Flame Classification through the use of an Artificial Neural Network trained with a Genetic Algorithm

Juan C. Gómez[1], Fernando Hernández[1], C.A. Coello Coello[2], Guillermo Ronquillo[1] and Antonio Trejo[1]

[1] Applied Research Management, CIDESI, Queretaro, Mexico
[2] Computer Science Department, CINVESTAV-IPN, Mexico

Abstract. This paper introduces a Genetic Algorithm (GA) for training Artificial Neural Networks (ANNs) using the electromagnetic spectrum signal of a combustion process for flame pattern classification.Combustion requires identification systems that provide information about the state of the process in order to make combustion more efficient and clean. Combustion is complex to model using conventional deterministic methods which motivates the use of heuristics in this domain. ANNs have been successfully applied to combustion classification systems; however, traditional ANN training methods get often trapped in local minima of the error function and are inefficient in multimodal and non-differentiable functions. A GA is used here to overcome these problems. The proposed GA finds the weights of an ANN than best fits the training pattern with the highest classification rate.

Key words: Genetic Algorithms, Artificial Neural Networks, Flame Classification, Electromagnetic spectrum

1 Introduction

Currently, combustion is the most important source of energy for power generation, heating, and transportation in the world, and this trend is expected to continue in the foreseeable future [1]. Control systems that provide information about combustion are of great importance. However, combustion is a dynamic, highly nonlinear and multivariable process, which is particularly complex to model using conventional deterministic methods.

Diagnostic methods based on monitoring flames have been implemented as strategies to provide status of the combustion process with which one can implement control and optimization systems to make more efficient the combustion process, optimizing fuel consumption and reducing emissions. Several monitoring flame techniques have been developed for combustion processes using Fuzzy Logic [2], Expert systems [1], Support Vector Machines [3], Artificial Neural Networks (ANNs) [4] and Genetic Algorithms (GAs) [5, 6], focussing mainly on combustion gases analysis and prediction.

Monitoring flames through spectral analysis approaches arises as an alternative to monitoring techniques such as image analysis, which are difficult to implement in combustion systems and require more computer processing. Combustion processes such as those occurring in the power generation industry frequently employ optical sensors as a safety measure indicating the presence or absence of the flame inside the furnace. These sensors could provided more information about the flame state that can be used for combustion optimization.

GAs are heuristic search methods based on the mechanism of genetics and natural selection. GAs require minimum specific domain knowledge about the search space, which makes their use very general. GAs are also easy to use and can be particularly useful when dealing with optimization problems having a very large, complex and little known search space, in which traditional mathematical programming techniques tend to fail [7].

Performance of ANNs is largely influenced by the architecture as well as by the weights used for its connections. The training stage in an ANN is the process of adjusting the weights such that the training patterns fit with the lowest error while having a profitable generalization ability to recognize new patterns. Traditional training methods for ANNs are based on gradient descent and get often trapped in local minima of the error function. Therefore, such methods are very inefficient in multimodal and non-differentiable functions [8]. In such cases, the use of metaheuristics such as a GA is more appropriate. The GA proposed here aims to adapt the parameters of an ANN including the weights of its connections [12], its architecture [13–15], its learning rules and its transfer functions [16].

The study reported here focuses on the electromagnetic spectrum signal analysis and on the use of GAs to train an ANN for the classification of pattern flames of a combustion process.

The remainder of this paper is organized as follows. In Section 2, we describe the methodology adopted for our study, including a description of the data acquisition and the features extraction processes of the electromagnetic spectrum. In Section 3, we describe the main features of the GA that is used to train an ANN and we also provide a description of the experimental design adopted. Our results are shown in Section 4 and our conclusions and some possible paths for future research are provided in Section 5.

2 Methodology

This section provides a description of the methodologies that have been used for flame classification. In Figure 1, we show a general diagram of the system adopted in our study, which includes three main stages: 1) data acquisition, 2) features extraction and 3) the use of an ANN trained by a GA.

