

A NOVEL EVOLUTIONARY LEARNING TECHNIQUE FOR MULTI-OBJECTIVE ARRAY ANTENNA OPTIMIZATION

Y. H. Lee, B. J. Cahill, S. J. Porter, and A. C. Marvin

The Applied Electromagnetics Group
Department of Electronics
University of York
Heslington, York, YO10 5DD, UK

Abstract—In this paper, a neural network is used to implement an optimized objective function for a genetic algorithm (GA) for application on array antenna design optimization. Traditional GAs are inefficient because a large amount of data that describes the problem space is discarded after each generation. Using the neural network enhanced genetic algorithm (NNEGA), this redundant information is fed back into the GA's objective function via the neural network. The neural network learns the optimal weights of the objective function by identifying trends and optimizing weights depending on the knowledge that it accumulates in-situ. The NNEGA is successfully applied to challenging array antenna design problems. This use of neural network to optimize a multi-objective function for the GA is a new idea that is different from other hybridization of GA and NN.

1 Introduction

2 The Neural Network Genetic Algorithm

- 2.1 Single Layer Neural Network
- 2.2 Widrow-Hoff Learning Algorithm
- 2.3 The Genetic Algorithm
- 2.4 Objective Function
- 2.5 Neural Network Enhanced Genetic Algorithm

3 Numerical Results

- 3.1 Case One
 - 3.1.1 Design Objectives
 - 3.1.2 Objective Function

- 3.1.3 Analysis
- 3.2 Case Two
 - 3.2.1 Design Objectives
 - 3.2.2 Objective Function
 - 3.2.3 Analysis
- 3.3 Convergence

4 Conclusion

References

1. INTRODUCTION

Recent research has clearly demonstrated the effectiveness of evolutionary computational techniques [1] such as genetic algorithms. A genetic algorithm (GA) [2] is a robust, stochastic search tool that is [3] effective at solving complex, electromagnetic (EM) problems. However, the formulation of an appropriate objective function for a multi-objective EM problem is often difficult.

In this paper, a neural network (NN) [4, 5] is used to implement a generalized objective function for a GA. This is applied to multi-objective array antenna design problems. When working with these problems, one of the most difficult tasks is to formulate an appropriate optimum objective function. The search space is highly dependent on the formulation of this function. Often, a number of GAs with different objective functions need to be tried before an adequate solution to a problem can be found.

Application of a neural network allows the optimization of the objective function to be carried out in-situ with the running GA, allowing trends to be detected that are otherwise hidden. Hence, an optimized objective function is obtained by the neural network from a more general initial guess.

The hybridization of a GA and a NN [6] has been examined in the survey by Schaffer, Whitley, and Eshelman [7]. There are three major uses of a GA with a NN. Firstly, GAs have been used to set the weights for a NN [8, 9]. Secondly, they have been used to learn NN topologies [10, 11]. Thirdly, they have been used to select training data and interpret the output behavior of the NN [12, 13]. In this paper a novel approach of GA and NN hybridization is examined where the NN is used to adapt the weights of the objective function used in the GA optimization [14].

2. THE NEURAL NETWORK GENETIC ALGORITHM

The neural network has long been used as a learning mechanism for pattern recognition and general mapping of functions and data. It has proved to be useful in many areas of engineering [5]. In this paper, the learning capability of the NN is used to identify trends and adapt weights in the search space of a GA so as to optimize the GA's objective function.

In the process of its operation, a GA generates a significant quantity of data describing the problem space while performing its search. This information is normally discarded after each generation. However, this otherwise redundant information can be used to implement a single layer neural network, to optimize the objective function used by the GA. As the data set generated by the GA is relatively small, only a simple NN is used so as to efficiently train the network to recognize trends in the search space. Multilayer feed-forward and back-propagation NNs usually require a large set of training data and are very computation-time intensive. They are therefore less suitable for our purposes.

2.1. Single Layer Neural Network

In this paper, a single layer feed-forward neural network with supervised learning [4] is used. In this implementation, there are neuron-like processing elements that interact using weighted connections. The output of each neuron element is determined by the input they receive. The input to the neural network is a set of values \mathbf{X} , the output is given by the set \mathbf{Y} and the set of weighted connections between the inputs and outputs is given by \mathbf{W} . In supervised learning, \mathbf{T} is the set of target values which defines the desired values for the network. The basic rule is to alter the weights such that the error between the output values and the target values is minimized.

There are many supervised learning algorithms for single layer feed-forward neural networks [5]. These include the least mean square (LMS), the ADALINE, the delta rule and the Widrow-Hoff algorithms. These learning algorithms all share the same rule and are slight variations of each other. These algorithms can be applied to any single layer feed-forward NN using a differentiable activation function. In this paper, the Widrow-Hoff algorithm, one of the commonly used learning algorithms, is implemented.

2.2. Widrow-Hoff Learning Algorithm

The learning procedure introduced by Widrow and Hoff [15] is a form of supervised learning that can be applied to any single layer feed-forward NN. For given input training pattern elements x_i ($x_i \subseteq \mathbf{X}$) and output elements y_j ($y_j \subseteq \mathbf{Y}$), the learning rule for the weights, w_{ij} ($w_{ij} \subseteq \mathbf{W}$), is given by the iterative formula

$$w_{ij}^{new} = w_{ij}^{old} + \eta(t_j - y_j)x_i \quad (1)$$

where η is a learning coefficient which is problem dependent, t_j is the desired or target output value, and y_j is the actual value computed by the neuron for input training pattern x_i .

