Evolved Design of Microstrip Patch Antenna by Genetic Programming

Thuan Bui Bach[1], *Linh Ho Manh[1], Kiem Nguyen Khac[1], Chien Dao Ngoc[2], Ricardo Zich[3]
[1]Hanoi University of Science and Technology
[2]Ministry of Science and Technology, Hanoi, Vietnam
[3]Department of Energy, Politecnico Di Milano
*linh.homanh@hust.edu.vn

***Abstract*—An evolved antenna is an antenna designed fully or substantially by an automatic computer design program that uses an evolutionary algorithm. In this article, we present our work in using evolutionary algorithms to an automated antenna design system. Based on the primal individual structure of genetic programming (GP) is a tree form, a new data-structure computer program which can be represented as entire parameters of an antenna has been explored. The first experiment has been done successfully for automated design the antenna for 5G mobile device which is microstrip patch antenna (MPA) that operates at 3.5 GHz with 50-160 MHz of bandwidth. The innovative MPAs are obtained by this software. This work shows a great potential of the development of the intelligent computer program for automated synthesis antenna as well as conformal antenna.**

***Index Terms*—Genetic Programming, automated design, evolved antenna, patch antenna.**

I. Introduction

Current methods of designing and optimizing antennas by hand are time, effortful, limit complexity and require significant professionalism and experience. With this approach, an antenna engineer will select specific class of antenna and then spend weeks or months for adjusting and testing to get desired results. Also, the antennas need to be inexpensive, efficient, and robust for the installation environment. As another approach which can overcome these limitations is building an evolutionary software that can find out the effective design solution that would originally not be found [1]. Therefore, this work places emphasis on new approach to automatically design the antennas and describe an example of innovative MPA created using this technology that operates at 3.5 GHz as an ideal and suggested bandwidth of 5G technology.

In Genetic Programming we evolve a population of computer programs. That is, generation by generation, GP stochastically transforms populations of program into new, hopefully better [2]. Also, GP is an evolutionary computation method which is capable of both synthesizing new topologies and optimizing design variables, while only requiring design specifications [3]. In order to have a true microstrip patch antenna design, GP is used to explore complex 2D design on a limited surface of the created substrate layer, the available design space in this work is discretized and encoded to a data structure. Also, the normal optimizers can only find the best solution among predefined shapes of antenna [4]. Therefore, an approach for solving the need of conformal and multiband in restricted area is introduced. Every found antenna layout is described by a distinct and unique tree-structured correspondingly, as an example illustrated in Figure 2 (the more detail can be seen later). Then, the parallelism technique is used to evaluate the fitness of all tree-structured MPAs via electromagnetic (EM) simulator HFSS [5]. Figure 1 shows the implementation of GP software and the entire program is controlled from PYTHON.

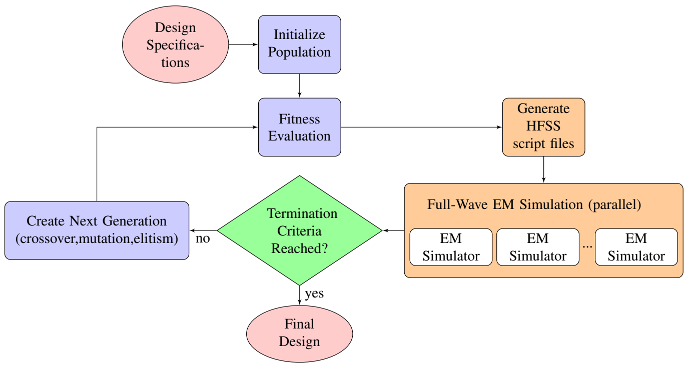
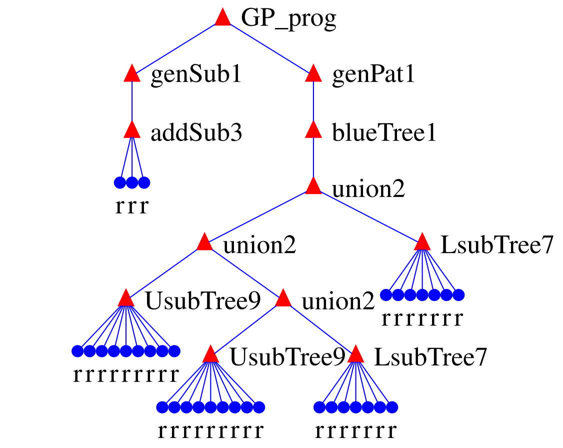