2.1 Data Acquisition

In our study, the electromagnetic spectrum of a combustion process was measured using a flame scanning system with a solid-state optical sensor that op-
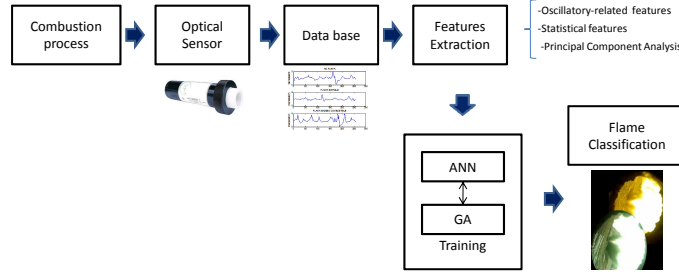
Fig. 1: Diagram of the system adopted in our study.

erates between the ultraviolet peak at 350 nm and the infrared peak at 700 nm. The scanning system output is 450 hexadecimal data, containing the sensor configuration and the flame signal in both the time and the frequency domains.

The most important condition of the combustion process optimization is the air/fuel ratio. Particularly, there are three conditions related to this ratio balance: fuel rich, fuel lean and air/fuel balance. In Figure 2, we show the signals associated to the following flame states:

1. No flame (background radiation)
2. Stable flame (air fuel balance)
3. Flame with air excess (fuel lean)
4. Flame with fuel excess (fuel rich)

A database was created using signals of the four flame states using a program written in Visual Basic for data acquisition. The database is composed of 480 signals (we stored 120 for each flame pattern). Each signal contains 256 values corresponding to the voltage equivalent to the flame intensity in 500 ms.

The database was divided in three subsets:

1. *Data Training:* Data used for training our ANN (see Section 3). The quadratic error is minimized in the fitness function of the GA adopted to train the ANN. This data corresponds to 50% of the total data set.
2. *Data Validation:* Data for computing the percentage of generalization. This data corresponds to 20% of the total data set.
3. *Data Test:* New data to test the ANN. This data corresponds to 30% of the total data set.

2.2   Features Extraction

Flame signals were preprocessed to extract features that capture the whole possible information (e.g., trends, periodicity, signatures of chaos) required to describe the flame patterns. We provide next a description of the formulation of these features.
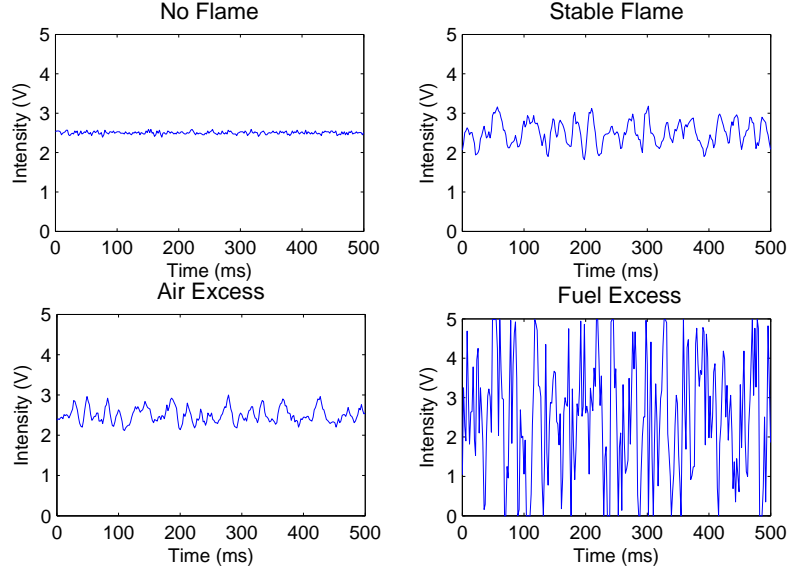
Fig. 2: Signal of the four flame patterns considered in our study

2.3 Statistical Features

Statistical Moments Statistical moment analysis is a technique that can be used for data series characterization, since it gives a set of parameters that describe and provide information of a probability distribution function. The second, third, and fourth normalized central moments of the distribution of the flame signal intensity were calculated in order to provide information derived from the comparison of the shape of the electromagnetic spectra.