The network used in this case consists of n inputs and m outputs, equivalent to the number of objectives to be considered by the GA, where $n = m$. The weights are adjusted to reduce the total error E_{tot} over all output units and all training patterns $p = 1, 2, \dots, P$ such that

$$E_{tot} = \sum_{p=1}^P E^p \quad (2)$$

where E_{tot} is the total error, E^p is the error associated with the p th chromosome and P is the total number of chromosomes in each generation. The error for a single pattern p over all output units is given by the sum of the squared errors

$$E^p = \sum_{j=1}^m (t_j^p - y_j^p)^2 \quad (3)$$

The error can be reduced by adjusting the weights in proportion to the negative gradient, the direction of most rapid decrease in the error function, E_{tot} , with respect to each weight change. Thus, we obtain an expression for the weight change Δw_{ij} proportional to the negative gradient of the error, that is given by

$$\Delta w_{ij} = -\eta \frac{\partial E_{tot}}{\partial w_{ij}} = -\eta \sum_{p=1}^P \frac{\partial E^p}{\partial w_{ij}} \quad (4)$$

where η is a positive constant related to the learning coefficient and i and j are the indices of the inputs and outputs associated with the weights respectively. Taking partial derivatives of each term in the sum gives the following,

$$\begin{aligned}
\frac{\partial E^P}{\partial w_{ij}} &= \frac{\partial}{\partial w_{ij}} \left(\sum_{j=1}^m \left(t_j^p - \sum_{t=1}^n w_{ij} x_j^p \right)^2 \right) \\
&= -2 \sum_{j=1}^m (t_j^p - y_j^p) x_i^p
\end{aligned} \tag{5}$$

Dropping the training pattern superscripts, p , and as $n = m$, where the i^{th} input is connected directly to j^{th} output, we have

$$\Delta w_i = \eta(t_i - y_i)x_i = \eta \varepsilon x_i \tag{6}$$

where the factor 2 in (5) is absorbed in the learning coefficient η and ε is the error term $(t_i - y_i)$. The learning coefficient η is problem dependent and can be determined independently for each problem.

In (6), if learning is required, $x_i = 1$ or $x_i = -1$ and if learning is not required (when there is no error), $x_i = 0$. Each Δw_i is then used to adjust the individual weights associated with each objective factor used by the GA in the overall function.

2.3. The Genetic Algorithm

In the GA, a population of chromosomes consisting of random bit strings is generated. The bit strings represent the parameters to be optimized where the number of bits used to represent each parameter determines its resolution. These bit strings are then decoded into parameters from which the fitness of each chromosome is calculated based on an objective function. The process will be elaborated.

As a rule of thumb, the size of population should be equal to the length of the chromosomes [2]. A new population is then produced by using the Elitism Roulette Wheel selection method [2]. The top 10% of the population is directly copied into the new population and the remaining 90% using the Roulette Wheel selection [17]. This ensures survival of the fittest members of the population. After the new population has been chosen, 80% of the chromosomes in this new population undergo uniform crossover [16] in order to produce new child chromosomes. In the uniform crossover scheme implemented, each bit in the chromosome is given a 25% probability of being swapped. This is then followed by mutation (with a probability of mutation of 2% per bit) which prevents premature convergence of the algorithm. The whole cycle is repeated until the population converges to a fixed threshold of 0.5% within each other or when the maximum number of iterations is reached, whichever comes first. The relationship

between the number of iterations and the convergence will be examined for each of the case studies.

Normally after each generation the old population is discarded, losing a significant amount of information describing the problem space.

2.4. Objective Function

In the GA, one of the most difficult tasks is the formulation of a suitable objective function for the multi-objective problem space. Most objective functions are formed using the weighted sum of the different objective factors.

$$f(w_i, y_i) = \sum_{i=1}^n (w_i \cdot y_i) \quad (7)$$

Equation (7) represents a general form of an objective function with n objective factors (y_i) to be optimized. w_i is the weight associated with each factor. In the standard GA, the w_i 's are a set of fixed values determined by the user and the same set of weights is used throughout the whole GA optimization process. In the NNEGA, this set of weights is modified by the neural network based on knowledge gained from previous iterations of the GA.

2.5. Neural Network Enhanced Genetic Algorithm

The objective values that contain information on the search space (generated by the GA after each generation) are fed into the NN as discussed in the previous section. The NN learns from this training pattern and adjusts its weights according to (1) where $i = j$. Equation (2) reduces the error between the training patterns and the target objective function so as to obtain a better set of weights. The adjusted weights are then fed back into the GA after a certain number of generations. By doing so, the convergence of the GA can be ensured and, at the same time, a larger set of training patterns is supplied to the NN for the training process.

