Asociación Argentina
de Mecánica Computacional

Mecánica Computacional Vol XXXIII, págs. 1557-1570 (artículo completo)
Graciela Bertolino, Mariano Cantero, Mario Storti y Federico Teruel (Eds.)
San Carlos de Bariloche, 23-26 Setiembre 2014

NATURE INSPIRED CURVE FITTING STRATEGIES FOR VISCOELASTIC MATERIALS MECHANICAL PROPERTIES

Waldir F. Filho and Flávio S. Barbosa

Federal University of Juiz de Fora, Juiz de Fora, Brazil, Graduation Program in Computational Modeling, waldir.felippe@gmail.com, flavio.barbosa@ufjf.edu.br

Keywords: Curve fit, Viscoelastic Materials, Structural dynamics.

Abstract. Viscoelastic models have been used to solve problems in different areas, as in structural damping problems or to simulate the mechanical behavior of biological tissues.

In order to properly simulate the behavior of these materials, the frequency dependency of theirs mechanical properties must be taken into account. In that way, one of the first steps to achieve a fair viscoelastic model is to curve fit mechanical parameters, adjusting experimental data from characterization tests. Traditionally this curve fitting procedure is made through minimum least squares methodologies.

It this work, three alternative curve fitting strategies for viscoelastic materials are studied: Artificial Neural Networks, Genetic Algorithms and Particle Swarm Optimization. These strategies are analyzed and the quality of each curve fitting procedure, based on real experimental data, is evaluated pointing the advantages and disadvantages of each methodology.

1 INTRODUCTION

Due to the frequency dependence of mechanical properties, time domain based models for viscoelastic materials (VEM) are not as numerous as frequency domain methods. In spite of that, because the facilities that time domain methods may directly provide, such as the maximum displacement range in a structural model analysis, many researchers have been developing numerical methods to simulate the dynamical response of VEM in time domain. The most successful models are the ones that introduce extra dissipation coordinates or internal variables in a Finite Element model. Due to its simplicity and capability to virtually model any complex geometry, this kind of methodology has been applied in several situations such as the ones presented by Wang et al. (2000), Roy et al. (2008), Friswell et al. (2010) and Wang and Inman (2013). Among the dissipation coordinates based methods it is possible to observe that Golla-Hughes-McTavish (GHM) method (Golla and Hughes, 1985; McTavish and Hughes, 1993) and Anelastic Displacement Field (ADF) method (Lesieutre and Mingori, 1990; Lesieutre, 1992; Lesieutre and Bianchini, 1993; Lesieutre and Govindswamy, 1996; Lesieutre and Lee, 1996) are frequently chosen in order to simulate the dynamic response of VEM.

The first step, in order to properly simulate the dynamical behavior of VEM, is to characterize the material's dynamical properties. There are several methodologies to do it, such as: ASTM method (ASTM, 1993), Direct Method (Faisca et al., 2001), Indirect Method (Masterson and Miles, 1995), using transmissibility functions (Soula et al., 1997) and measured strains (Mousavi et al., 2004), although the ASTM method is the most employed. In general, these characterization tests excite dynamically specimens registering its behavior along the time in order to determine the value of complex modulus at a specific frequency.

The characterization tests must be carried out several times at each analyzed frequency, due to tests errors. These errors may appear because of materials' mechanical properties variability; geometric variability of specimens; reading errors, such as signal saturation, low ratio signal/noise, intermittent noise, spurious trends, influence of electric power distribution network frequency and singular points or other external interferences. It leads the results to show dispersions, in other words, a mean value and a standard deviation at each frequency analyzed. Once the characterization process is done, the next step is to find the parameters that adjust the experimental curves.

The most usual curve fit methods are the ones based in the least squares theory, as could be seen in the works of Hillström et al. (2000), Barbosa and Farage (2008) and Felippe et al. (2012). Another possibility is to use methods like Artificial Neural Networks (ANN), Genetic Algorithms (GA) and Particle Swarm Optimization (PSO). They may achieve good results without the typical problems of the usual methods, such as being trapped in local minimums or the need to ensure the same number of parameters to be fitted and experimental points.

In this work, three curve fitting strategies based in nature-inspired algorithms will be analyzed, namely ANN, GA and PSO, and the main features of each method will be pointed. They will be applied to viscoelastic modelling with GHM model.

2 CURVE FITTING PROBLEM

Traditionally, the curve fit problem searches for specific coefficients from a closed form function in order to approximate it to given data points. It is a process that could be summarized by the balance between two conflicting elements: Accuracy and Simplicity. These elements, accuracy and simplicity, can be measured through the distance between the fitted curve and the data points and by the number of parameters in the curve, respectively.