The formal definition of a statistical moment is:

$$M_k = E[x - E(x)]^k \tag{1}$$

Where:
$M_k = k$ order statistical moment
$E[\ ] =$ Expected value
$x =$ Data signal

Autocorrelation sum (Box-pierce) Autocorrelation measures the linear correlation in a time series. The autocorrelation sum is calculated as:

$$Q(\tau_{max}) = n\Sigma_{\tau=1}^{\tau_{max}} r(\tau)^2 \tag{2}$$

where:

$$r(\tau) = \frac{\sum_{t=\tau+1}^{n} (x_t x_{t-\tau} - \overline{x}^2)}{\sum_{t=\tau+1}^{n} x_t^2 - \overline{x}^2} \tag{3}$$

2.4 Oscillation-related Features

Oscillation-related features have been applied considering an oscillating behavior in the flame signal (although this is not always true) and are calculated as in [9]. The signal is spanned with a data window of length k, and we checked if the center is either a minimum or a maximum. The oscillation period is defined as the time between successive peaks. The Oscillation-related features calculated are the mean, and standard deviation of the period and peak, which are defined as:

Period average:

$$\bar{T} = \frac{1}{n}\Sigma_{i=1}^{n}T_i \tag{4}$$

where:
T_i= Period of the ith oscillation.

Peak average:

$$\bar{z} = \frac{1}{n}\Sigma_{i=1}^{n}z_i \tag{5}$$

where:
z_i=Peak of the ith oscillation.

Period standard deviation

$$S_T = \sqrt{\frac{1}{n-1}\Sigma_{i=1}^{n}(T_i^2 - \bar{T}^2)} \tag{6}$$

Peak standard deviation

$$S_z = \frac{1}{n-1}\Sigma_{i=1}^{n}(z_i - \bar{z}^2) \tag{7}$$

2.5 Principal Components Analysis

Principal Components Analysis (PCA) is a data transformation technique that can be useful to reveal simple structures, patterns or tendencies underlying in complex data sets using analytical solutions. This technique provides a measure to quantify the relative importance of each dimension allowing the characterization of large data sets with a reduced number of components.

Let X be a $(m \times n)$ matrix and X^tX a quadratic matrix of range q. Then, X could be expressed as:

$$X = U\Sigma V^t \tag{8}$$

where U and V are m order matrices containing the eigenvector of X^tX and Σ is a diagonal matrix that contains the square roots of the eigenvalues of X^tX: $(\sigma_1, \sigma_2, \sigma_3, \ldots, \sigma_q)$, with $\sigma_1 \geq \sigma_2 \geq \sigma_3 \geq, ..., \sigma_q > 0$.

In this study, we first compute the distance matrix of the data of a flame signal and then, PCA is applied.

Principal Components Selection It is expected that keeping $n \ll m$ components produces a high variance of the original data set. Then, the number of components to retain is based on the cumulative contribution of the variance of the first several components, which can be expressed as:

$$CV_k = \sum_{i=1}^{k} \frac{100\lambda_i}{\sum_{j=1}^{m} \lambda_j} \tag{9}$$

where:
CV_k = cumulative variance of the component k
m = Total number of components

In Figure 3, we show the cumulative variance of the first 20 principal components of a flame signal. As we can see, the first five components explain the 92.7% variability percentage and the 6th component increases it by only 1.28%. Therefore, since the first five components have a high percentage of variability, only these are retained.
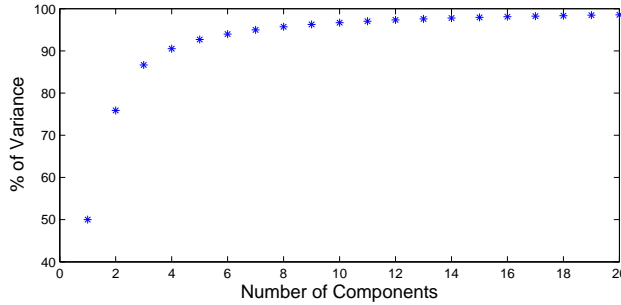

Fig. 3: Cumulative variance of the first 20 principal components

3   Genetic Algorithm Parameters

A real-coded GA was implemented, together with a two-layer perceptron architecture using a hyperbolic tangent transfer function in both the hidden and the output layers. Figure 4 highlights the architecture of the ANN and in Figure 5 we show the weights encoding scheme adopted.