Fig. 1 shows the integration of the neural network and the genetic algorithm. In the initialization process, a random population of chromosomes consisting of bit strings is generated. The initial, uniform, weights and the control parameters, such as the learning factor for the NN and probability of crossover and mutation for the GA, specified by the user are initialized. The initial population is then used to calculate the n objective factors to be optimized. These n objective factors (y_i) for each chromosome in the whole population are then passed into the NN as training patterns. The learning

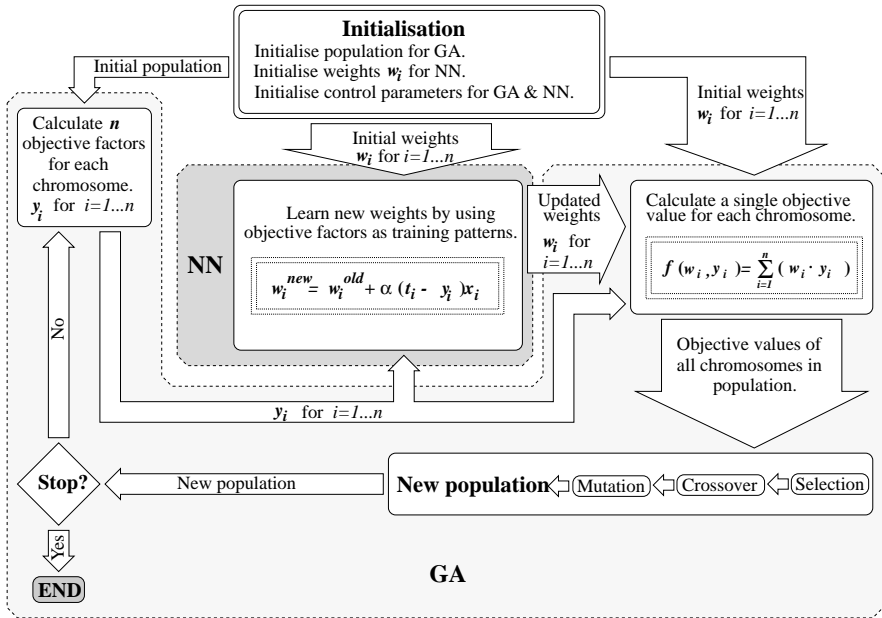


Figure 1. Neural network enhanced genetic algorithm.

coefficient η is problem dependent. Generally, η is chosen to be a small positive constant of value less than 3. The learning coefficient, η , is dependent on the complexity of the problem space. The more complex the problem space, the higher the value of η . Simultaneously, the objective factors are used by the GA with the initial uniform weights (w_i) to calculate one objective value ($f(w_i, y_i)$) for each chromosome in the population. Using this set of objective values, the chromosomes then go through the standard GA optimization process of selection, crossover and mutation. The new population produced is then tested for the termination criteria. If the termination criteria are not met, the n objective factors are again calculated and the whole process is repeated. The weights, w_i , associated with each objective factor are modified by the NN in-situ with the GA and updated in the GA after a user defined number of generations.

Through experimentation, it was found that the number of generations before the weights are updated is dependent on the number of chromosomes in each population (training patterns for the NN) and the number of generations before convergence occurs. For the population sizes used here, there should be about 5 updates before the population converges. After the weights have been fed back into the

GA, the NN is not reinitialized and the training process continues. The NNEGA is an automated evolutionary learning optimization algorithm. A further advantage of this method is that it provides diagnostics. The weights produced by the NNEGA give insight into the behavior of the multi-objective problem space.

3. NUMERICAL RESULTS

To demonstrate how the NNEGA is effective in identifying trends and adapting weights, two practical antenna design problems are considered. These two case studies illustrate two different advantages of using the NNEGA.

3.1. Case One

Fig. 2 shows the original Bilog design together with the new Bilog design to be optimized. There are 5 parameters, P_i , where $i = 1 \dots 5$, associated with the antenna design that are optimized as shown in Fig. 2. Each parameter is encoded into 8 bits resulting in a total chromosome length of 40 bits. In this study, a population of 50 chromosomes is used. The NNEGA goes through a maximum of 50 iterations as investigation into using larger number of iterations has shown negligible improvement to the final objective value.

In order to enhance the low frequency performance of the antenna, the antenna design in Fig. 2 is modified by adding a second low frequency dipole (replacing the Bowtie with the Quad-Wing). The original Bilog is 1.4 m by 0.6 m for the Bowtie and 0.8 m in length along the Log-periodic section. The overall physical size of the new antenna design is limited to 1.5 m by 1.0 m for the Quad-Wing and 0.8 m long for the Log-periodic section. This maintains the new antenna design size such that it is compact and easy to handle. Due to this physical constraint in size, the improvement in low frequencies performance is limited.

3.1.1. Design Objectives

The antenna factor (AF) is defined as the ratio of the incident electric field, $|\hat{E}_{inc}|$, at the surface of the measurement antenna to the received voltage, $|V_{rx}|$, at the antenna terminals [19].

$$AF = 20 \cdot \log_{10} \left(\frac{|\hat{E}_{inc}|}{|V_{rx}|} \right) \text{ dB/m} \quad (8)$$