At first glance, one would chose a curve with a large number of parameters, which will ensure a great fit at data set. In this way, the fitted curve could be overfitted and the model will miss its predictive function, trends and regularities could fall away. On the other hand, curves with few parameters can make false assumptions about the true behaviour of the model. The best way, then, is to chose a simple model (a model with few parameters) that presents a reasonable fit to the data.

The usual way to measure the Accuracy of a curve fit could be stated by Equation (1), the error metric function:

$$f(\mathbf{X}) = \sqrt{\sum_{i=1}^{N_{pts}} \left(h(\mathbf{X}, \omega_i) - \overline{h}_i\right)^2}, \tag{1}$$

where ω_i is the $i-th$ abscissa of the data point set; $h(\mathbf{X}, \omega_i)$ is the equation tho be fitted, written as a function of whose parameters in vector $\mathbf{X}$ and ω_i; $\overline{h}_i$ is the $i-th$ ordinate of data point set and; N_{pts} is the number of data points.

Curve fitting is a preliminary activity to many techniques to model and solve problems, such as simulation, predictive modeling, and statistical inference. The VEM modeling is not different, before any simulation is performed, one needs to determine the parameters of the adopted model. When modeling a VEM with GHM model the curve to be fitted is not one but two and they are given by:

$$G'(\omega) = G^0 + \sum_{j=1}^{N} \alpha_j \frac{\omega^2(\omega^2 - \delta_j + \beta_j^2)}{(\delta_j - \omega^2)^2 + \beta_j^2\omega^2}, \tag{2}$$

$$\eta(\omega) = \frac{1}{G'(\omega)} \sum_{j=1}^{N} \frac{\alpha_j\beta_j\delta_j\omega}{(\delta_j - \omega^2)^2 + \beta_j^2\omega^2}. \tag{3}$$

where ω is the frequency, N is the number of GHM terms, $i = \sqrt{-1}$ and α_j, β_j, δ_j and G^0 are the parameters to be adjusted.

As the GHM model behaviour is described by two Equations, then Equation (1) need to be re-written as:

$$f(\mathbf{X}) = \psi_1 \sqrt{\sum_{i=1}^{N_{pts}} \left(G'(\mathbf{X}, \omega_i) - \overline{G'}_i\right)^2} + \psi_2 \sqrt{\sum_{i=1}^{N_{pts}} \left(\eta(\mathbf{X}, \omega_i) - \overline{\eta}_i\right)^2}, \tag{4}$$

where ψ_1 and ψ_2 are weights to balance the values magnitude of functions $G'(\omega)$ and $\eta(\omega)$. This equation will be employed in GA, PSO and ANN curve fit strategies as described in sections 3.1, 3.2 and 3.3.

3 CURVE FITTING ALGORITHMS

The three approaches considered will be discussed in this section and an investigation about theirs best parameters configuration will be performed. In order to accomplish this investigation, some numerical tests will be performed considering the data set on Table 1.

Frequency (Hz)	$G'(\omega)$ (MPa)	$\eta(\omega)$
3	0.1437	0.2784
50	0.4269	0.5120
100	0.5370	0.3489
250	0.6376	0.1823
300	0.6489	0.1557

Table 1: GA curve fitting results summary.

3.1 Genetic Algorithm curve fit

GA was firstly described by Holland (1975) and DeJong (1975). It is a global heuristics based optimization method to find a point that minimizes a function, called objective function. Once the curve fit problem searches curve parameters in order to minimize de distance between the experimental data and the fitted curve, the GA strategy could be applied to this kind of problem straight forward, as shown in Manela et al. (1993), Gulsen et al. (1995) and Morbiducci et al. (2005).

GA tries to mimic the evolutionary processes observed in Nature, where the best individuals survive and procreate. Under this paradigm, rather than generating a sequence of candidate solutions once at a time, a population of candidate solutions is maintained. Each population member, or individual, is described accordingly to a chromosome, a data structure with the problem parameters to be determined where each of these parameters are called gene. This algorithm starts generating a random initial population. At each iteration, or generation, the individuals are selected by the value obtained accordingly to the objective function, the fitness value. The best individuals are selected to procreate (crossover) and new individuals are generated, substituting the worst ones. Some of the best individuals are maintained (Elitism) and a mutation is imposed to the others. Then the next generation is obtained with the ones created by crossover, mutation and elitism.