The inputs vector of the ANN is formed by the 13 features described in the previous section, while the outputs correspond to the four different flame patterns being considered. In Figure 6, we show the targets for the flame states.

The pseudocode of the GA that we implemented is depicted in Algorithm 1.

Algorithm 1 Pseudocode of the GA used to train our ANNs

Require: Population size N, Maximum number of generations G
Ensure: Trained artificial neural network
 1: Load features extracted from flame signals.
 2: Normalize features.
 3: Perform data training, data validation and data testing.
 4: Initialize the population P_i of N individuals:
 5: $k = 1$
 6: **repeat**
 7:   Generate random weights for the adjacency matrix of ANN_k .
 8:   Define the first chromosome with a concatenation of the weights of the hidden layer of the adjacency matrix.
 9:   Define the second chromosome with a concatenation of the weights of the output layer of the adjacency matrix.
10:   Perform an elimination of the connection weights using the connection elimination operator with a probability of 0.35.
11:   $k = k + 1$
12: **until** $k = N$
13: $i \leftarrow 0$
14: **repeat**
15:   Evaluate fitness of population P_i.
16:   Perform roulette wheel selection
17:   Generate offspring P'_i.
18:   Apply mutation operator to P'_i.
19:   Apply the elimination connection operator to P'_i with a probability of $\frac{P_{mut}}{2}$.
20:   Apply elitism$P_{i+1} \leftarrow P'_i$.
21:   $i \leftarrow i + 1$
22: **until** Termination condition is reached

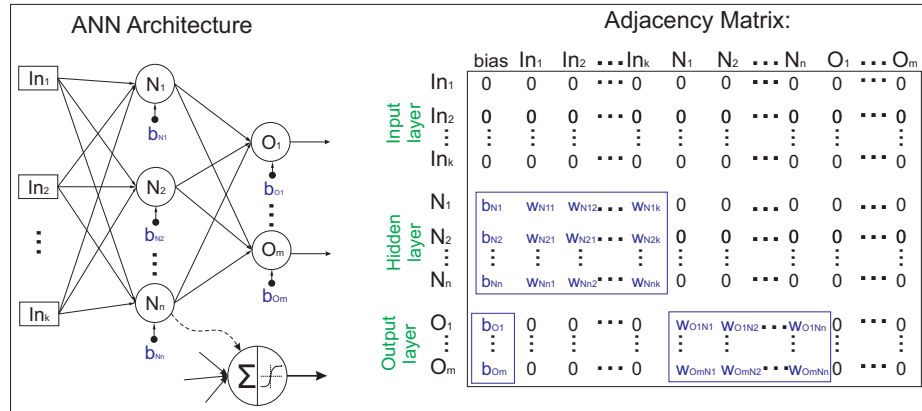
Fig. 4: Architecture of the ANN adopted and the representation as a adjacency matrix.
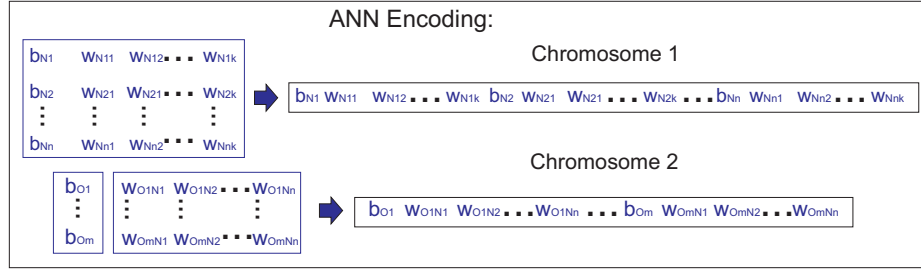
Fig. 5: Adjacency matrix encoding in our GA.