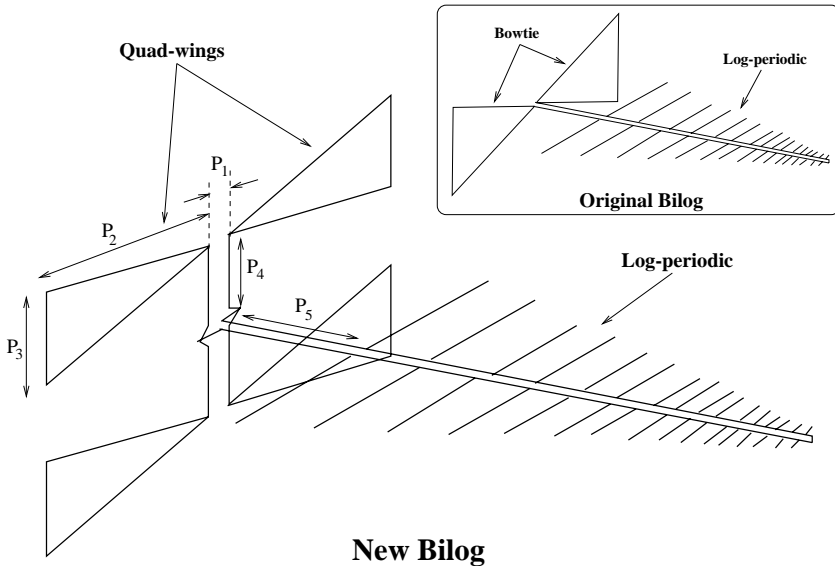


Figure 2. Bilog designs.

The design objective of the Bilog [18] is to reduce the antenna factor at low frequencies so as to make it a better receiving antenna and also require a lower input power when used for immunity testing purposes. To minimize the input power required to the desired level, the antenna factor (8) between 60 MHz and 160 MHz has to be reduced by at least 3 dB from the original Bilog AF. From 160 MHz to 240 MHz, the performance of the antenna should be at least as good as that of the original Bilog. It is important to note that the Bilog is already a highly optimized, electrically small design and further optimization to this design is therefore difficult. Reducing the antenna factor by 3 dB means an improvement in performance of the antenna for emissions testing and a reduction in the power required by the antenna for immunity testing by a factor of 2. For example, a 50 W amplifier can be used instead of a 100 W one. This makes the Bilog a more economical antenna. The optimization of the Bilog design is also a good test of the NNEGA since a standard GA struggles to make any practical improvements to the existing design.

3.1.2. Objective Function

The objective function (9) is given by the weighted sum of the difference in antenna factor at 39 discrete frequency points over the band 60 MHz

to 240 MHz:

$$f(w_i, AF_i) = \sum_{i=1}^{39} w_i \cdot \Delta AF_i \quad (9)$$

where the w_i 's are the weights associated with each objective factor and ΔAF_i is the difference between the target AF and the actual AF. In (9), ΔAF_i is also the training pattern used by the NN to train the weights for the GA.

The 39 points are chosen over two logarithmic scales for the Bowtie and the Log-periodic operational ranges according to the periodic behavior of the antenna.

3.1.3. Analysis

Fig. 3 shows the target objectives, the optimized objective achieved by both the standard GA and the NNEGA respectively, the uniform weights used in a standard GA and the optimized weights achieved and used by the NNEGA.

As can be seen from Fig. 3, the left-hand axis shows the antenna factor in dB/m for the optimized and target objectives. The right-

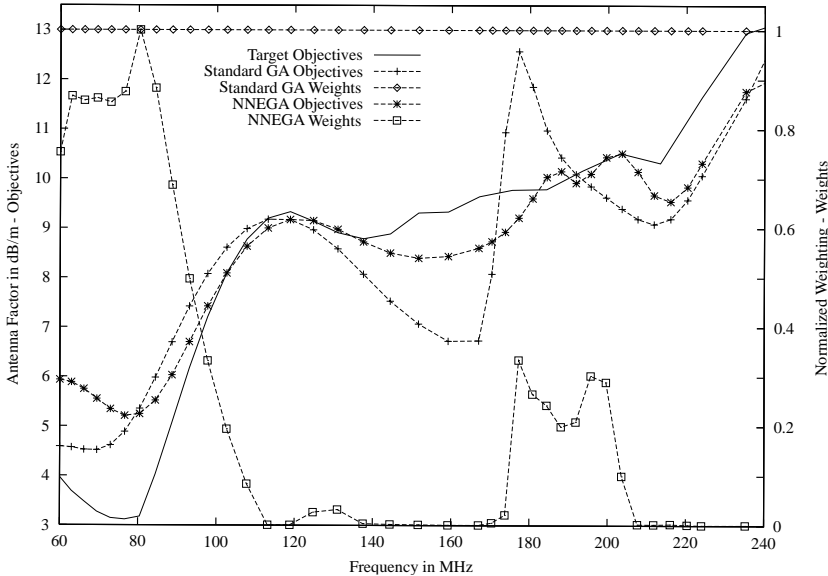


Figure 3. Weights and objective values for a standard GA and a NNEGA optimization on the new Bilog design.

hand axis represents the weights used by a standard GA and those optimized by the NN where the values are normalized to the largest value.

After several manual runs of a standard GA optimization on the new Bilog design it was found that, although the AF was reduced at low frequencies, a sharp discontinuity often appeared between 170 MHz and 200 MHz (Fig. 3). This peak occurs around the region where the operation of the Bilog switches between the Log-periodic section and the Bowtie section. The resulting *optimized* design is undesirable and should be discarded. When using a standard GA, the designer must consider the results after the GA optimization has completed and then make the appropriate changes to the weights in the region where the discontinuity appears in order to discourage these designs. By doing so, an improved objective function is implemented, producing a better Bilog design. This requires operator skill, judgment and time. However, the NN can automatically perform this task during the course of a single GA.