Design Specifica-tions
Initialize Population
Fitness Evaluation
Generate HFSS script files
Full-Wave EM Simulation (parallel)
EM Simulator
EM Simulator
...
EM Simulator
Termination Criteria Reached?
no
yes
Create Next Generation (crossover,mutation,elitism)
Final Design

Fig. 1. Flow chart of GP software

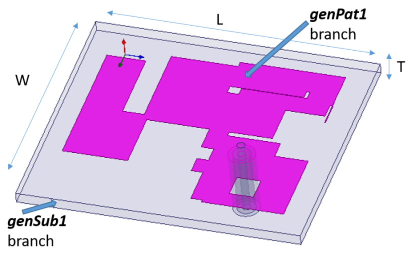
II. Genetic Programming Methodology

When GP is used to design the microstrip patch antenna, the solution is represented as the computer program and running the program generates the geometry of the MPA. In order to obtain the desired results from computer program, the following preliminary preparation steps need to be defined before implementation: program architecture, function set, terminal set, fitness function and termination criterion. In addition, the methods to initialize the population and GP operators is also determined. All of these are illustrated in more detail next.

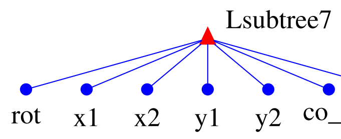
1) Program Architecture: The program architecture of GP software presented in this paper contains of two main branches: one for creating the substrate layer and the other for creating the patterns, as illustrated in Fig. 2(a). The branches of pattern grow depending on the substrate layer, therefore when the full-tree is created the branch of substrate layer need be generated before. The tree-based data structure is represented as a program, and in this implementation there are five types of sub-tree: (1) substrate, which creates a new substrate layer,



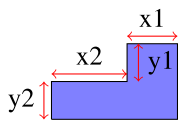
(a)



(b)



(c)



(d)

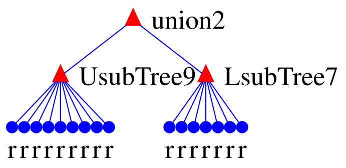
Fig. 2. An example of (a) the program architecture, (b) and it's antenna structure result, (c) the tree-structured of basic L shape and (d) the detail in observation of L tree-structured. For the tree, the functions are specified as red triangles, whereas red types "r" are specified as blue circles. The last character of each name is to indicate it's the number of inputs.

(2) pattern, which create a new pattern, (3) blue, which generates a branch of a pattern, (4) union, which unite two basic polygons,(5) basic, which generate the basic polygons. To make sure that the program running correctly when the crossover and mutation genetic operations are performed, the replacement of the sub-trees must be operated in the same type. In addition, the program has predetermined internal storage for saving user specifications (including maximum-minimum thickness, substrate size, size of each basic polygon edge, material properties for both substrate and patterning, return loss goals, frequency range and number of points to simulate, some essential parameters for GP process) and sample structures which are used to create the script files for the full-wave electromagnetic simulator.

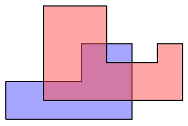
2) Function and Terminal Sets: The function and terminal (inputs to the functions) sets are the main elements to construct the desired antenna and the program architecture, thus the pursuing categories of functions are used: drawing, connecting, creating/modifying and drawing. In next, we explain in detail of the function and terminal sets that used to synthesis each of five sub-tree.