Our GA uses elitism (the best individual from each generation passes intact to the next one), as well as roulette wheel selection, arithmetic crossover [10] and uniform mutation. Each of the main elements of our GA are briefly described next.

Fig. 6: Targets for the four classes of flames during the training of the ANN.

3.1 Initial population

The initial population is created with randomly generated real values in the range $[-50, 50]$ for both chromosomes. Then, a connection elimination operator in applied. This operator sets the weights equal to zero with a probability of 0.35.

3.2 Fitness function

The objective function commonly used to adjust the weights of an ANN is the mean squared error (MSE). However, this is not necessarily the best choice when using a GA. In our study, we adopted a different scheme in which we aim to find the weights of an ANN that provide a good generalization performance while also providing the best matching with respect to the training set.

Thus, the fitness function adopted by our GA is:

$$Fitness = \| \prod_{i=1}^{n}(1 + e_i^2), \sum_{i=1}^{n} e_i^2, E_{test} \| \qquad (10)$$

where:

e_i=Error associated to the training data set i

E_{test}=Percentage of misclassifications in the test data set

n=Size of the training data set

From now on, this fitness function will be referred to as the *error norm*.

3.3 Genetic Operators

As indicated before, we adopted roulette wheel selection with a probability of 0.9.

We also incorporated uniform crossover, which is defined as follows:

Let's consider the following two parents F_1 and F_2:

$$F_1 = \langle v_1, \ldots, v_k, \ldots, v_m \rangle \tag{11}$$

$$F_2 = \langle w_1, \ldots, w_k, \ldots, w_m \rangle \tag{12}$$

Their offspring are generated, using:

$$O_1 = \langle av_1 + (1-a)w1, \ldots, a \times w_k + (1-a)w_k, \ldots, a \times v_m + (1-a)w_m \rangle \tag{13}$$

$$O_2 = \langle aw_1 + (1-a)v1, \ldots a \times w_k + (1-a)v_k, \ldots, a \times w_m + (1-a)v_m \rangle \tag{14}$$

In our case, we adopted $a = 0.6$.

We also adopted uniform mutation with a probability $P_{mut} = 0.05$.

Given an individual P, the mutated version is:

$$P' = \langle v_1, \ldots, v'_k, \ldots, v_m \rangle \tag{15}$$

where

$$v'_k = \begin{cases} v_k + m_k & \text{if } LB \leq v_k + m_k \leq UB, \\ v_k - m_k & \text{other case.} \end{cases} \tag{16}$$

and:

$m_k = rand(LB, UB)$ and $[LB, UB]$ are the lower bound (LB) and upper bound (UB) of v_k, which is the original position of the individual to be mutated.

Finally, we also adopted the elimination connection operator, in order to allow the remotion of connections during the evolutionary search. This operator was applied with a probability of $\frac{P_{mut}}{2}$ (except for the initial generation in which a higher probability was used, as indicated before).

3.4 Experimental Design

The 13 features extracted from the database of four experimental flames patterns were linearly normalized and were used as the inputs of our ANN. The GA was

tested with a population of 30 ANNs having 10 neurons in the hidden layer. First, MSE was used as our fitness function and then, we adopted the fitness function defined in equation (10).

The stopping criterion for all the runs of the GA was to reach the best possible fitness value (i.e., $Fitness = 1$ for equation (10) and $Fitness = 0$ for MSE), or when reaching 2000 generations (whatever happened first).

The results obtained from the GA when using equation (10) were compared with respect to those produced by the scaled conjugate gradient method (SCG) [11], which is a traditional approach for training ANNs. Our results are presented next.