In this antenna optimization, there are two regions where emphasis needs to be placed: the region below 110 MHz where there is a stringent need to reduce the antenna factor by at least 3 dB, and the region where the discontinuity occurs persistently (between 170 MHz and 200 MHz). As can be seen in Fig. 3, the optimized weights found by the NN have successfully identified the region of discontinuity and placed an appropriate emphasis on it so as to prevent the discontinuity from occurring. The highest weight is placed at 80 MHz, as that is the point where the optimized design is furthest away from the target design.

This case study shows that the NNEGA is able to successfully identify trends in the population, and adjust the GAs' weights to react to unwanted trends in the population. The NNEGA has found a superior design for the Bilog than a standard GA because it was able to discard designs with a discontinuity in them. This has been achieved by using the NN to produce an optimized set of weights for the GAs objective function. As explained, to ensure convergence, the weights trained by the NN are fed back only after a number of generations have elapsed. In this case, since the optimization process generally takes 20 to 25 generations to converge, and in order to provide a significant training pattern for the NN (about 200), the weights are updated after every 5 generations. The convergence trend is examined later.

3.2. Case Two

The M-Antenna design [20] problem optimizes the spacing and length, P_i where $i = 1 \dots 6$, of an array of 4 dipoles as shown in Fig. 4. Each parameter is encoded into 8 bits resulting in a total chromosome length

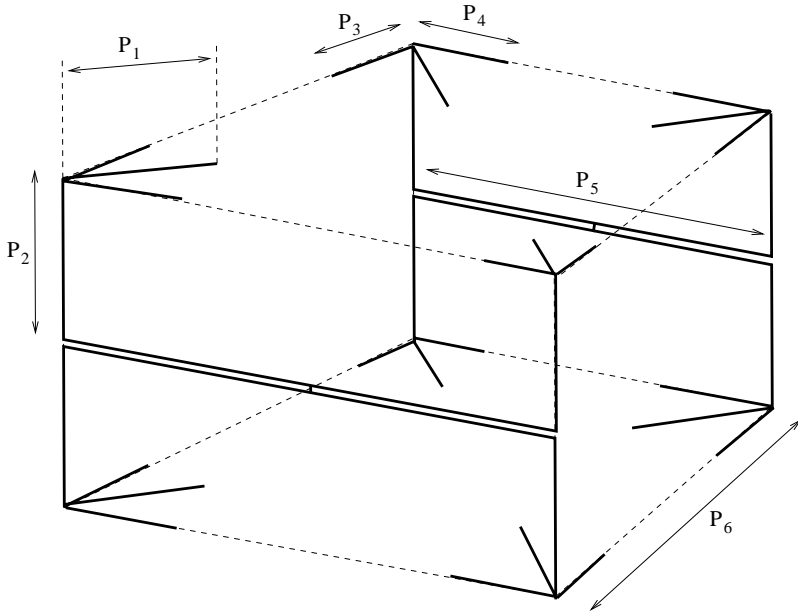


Figure 4. M-Antenna design.

of 48 bits. In this study, a population of 50 chromosomes is used. Similar to case one, it was found that a larger number of iterations has also shown negligible improvement to the final objective value achieved. The design of this antenna is based on Balanis' book on antenna theory [21]. In order to improve the bandwidth of an antenna, the antenna should occupy the maximum available volume of space. As can be seen in Fig. 4, the M-Antenna consists of 4 array dipoles arranged in a manner so as to maximize the volume of space occupied. The name M-Antenna came about due to the geometry of the design achieved by the GA. The dimensions of the M-Antenna are a cube with a height of 3 m, width of 0.8 m and depth of 1 m.

3.2.1. Design Objectives

This array antenna is designed for electromagnetic immunity testing purposes. It is required that the antenna be able to radiate a uniform near electric field over a plane of 3 m by 5 m with a field strength of 10 V/m, at a distance of 3 m in front of the antenna. It is also desirable to minimize its power requirement. The maximum input power allowed is 100 W. These requirements apply over the frequency range from 30 MHz to 100 MHz.

There are three factors to be satisfied: near electric field uniformity; near field strength level; and maximum radiated power relative to the input power. To optimize these objectives: the field uniformity is ensured by taking 77 uniformly distributed points on the plane of interest and ensuring that these points are within a 3 dB range; the uniform near field strength is at least 10 V/m; the power radiated is maximized for an input power of 100 W. Each of these factors is optimized at 8 frequency points from 30 MHz to 100 MHz each 10 MHz apart.

3.2.2. Objective Function

The objective function consists of the weighted sum of each of the 3 factors at the 8 frequency points as explained.

$$f(w_i, y_i) = \sum_{i=1}^{24} w_i \cdot y_i \quad (10)$$

where

$$y_i = \frac{N_{points3dB}}{77} \quad \text{for } i = 1 \dots 8$$

$$y_i = \begin{cases} \frac{E_i}{10.0} & \text{if } E_i < 10.0, \\ 1.0 & \text{if } E_i \geq 10.0. \end{cases} \quad \text{for } i = 9 \dots 16$$

$$y_i = \frac{Eff}{100.0} \quad \text{for } i = 17 \dots 24$$

where $N_{points3dB}$ is the number of points on the plane of interest which lie within a 3 dB range. E_i is the average electric field strength on the plane and Eff is the efficiency of the antenna as a percentage. Notice that these values are all normalized where a unit represents the target objective being achieved. y_i is the normalized value of the i^{th} factor.

In this design problem, the 3 design goals are not of equal importance.