Before go through those, an important terminal type as red-node (red types "r" are specified as blue circles shown in Fig. 2 (b)) need be defined. This node is a random number with range [-1 1], all red-nodes are interchangeable and thus to achieve a final value within the allowed ranges of values (e.g. minimum and maximum movement range of substrate size). Red-nodes are actually the inputs to the drawing function as thickness, angular rotation, substrate size, basic shape edge size, etc. This is similar to what is done when GP is used for synthesize 3D artificial magnetic conductor ground planes [6].

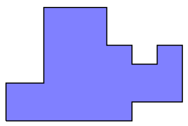
In *substrate* sub-tree, the generating the substrate branch (particularly the nodes that grow up from the left of the root node *GP_prog*), the *genSub1* node is the connecting function used for adding one or more substrate layers. Also, It's terminal set is the substrate creating function *addSub3* and three inputs of *addSub3* specify the thickness, the length and the width of specified substrate layer size. When the substrate layer is added, then the *pattern* sub-tree, the branch for creating the patterns will start generating (the nodes that grow up from the right of the root node *GP_prog*). The *pattern* sub-tree has connecting function *genPat1* as the function set and the pattern creating function *blueTree1* as the terminal set. As the *genSub1* node, the purpose of *genPat1* function is to enable more than one pattern (the other can be the pattern that's used for subtracting) to be included in the MPAs. The *blueTree1* is the main part for exploring new forms of the printed layer. In order to have any true *blueTree1*, the *union2*, the *Lsubtree7* or the *Usubtree9* will be represented as the terminal set. Where the goal of *union2* function is to merge two basic shapes into a specified polygon, illustrated more detail in the Fig. 3. The idea to explore automatically the complex polygon as the patch of MPA is using the basic shapes like characters U, I and L which represented as *basic* sub-tree (the basic shapes that used in current version is U and L). In Fig. 2(c) and Fig. 2(d) illustrate more clear in how these basic shapes represented in this paper (the basic shape U is similarly). The L sub-tree consists of seven inputs, the *rot*



(a)



(b)



(c)

Fig. 3. An illustration of how *union2* sub-tree work

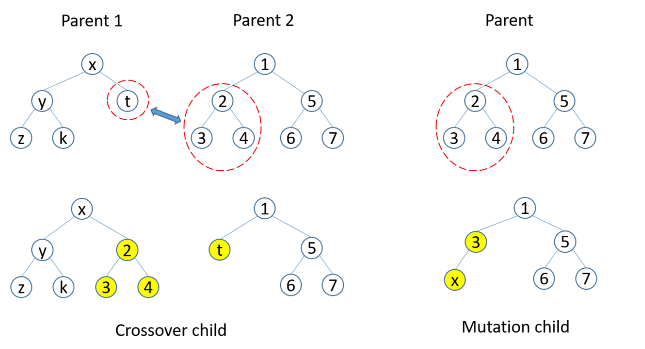


Fig. 4. Example of the crossover and mutation genetic operations, where crossover child is created by two parents and mutation child is created by only one parent

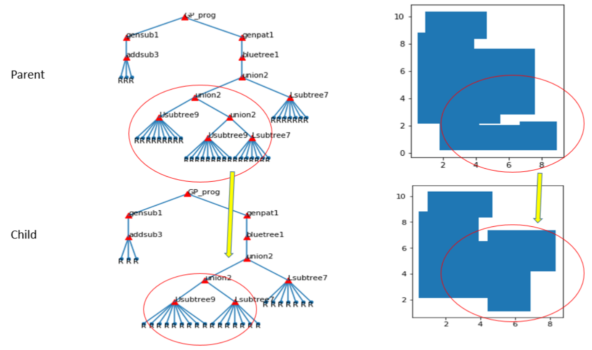


Fig. 5. An actual example of mutation process in the software for the changing of printed layer

indicates for rotational angle (0°, 90°, 180° or 270°), *x1, x2, y1, y2* specify the L size and *co_x, co_y* indicate the location of that basic polygon on the substrate layer surface.