4 Discussion of Results

In Figure 7, we plot the fitness values versus the generation number. We show there the results corresponding to the best individual found in a run of the GA using equation (10). This plot shows how, in a few generations, an individual with a fitness value of one (i.e., the best possible value) was found.
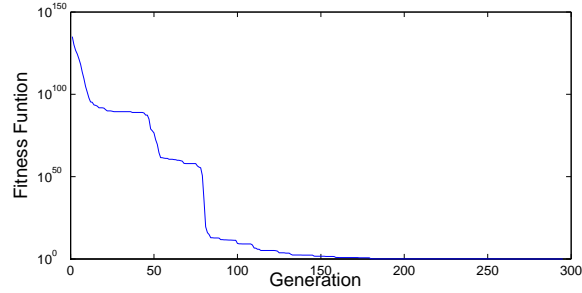

Fig. 7: Fitness function of the best individual as defined in equation (10)

Table 1: ANN training with GA. Average of 10 runs

	average % correct classification	average generations
MSE	98.1712963	2000
Norm	99.5138889	913.333333

In Table 1, we provide the results of the ANN training when using MSE as the fitness function. In Figure 8, we show a comparison of the MSE of the best individual using both fitness functions. As we can see, the use of MSE needs more
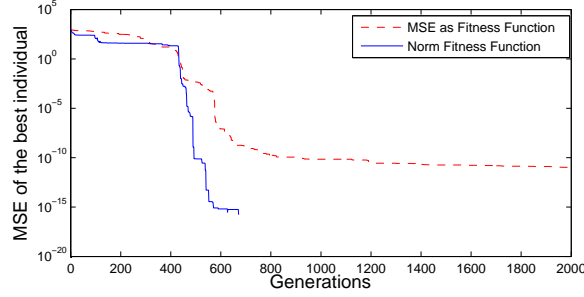
Fig. 8: Comparison of the MSE with the two different fitness functions adopted in our study

generations to reach an acceptable fitness value, whereas the use of equation (10) provides good results with a lower number of generations.

The results of 30 runs of the GA with the fitness function defined in equation (10) are given in Table 2. A fitness value equal to 1 was reached, on average, after 881.4 generations. Having a low average MSE implies a good fit with the data training set, and having a high percentage of correct classification corresponds with a good generalization ability. As Table 3 indicates, our results are better than those obtained using SCG (this approach produced higher MSE values than our GA using the fitness function defined in equation (10)).

Table 2: Average of 30 independent runs of ANN training using a GA

	Average MSE	Total misclassified signals	Average % Correct Classification	Average of misclassified signals
Class 1	5.7628E-17	2	99.9444444	0.06666667
Class 2	6.8257E-15	52	98.5555556	1.73333333
Class 3	1.421E-13	13	99.6388889	0.43333333
Class 4	1.0938E-17	0	100	0
Total data set	2.3157E-15	67	99.5347222	2.23333333

In Table 5, we compare the best and worst results of both training algorithms. The best results obtained by our proposed GA significantly outperform the results obtained by SCG.

Table 3: ANN training using SCG. Average of 30 independent runs

	Total misclassification	Average % Correct Classification	Average of misclassified signals
Class 1	120	96.66666667	4
Class 2	121	96.63888889	4.03333333
Class 3	1	99.97222222	0.03333333
Class 4	246	93.16666667	8.2
Total data set	488	96.61111111	4.06666667

Table 4: MSE results with Scaled Conjugate Gradient method

	Average MSE	Average %error
training	0.0084154	3.26385333
validation	0.0088272	3.33308833
test	0.00959884	3.63403133

Table 5: Comparison of the best and worst results obtained with a GA and with the Scaled Conjugate Gradient method

	MSE training	MSE Validation	MSE Test	% Correct Classification
AG best result	6.77E-18	7.62E-18	9.01E-16	100
AG worst result	2.77E-13	1.66E-09	9.66E-01	98.95833
SCG best result	9.25E-08	8.81E-08	3.30E-04	100
SCG worst result	6.73E-02	6.10E-02	6.10E-02	74.8

5   Conclusions and Future Work

A Genetic Algorithm was developed to train Artificial Neural Networks for flames classification using the electromagnetic spectrum. The proposed GA was compared with respect to the use of the Scaled Conjugate Gradient method in the training of artificial neural networks. Our preliminary results indicate that our proposed approach has a better performance, since it generates solutions with a lower error and an improved generalization ability. Additionally, our results show that the features extracted from signal spectra could provide information about the combustion state and could be used for flame characterization and combustion monitoring. All flame classes were classified with a high percentage while using ANNs trained with our proposed GA.