The highest priority, and thus weight, is placed on ensuring a uniform near electric field, followed by ensuring the field is at least 10 V/m and the least important factor is to maximize radiated power.

3.2.3. Analysis

In Fig. 5, the x -axis shows the 24 factors (11) to be optimized, the left-hand y -axis shows the normalized magnitude of the objective values and the right-hand y -axis shows the normalized weights. The values

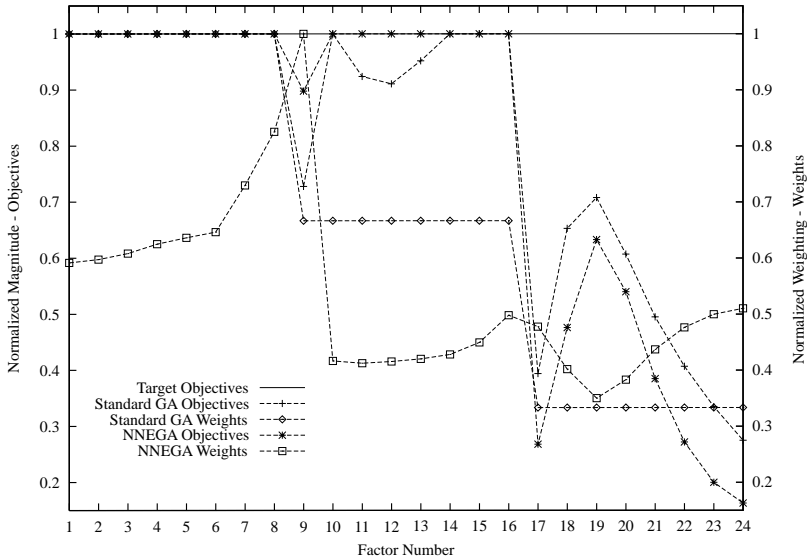


Figure 5. Weights and objective values for a standard GA and a NNEGA optimization on an array of 4 dipoles (M-Antenna).

are normalized to the maximum value achieved by the objectives and weights respectively. As described above, the most important factor is to achieve uniform near electric field. Therefore, when optimizing this antenna, the authors have distributed the weights of each objective in the ratio of 3 : 2 : 1. In Fig. 5 the user defined weights used for the standard GA and the optimized set of weights found by the NN are shown. The *optimized* objective value achieved by using the standard GA and the NNEGA is also shown. Note that the target objective is a straight line at a normalized value of 1 over all the 24 factors to be optimized.

Through various simulations, it was found that it is relatively easy to achieve a uniform field for the 8 frequencies (given by factor numbers 1 to 8). Therefore, the weights for field uniformity can be adapted and redistributed to weights for optimizing the field strength (factor numbers 9 to 16) and the antenna efficiency (factor numbers 17 to 24). In the standard GA, this would have to be observed after numerous runs of the GA and then the weights changed manually during the optimization while constantly monitoring the objective factors to ensure that the most important factor is not compromised.

When the NNEGA is used, the NN identifies factors that are easily achieved and adapts the weights, thus dynamically readjusting them so

that factors that are more difficult to achieve are given higher weight and yet ensuring important factors are always satisfied. The weights achieved by the NNEGA are as shown in Fig. 5.

As can be seen, factor numbers 1 to 6 are reduced and redistributed to factor number 9 and also 16 to 24, resulting in a much flatter weight distribution rather than a step distribution. By running the NNEGA a number of times, it was found the factors that require a higher weight occur consistently around the same regions. This gives an indication of the ease of attaining one objective factor relative to the another. In both case studies, it was found that regions in which the objective factors are relatively more difficult to achieve appear consistently.

This example shows how the NN enables the weights to be adapted between different factors while ensuring the importance of each factor is always maintained and thus achieving a much better antenna design. By using the NNEGA, additional information about the problem space can also be observed by examining the set of weights produced by the NNEGA.

3.3. Convergence

Fig. 6 shows the convergence trend for the 50 generations of the above two case studies. The objective values are maximized and, therefore, the maximum objective value achieved in each generation is plotted. These objective values are normalized to the highest objective value achieved over 50 generations in each case study for both the standard GA and the NNEGA, so as to ease comparison. For fair comparison, these objective values are a sum of all the objective factors and do not include the weights.

In the convergence trend for the Bilog optimization, it is found that the NNEGA has produced a design which has an objective value that is lower than that for the standard GA. However, in Fig. 3 it can be seen that the NNEGA has found a better design than that found by a standard GA which is not practical due to the discontinuity. The AF at low frequency is close to the target objective for the standard GA: this compensates for the discontinuity at 175 MHz, resulting in a better overall objective value. While the optimized design achieved by the NNEGA meets the target objective from 100 MHz to 240 MHz, but deviates at low frequency thus summing up to a lower objective value. This gives an indication that a simple summing of objective factors is often insufficient for representing the search space. It can also be seen that for a standard GA, the objective value always rises, since the best objective value is never discarded from the population. However, in the NNEGA optimization, due to the change in representation of the