3) Fitness Function: The fitness function is used to measure how well the design meets the desired specifications. To evaluate the MPA, the performance score *Cost* is assigned with the following fitness function, where the lower score is considered better:

$$Cost = \sum_{i=m}^{n} S_{11}[i] + \begin{cases} -200, & \text{if } \textit{min(A)} \text{ in } \textit{center range} \\ 0, & \text{otherwise} \end{cases}$$

$$A = [S_{11}[m], S_{11}[m+1], ..., S_{11}[n]]$$

where *m, n* and *center range* are calculated from the design specifications. The scope from *m* to *n* is represented as operation range of the antenna following the user specifications. The *center range* is the small scope which is approximately the resonant frequency. And *A* is an temporary array that saves all S_{11} values from *m* to *n*. Then if the minimization value of array *A* that lies in the *center range*, the *Cost* of this antenna model will be evaluated better than other.

4) Control Parameter and Termination Criteria: The control of initialization and the run are managed from the control parameters which are the GP specifications. Also, the population size, genetic operation rates, maximum initial depth for the five types of sub-tree and the termination criteria are controlled from these. The guaranteed reproduction rate is 10% (this rate keep for the evolutionary process to take place), the probability of crossover is 40% and the probabilistic mutation is 50% (Fig. 4 and Fig.5 show how crossover and mutation work in here). Where the rates to select one of five sub-tree for crossover and mutation operation are 75% for blue sub-tree (including for basic sub-tree, union sub-tree), 20% for red-node and 0.5% for substrate sub-tree. In this implementation, mutation rate is assigned highest to help the GP converge rapidly (easily get out of a local minimum). The rate within crossover and mutation put a strong focus on creating new topologies and lesser focus on modifying the numeric values for optimization problem.

TABLE I
GP RUN PARAMETERS FOR MPA DESIGNS

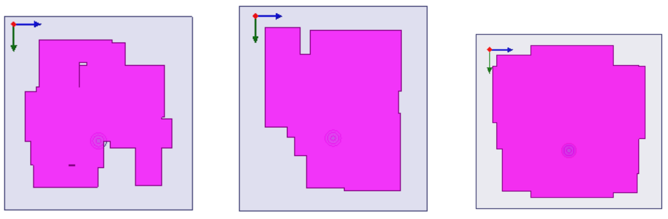
Goal	S_{11} < -10 dB at 3.5 Ghz
Initialization	initial sub-tree depths: blue = 3, pattern = 1, substrate = 1
Population size	20
Number generation	10
Termination	Cost ≤ -900 or run for at least 10 generations
Substrate size	15 mm ≤ L ≤ 26 mm, 15 mm ≤ W ≤ 26 mm, 0.8 mm ≤ T ≤ 1.5 mm
Substrate material	FR4 epoxy