As part of our future work, we are considering the use of ANNs for the classification of signals of a combustion process in power generation systems, in which there is a more complicated dynamics. We are also interested in evolving weights connections of different ANN architectures, such as recurrent ANNs and generalized multilayer perceptrons, using genetic algorithms.

The results show that the features extracted from signal spectra could provide information about the combustion state and could be useful for flame characterization and combustion monitoring.

Acknowledgments

References

1. Mahajan, A.,& Mahajan, R. (2004). Expert system for flame analysis. In Cybernetics and Intelligent Systems, 2004 IEEE Conference on (Vol. 2, pp. 1160-1165). IEEE.
2. Xu L, Yan Y, Cornwell C, Riley G. On-line fuel identification using digital signal processing and fuzzy inference techniques. IEEE Transactions on Instrumentation & Measurement 2004;53(4):131620.
3. Torres, C. I., Hernández, F., Trejo, A., & Ronquillo, G. Support Vector Machines Applied to a Combustion Process.
4. Li, X., Sun, D., Lu, G., Krabicka, J., & Yan, Y. (2012, July). Prediction of NOx emissions through flame radical imaging and neural network based soft computing. In Imaging Systems and Techniques (IST), 2012 IEEE International Conference on (pp. 502-505). IEEE.
5. Hao, Z., Qian, X., Cen, K., & Jianren, F. (2004). Optimizing pulverized coal combustion performance based on ANN and GA. Fuel Processing Technology, 85(2), 113-124.
6. Kesgin, U. (2004). Genetic algorithm and artificial neural network for engine optimisation of efficiency and NOx emission. Fuel, 83(7), 885-895.
7. Palmes, P. P., Hayasaka, T., & Usui, S. (2005). Mutation-based genetic neural network. Neural Networks, IEEE Transactions on, 16(3), 587-600.

8. Yao, X. (1999). Evolving artificial neural networks. Proceedings of the IEEE, 87(9), 1423-1447.
9. T simpiris, A., & Kugiumtzis, D. (2012). Feature selection for classification of oscillating time series. Expert Systems, 29(5), 456-477.
10. Z. Michalewicz, Genetic Algorithms + Data Structures = Evolution Programs, 2nd ed., Springer- Verlag, Berlin, 1994.
11. Moller, M. F. (1993). A scaled conjugate gradient algorithm for fast supervised learning. Neural networks, 6(4), 525-533.
12. Sexton, R. S., Dorsey, R. E., & Johnson, J. D. (1998). Toward global optimization of neural networks: a comparison of the genetic algorithm and backpropagation. Decision Support Systems, 22(2), 171-185.
13. Rivero, D., Dorado, J., Fernndez-Blanco, E., & Pazos, A. (2009). A genetic algorithm for ANN design, training and simplification. In Bio-Inspired Systems: Computational and Ambient Intelligence (pp. 391-398). Springer Berlin Heidelberg.
14. Yao, X., & Islam, M. M. (2008). Evolving artificial neural network ensembles. Computational Intelligence Magazine, IEEE, 3(1), 31-42.
15. Ballabio, D., Vasighi, M., Consonni, V., & Kompany-Zareh, M. (2011). Genetic algorithms for architecture optimisation of counter-propagation artificial neural networks. Chemometrics and intelligent laboratory systems, 105(1), 56-64.
16. Annunziato, M., Bertini, I., Lucchetti, M., & Pizzuti, S. (2003). Evolving weights and transfer functions in feed forward neural networks. Proc. EUNITE2003, Oulu, Finland.