In order to curve fit Equations (2) and (3), the objective function will be the one given by Equation (4). The genes will be the model's parameters α_j, β_j, δ_j and G^0 then the chromosomes can be represented in the following vector way:

$$\mathbf{X} = \left\{G^0, \alpha_1, \beta_1, \delta_1, \alpha_2, \beta_2, \delta_2, \ldots, \alpha_j, \beta_j, \delta_j, \ldots, \alpha_N, \beta_N, \delta_N\right\}, \tag{5}$$

where N is the number of terms adopted in Equations (2) and (3).

The chromosomes can be coded in two different ways: binary and real. For each chromosome coding type, there are a large variety of crossover operators algorithms, as shown by Hasançebi and Erbatur (2000), Pendharkar and Rodger (2004) and Deep and Thakur (2007), among others.

In order to determine which chromosome coding and which crossover operator suit better for the problem of curve fitting GHM model's parameters, some preliminary tests were carried out. The characterization data points presented in Table 1, an initial population of 200 individuals and the limit of 500 generations were considered in these tests.

The individuals able to reproduce were selected using the algorithm Stochastic Uniform Selection (SUS) (Baker, 1987). This selection is performed by mapping the individuals into a continuous segment of line, where each individual's segment is proportional to its fitness value (the better fitness gets more space at this line); Equally spaced pointers are placed over the line as many as there are individuals to be selected. If one has n individuals to select, then

the distance between the pointers would be $\frac{1}{n}$ and the position of the first pointer is given by a random number in $[0, \frac{1}{n}]$; The selected individuals are those pointed.

Other algorithm parameters adopted were: Chromosome with 7 genes ($N = 2$); $\psi_1 = 1$ and $\psi_2 = 7.89 \times 10^{-7}$; 80% of the new population is created by crossover; 2 individuals of the new population are Elite individuals and 38 new individuals are obtained by mutation.

Firstly, a real coded chromosome was considered and four crossover algorithms: Single point, Two point, Heuristic and Arithmetic. The single point algorithm works by selecting a random point shared by the two parents chromosomes and then swapping the genes after this point in order to generate two children. This algorithm is the simplest and most used crossover operator. On the Two point crossover, both parents chromosomes are split at two random points, the children are generated swapping the middle part of the parents chromosomes. When using this crossover operator, one can expect poorer performance results because good chromosomes are more likely to be disrupted. On the other hand, when one uses this crossover algorithm, the problem search space could be explored widely on early ages and when population becomes homogeneous, on the last generations, it explores a smaller region which provides an refinement of the solution (Dumitrescu et al., 2000).

The Heuristic crossover generates a child, β^c, taking two parents, β^{p_1} and β^{p_2}, where β^{p_1} has better fitness value, using the following equation:

$$\beta^c = \beta^{p_1} + \alpha \left(\beta^{p_1} - \beta^{p_2}\right), \tag{6}$$

where α is a random number in interval $[0, 1]$. If the new individual is infeasible or its gens are out of the boundaries, then the parent β^{p_1} is taken as the new one. This algorithm generates offspring close to the best parent, with the objective to lead the search process towards the most promising zones in the search space (Herrera et al., 1996).

The Arithmetic crossover takes the weighted sum of two parents chromosomes obtained via:

$$\begin{aligned} \beta^{c_1} &= \alpha\beta^{p_1} + (1-\alpha)\beta^{p_2}, \\ \beta^{c_2} &= \alpha\beta^{p_2} + (1-\alpha)\beta^{p_1}, \end{aligned} \tag{7}$$

where β^{c_1} and β^{c_2} are the children chromosomes, β^{p_1} and β^{p_2} are the parents chromosomes and α is a random weight in interval $[0, 1]$ (Dumitrescu et al., 2000; Eiben and Smith, 2003).

Four hundred independent runs of the GA were performed for each configuration and the population's best historic fitness value was registered after each run. At the end of these tests the results could be summarized, as shown in Table 2. In this table are presented the mean values, $\bar{x}$, standard deviations, σ, minimum and maximum values for each configuration considered. As can be seen, the GA with Heuristic crossover had a better performance in terms of its mean value and presented the lowest standard deviation, despite all variants presented close minimum values.

Crossover algorithm	$\bar{x}$	σ	Minimum	Maximum
Heuristic	0.8722	0.1600	0.6409	1.4402
Arithmetic	1.2923	0.2357	0.7000	2.9106
Single point	1.1725	0.3327	0.6682	2.4887
Two point	1.0293	0.2349	0.6552	1.7758

Table 2: Test's results summary for GA curve fitting with real coded chromosomes.