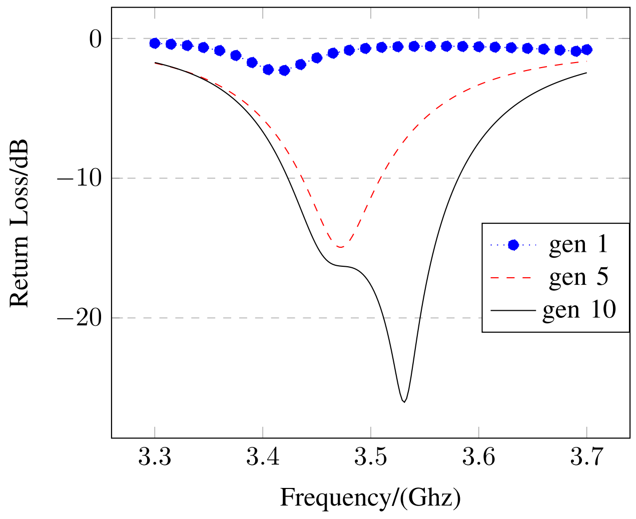
III. Results

To illustrate the use of the software for automated synthesis MPAs, three design examples are given. The GP run parameters for design three these examples are given in Table I. The maximum initial depth of the blue sub-tree is 3. This mean that the depth of blue sub-tree which grows between one and three. And the depth of pattern and substrate sub-tree are 1, mean that the antenna models have only one substrate layer and one pattern (printed layer). The goal is to synthesize an MPA which operates at 3.5 GHz following the user specifications given in Table I.

To show the antenna to be better during the evolution, Fig. 8 visualize all obtained fitness values of each generation. The fitness value decreases gradually according to the increasing of the generation, where lower fitness is considered better. And Fig. 6 shows the difference of three found antennas in generation 1, 5 and 10. Three reflection coefficient S_{11} curves of three antenna models are shown in Fig. 7. These models are some examples that get from the best models at generation 10. The good obtained antennas that have bandwidth from 50 to 165 MHz. Table II shows in detail the sizes and the bandwidth of three antenna models 1, 2 and 3 respectively. And Fig. 9, Fig. 10 and Fig. 11 shows these three antenna models in intuition and it's the radiation patterns, and the program used to create the antenna model 2 is:

GP_prog(gensub1(addsub3(0.1261,0.1776,0.966)),genpat1 (bluetree1(union2(union2(Lsubtree7(0.6384,0.8986,0.324,


(a) generation 1 (b) generation 5 (c) generation 10



(d)

Fig. 6. Comparison of the best found antenna in generation 1, generation 5 and generation 10 respectively antenna model (a), (b) and (c)

TABLE II
THE FOUND ANTENNA SIZES OF THREE EXAMPLE MODELS AND THEIR BANDWIDTHS

	model 1	model 2	model 3
L (mm)	23.75	25.75	16.79
W (mm)	25.33	25.85	19.00
T (mm)	1.42	1.48	1.37
Bandwidth (MHz)	161	66	51

0.7976,-0.6504,0.6151,-0.6009),union2(Lsubtree7(0.9157, -0.0637,-0.405,0.6682,0.829,-0.7426,-0.5451),Usubtree9(0.9869,0.2152,-0.348,-0.1373,0.4313,0.7544,-0.7666, -0.8347,0.3005))),Usubtree9(0.4461,0.617,0.9871,0.7213, -0.371,-0.9607,0.3246,0.7903,-0.9217)))))

Current typical simulation time required for a population size of 20 is about 18 minutes, for entire 10 generation is about 3 hours on a 3.3 GHz Intel Core i3-3220 with 4 CPUs, 4 Gb RAM.

IV. Conclusion

The development and application of GP to synthesizing and optimizing the antenna has been presented. The sets of functions and terminals necessary of the tree-based structure for antenna representation was performed. The synthesis process can match the antenna impedance and do so with minimal

