of analog electrical circuits using genetic programming. In Gero, John S. and Sudweeks, Fay (editors). *Artificial Intelligence in Design '96*. Dordrecht: Kluwer Academic Publishers. Pages 151-170.
- 17 Koza, John R., Bennett III, Forrest H, Andre, David, and Keane, Martin A. 1996b. Reuse, parameterized reuse, and hierarchical reuse of substructures in evolving electrical circuits using genetic programming. In Higuchi, Tetsuya, Iwata, Masaya, and Liu, Weixin (editors). *Proceedings of International Conference on Evolvable Systems: From Biology to Hardware (ICES-96)*. Lecture Notes in Computer Science, Volume 1259. Berlin: Springer-Verlag. Berlin: Springer-Verlag. Pages 312-326.
- 18 Koza, John R., Bennett III, Forrest H, Andre, David, and Keane, Martin A. 1999. *Genetic Programming III: Darwinian Invention and Problem Solving*. San Francisco, CA: Morgan Kaufmann.
- 19 Koza, John R., Keane, Martin A., Streeter, Matthew J., Mydlowec, William, Yu, Jessen, and Lanza, Guido. 2003. *Genetic Programming IV: Routine Human-Competitive Machine Intelligence*. Kluwer Academic Publishers.
- 20 Linden, Derek S. 1997. *Automated Design and Optimization of Wire Antennas Using Genetic Algorithms*. Ph.D. Thesis. Department of Electrical Engineering and Computer Science. Massachusetts Institute of Technology.
- 21 Lohn, Jason D., Hornby, Greg S., and Linden, Derek S. 2004. An evolved antenna for deployment on NASA’s Space Technology 5 Mission. In O’Reilly, Una-May, Riolo, Rick L., Yu, Gwoing, and Worzel, William (editors). *Genetic Programming Theory and Practice II*. Boston: Kluwer Academic Publishers. Chapter 18.
- 22 Marcano, Diogenes and Duran, Filinto. 1999. Synthesis of linear and planar arrays using genetic algorithms. In Rahmat-Samii, Yahya and Michielssen, Eric (editors). *Electromagnetic Optimization by Genetic Algorithms*. New York, NY: John Wiley and Sons. Chapter 6. Pages 157–179.
- 23 Rahmat-Samii, Yahya and Michielssen, Eric (editors). 1999. *Electromagnetic Optimization by Genetic Algorithms*. New York, NY: John Wiley and Sons.
- 24 Prusinkiewicz, Przemyslaw, and Lindenmayer, Aristid. 1990. *The Algorithmic Beauty of Plants*. New York: Springer-Verlag.
- 25 Wilson, Stewart. W. 1987. The genetic algorithm and biological development. In Grefenstette, John J. (editor). *Genetic Algorithms and Their Applications: Proceedings of the Second International Conference on Genetic Algorithms*. Hillsdale, NJ: Lawrence Erlbaum Associates. Pages 247-251.