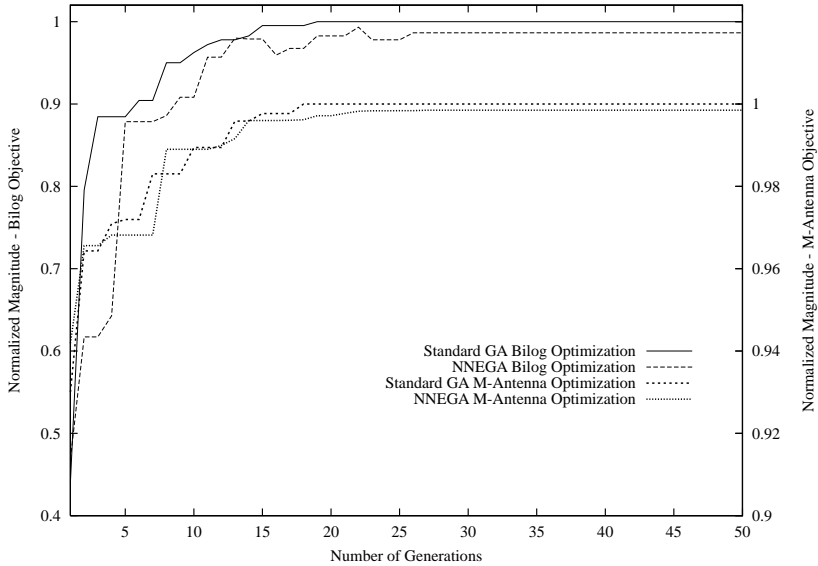


Figure 6. Convergence trend of standard GA and NNEGA optimizations.

search space or objective function, the unweighted objective value can decrease. This indicates a change in the search space and possibly the removal of undesirable designs.

In the M-Antenna optimization, in order to satisfy the requirement for a 10 V/m field strength, the power requirement (least important factor) is compromised. This resulted in an overall lower objective value achieved by the NNEGA but a more desirable design. For each of the case studies, the final objective value achieved by both the standard GA and the NNEGA do not differ much. It gives an indication that both solutions (the standard GA and the NNEGA solution) lie on the Pareto front [22] of the problem but at different locations on the front.

From the convergence trend, it can also be observed that the NNEGA tends to converge about 5 to 10 generations after the standard GA. This is expected due to the continuous change in the objective function. This delay in convergence, although not ideal, requires significantly less computation than a Pareto GA. The frequency of updating the weights is a delicate issue, and needs to be dealt with in further study. However, from initial examination, about 5 updates before the solution converges and about 100 to 200 for the training pattern to the NN before update, has been found to give good results for the two case studies.

4. CONCLUSION

In this paper, the NNEGA, a new evolutionary learning optimization technique, has been developed. This technique uses the learning ability of a neural network and the discarded information generated by the genetic algorithm to optimize the objective function used by the GA. The NN runs in-situ with the GA: after each generation of the GA, the data generated is passed to the NN which learns from the data so as to identify trends and compromise weights in order to achieve an optimized objective function. The weights learnt by the NN are only fed back into the GA after a certain number of generations have elapsed so as to ensure convergence.

Practical antenna design problems are examined. In the Bilog design, the NNEGA was able to identify trends and alter the weights so as to reduce an unwanted discontinuity in the design. For the M-Antenna design, the weights were adjusted and compromised such that when the most important factor has been achieved, the weight was decreased and redistributed to other factors that were more difficult to achieve so as to put more emphasis on them. In both design problems, the weights were observed and found to give a good indication of the behavior of each objective factor, that is, the ease or difficulty of satisfying each factor relative to others.

The adjustment and compromising of weights could be done manually by running several GA optimizations on the problem and then by identifying any trends. The advantage of using the NNEGA is that this process can be done by the neural network automatically. The NNEGA is able to generate an optimized objective function for the standard GA, thus achieving improved designs. NNEGA is an optimization technique that is able to generate an optimized objective function for use with the GA.

In the NNEGA, since the weights in the objective function are determined dynamically through a learning interface, there are two distinct advantages of the present formulation over traditional methods. Firstly, it is superior to penalty-function incorporated optimization problems. This is illustrated in case two in this paper. Secondly, the scalarization of multi-criteria objective functions is often very challenging and the results can only be interpreted in Pareto space. Therefore, the NNEGA optimization methodology also estimates the need for explicit Pareto-front construction in multi-objective optimization problems. The success of this dynamic objective function construction mainly depends on the choice of learning mechanism and its integration within the optimization process. For the two case studies presented, an update every 5 generation performs well.

However, the feedback of weights from the NN to the GA is dependent on the problem type and population size. Further study into the effect of feedback on the rate of convergence for different problems will be required.