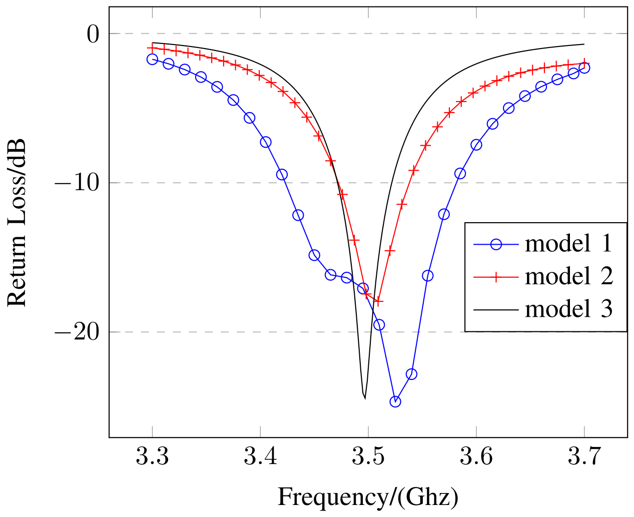


Fig. 7. Return loss

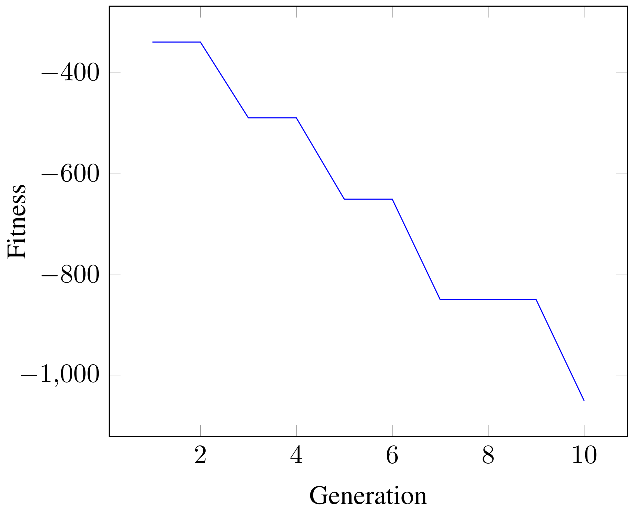


Fig. 8. The change of fitness value during 10 generations

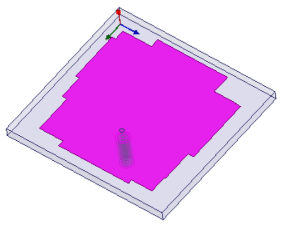


Fig. 9. Antenna model 1

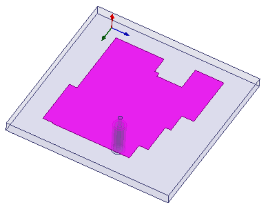


Fig. 10. Antenna model 2

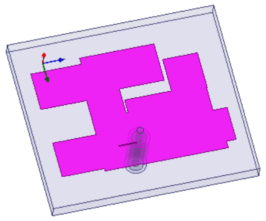


Fig. 11. Antenna model 3

human effort required. This new capability within HFSS can be used for many applications in addition to MPAs design. The software has automatically synthesized new MPAs according to user specifications.

Future work in this area will adding multiple bands, synthesis, and exploring the limits of how small the antenna can be when synthesized in this fashion. Additional features will also be added such as pattern modifying functions, additional methods of drawing patterns. Especially, trying with other type of antenna like conformal antenna, PIFA antenna, etc will be implemented. Finally, performance of the GP software will be increased by parallel evaluation and hybridization of GP with a low-level optimizer.

Acknowledgment

The work described in this paper was supported by CRD laboratory at Ha Noi University of Science and Technology

References

[1] Gregory. S. Hornby ; Jason D. Lohn ; Derek S. Linden ”Computer-Automated Evolution of an X-Band Antenna for NASA’s Space Technology 5 Mission”

[2] Riccardo Poli, William B. Langdon, and Nicholas F. McPhee ”A Field Guide to Genetic Programming ” 2008

[3] J.R. Koza, Genetic Programming: On the Programming of Computers by Means of Natural Selection, MIT Press, 1992

[4] Ho Manh Linh, M. Mussetta, F. Grimaccia, R.E. Zich ”Differentiated Meta-PSO for Rectangular Ring Antenna with Proximity-Coupled Feed” 2013

[5] Linh Ho Manh, Nguyen Khac Kiem, and Chien Dao Ngoc ”Parallel computing in PSO for antenna design”

[6] J.Rayno, M.F. Iskander, N. Celik , ”Synthesis of Broadband True-3D Metamaterial Artificial Magnetic Conductor Ground Planes Using Genetic Programming,” in IEEE Transactions on Antennas and Propagation, vol.62, no.11, pp.5732-5744, Nov. 2014.