REFERENCES

1. Corne, D. and J. L. Shapiro, *Evolutionary Computing*, Balin, Springer, 1997.
2. Goldberg, D. E., *Genetic Algorithms in Search, Optimization & Machine Learning*, Addison-Wesley, Reading, MA, 1989.
3. Johnson, J. M. and Y. Rahmat-Samii, "Genetic algorithms in engineering electromagnetics," *IEEE Antennas Propagat. Mag.*, Vol. 39, No. 4, 7–21, Aug. 1997.
4. Beale, R. and T. Jackson, *Neural Computing: An Introduction*, Institute of Physics Publishing, Bristol & Philadelphia, 1991.
5. Patterson, D. W., *Artificial Neural Networks: Theory and Applications*, Prentice Hall, Singapore, 1995.
6. Booker, B., D. E. Goldberg, and J. H. Holland, "Classifier systems and genetic algorithms," *Artificial Intelligence*, Vol. 40, 235–282, 1989.
7. Schaffer, J. D., D. Whitley, and L. Eshelman, "Combinations of genetic algorithms and neural networks: a survey of the state of the art," *Combination of Genetic Algorithms and Neural Networks*, 1–37, IEEE Computer Society Press, Los Alamitos, CA, 1992.
8. Montana, D. J. and L. Davis, "Training feedforward neural networks using genetic algorithms," *Proc. of 11th Int. Joint Conf. on Artificial Intelligence*, 762–767, Morgan Kaufmann, San Mateo, CA, 1989.
9. Chen, W. C., N. B. Chang, and J. C. Chen, "GA-based fuzzy neural controller design for municipal incinerators," *Fuzzy Sets & Systems*, Vol. 129, No. 3, 343–369, Aug. 2002.
10. Whitley, D., T. Starkweather, and C. Bogart, "Genetic algorithms and neural networks: optimizing connections and connectivity," *Parallel Computing*, Vol. 14, 347–361, 1990.
11. Xin, Z. H. and H. J. Zhang, "Neural network and genetic algorithms for topology optimization of the CCS7 network," *Int. Trans. in Operational Research*, Vol. 9, No. 4, 427–436, Jul. 2002.
12. Koza, J. R. and J. P. Rice, "Genetic generation of both the weights

- and architecture for a neural network,” *Int. Joint Conf. on Neural Networks*, 397–404, Seattle, USA, July 1991.
13. Capi, G., Y. Nasu, L. Barolli, K. Mitobe, M. Yarnano, and K. Takeda, “A new gait optimization approach based on genetic algorithm for walking biped robots and a neural network implementation,” *Trans. of the Information Processing Society of Japan*, Vol. 43, No. 4, 1039–1049, April 2002.
 14. Lee, Y. H., B. J. Cahill, S. J. Porter, and A. C. Marvin, “Optimization of the cost function for genetic algorithms using neural networks applied to antenna design,” *Proc. of International Conference on Antenna and Propagation*, Manchester, U.K., 2001.
 15. Widrow, B. and M. E. Hoff, *Adaptive Switching Circuits*, IRE WESCON Convention Record, New York, 1960.
 16. Eshelman, L. J., R. A. Caruana, and J. D. Schaffer, “Biases in the crossover landscape,” *Proc. 3rd Int. Conf. on Genetic Algorithms*, 10–19, Washington, USA, June 1989.
 17. Goldberg, D. E. and K. Deb, “A comparative analysis of selection schemes used in genetic algorithms,” *Foundations of Genetic Algorithms*, 69–93, Morgan Kaufmann, San Mateo, USA, 1991.
 18. Porter, S. J. and A. C. Marvin, “A new broadband EMC antenna for emissions and immunity,” *Int. Sym. on Electromagnetic Compatibility*, 75–79, Rome, Italy, Sep. 1994.
 19. Paul, C. R., *Introduction to Electromagnetic Compatibility*, John Wiley & Sons, New York, 1992.
 20. Lee, Y. H., A. C. Marvin, and S. J. Porter, “Near field EMC antenna design using genetic algorithms,” *Millennium Conference on Antennas & Propagation*, Davos, Switzerland, April 2000.
 21. Balanis, C. A., *Antenna Theory*, 443, John Wiley & Son, New York, 1982.
 22. Horn, J., N. Nafpliotis, and D. E. Goldberg, “A niched Pareto genetic algorithm for multiobjective optimization,” *Proc. 1st IEEE Conference on Evolutionary Computation IEEE World Congress on Computational Intelligence*, Vol. 1, 67–72, IEEE Service Center, Piscataway, NJ, 1994.

Y. H. Lee received her Ph.D. from the University of York, U.K., in July 2002. She received her B.Eng. and M.Eng. from Nanyang Technological University, Singapore in 1992 and 1996 respectively. She then Joined Nanyang Technological University and is currently an Assistant Professor in the School of Electrical and Electronic

Engineering. She is a member of IEEE and IEE. Her current research interest includes optimization based on heuristic techniques, computational electromagnetic and antenna design for electromagnetic compatibility testing purposes.

B. J. Cahill received his Ph.D. from the University of York in July 2002. He is currently the Chief Technical Officer in SkyLINC. Pte. Ltd. Dr. Cahill is an expert in RF systems and has held positions in RACAL and the University of York. He has worked on military and commercial projects in both Electromagnetic Compatibility and Electronic Warfare.

S. J. Porter received the B.Sc. and D.Phil. degrees from the Department of Physics, University of York, U.K., in 1985 and 1991, respectively. He is a Lecturer and Member of the Applied Electromagnetic Research Group at the Department of Electronics, University of York. His research interests include computational electromagnetics, particularly as applied to radio frequency and microwave problems, computational and computer-aided tools for electromagnetic compatibility design, antenna design, application of evolutionary computation optimization methods to antenna design and computational acoustics.

A. C. Marvin received the M.Eng. and Ph.D. degrees from the University of Sheffield, U.K., in 1974 and 1979, respectively. He is a Professor of applied electromagnetics at the University of York, U.K., and Technical Director of York EMC services, Ltd. His research interest include electromagnetic compatibility measurement techniques, electromagnetic shielding measurements, electromagnetic compatibility antenna design, and equipment design techniques for electromagnetic compatibility. Dr. Marvin is currently Chairman of the Management Committee of COST Action 261 and contributes to various national and international standards committees. He is a member of the IEE and a Chartered Engineer in the U.K.