The same approach was conducted to test the performance of the binary coded GA, but, in

this case, the crossover functions were: Single point, Two point and Scattered; and each gen was represented with a binary number with 28 bits. The Scattered crossover operator generates a random auxiliary binary vector; It selects the genes where the auxiliary vector is 1 form the first parent, and where it is 0 from the second parent in order to create a new child. The results are summarized in Table 3. As can be seen, the GA with Scattered crossover had a better performance in terms of its mean value. It is interesting that all variants presented the same minimum and maximum values.

Crossover algorithm	$\bar{x}$	σ	Minimum	Maximum
Single point	0.7954	0.0286	0.7826	0.8926
Two point	0.7913	0.0248	0.7826	0.8926
Scattered	0.7899	0.0232	0.7826	0.8926

Table 3: GA curve fitting with binary coded chromosomes test results summary.

3.2 Particle Swarm Optimization curve fit

Some recent researchers successfully applied PSO algorithms to curve fit problems as Adi and Shamsuddin (2009), Islam et al. (2009) and Gálvez and Iglesias (2011). This algorithm was firstly proposed by Kennedy and Eberhart (1995). Its working process mimics a flock of birds collectively foraging for food. In this manner, a PSO algorithm places a user specified number of simple entities, the particles, over the problem's search space and the objective function is evaluated at each particle position. The particles, then, determine theirs movement through the search space. For that to happen, the knowledge of its own current and best historic locations and those of one or more particles are combined with random perturbations. Each iteration begins after all particles take their new place and new movements' directions are established. This process is repeated until the particles converge to the optimum point or some stop criterion is met. A comprehensive description of this process can be found in Poli et al. (2007).

The $i-th$ particle's velocity and position are changed according to the following equations:

$$\mathbf{v}_i = w_1 \mathbf{v}_i + \mathbf{U}\left(0, \phi_1\right) \otimes \left(\mathbf{p}_i - \mathbf{x}_i\right) + \mathbf{U}\left(0, \phi_2\right) \otimes \left(\mathbf{p}_g - \mathbf{x}_i\right), \tag{8}$$

$$\mathbf{x}_i = \mathbf{x}_i + \mathbf{v}_i, \tag{9}$$

where $\mathbf{v}_i$ and $\mathbf{x}_i$ are respectively the velocity and position of the $i-th$ particle; $\mathbf{p}_i$ is the $i-th$ particle's historic best position; $\mathbf{p}_g$ is the global historic best position; $\mathbf{U}\left(0, \phi_i\right)$ represents a vector of random numbers uniformly distributed in $[0, \phi_i]$, which is randomly generated at each iteration for each particle; parameters ϕ_1 and ϕ_2 balance the importance of particles' historic best position and importance of global historic best position, respectively. The parameter w could be interpreted as a inertia weight and; $\otimes$ represent a multiplication of two vectors element by element. Particles' velocity components are limited to a maximum value given by the vector $\mathbf{v}_{max}$; If some component j, $\mathbf{v}_i(j)$, on the velocity vector exceeds $\mathbf{v}_{max}(j)$ then the velocity $\mathbf{v}_i(j)$ is set $\mathbf{v}_{max}(j)$.

As can be seen, PSO algorithm has fewer parameters to be adjusted comparing to the Genetic Algorithm. Considering a simple PSO algorithm, these parameters are: Maximum number of iterations, number of particles, w, ϕ_1, ϕ_2 and $\mathbf{v}_{max}$. In the preliminary tests, the maximum number of iterations was adopted 500, the number of particles on each run was 50 and the constants $\psi_1 = 1$ and $\psi_2 = 7.89 \times 10^{-7}$ were adopted in Equation (4).

Two different strategies for inertia weight, w, were tested: Constant weight and Linear decreasing strategy (Xin et al., 2009). The constant weight strategy was presented by Shi and Eberhart (1998). They stated that a large inertia weight facilitates a global search while, on the other hand, a small inertia weight facilitates a local search. This strategy will be evaluated taking $w = 1.0$, as originally proposed by Kennedy and Eberhart (1995), and $w = 0.7$ (Shi and Eberhart, 1998). In the case of Linear decreasing strategy, w will vary from 0.9 to 0.4, (Xin et al., 2009) stated that it provided the best results with some benchmark functions.

Nine configurations were adopted for parameters ϕ_1 and ϕ_2: $\phi_1 = 0.5$ and $\phi_2 = \{3.5, 3.0, 2.5\}$; $\phi_1 = 1.0$ and $\phi_2 = \{3.0, 2.5, 2.0\}$ and; $\phi_1 = 2.0$ and $\phi_2 = \{2.0, 1.5, 1.0\}$. The maximum velocity was adopted equal to the length of the search space for all particles' velocity vector components (Kennedy and Eberhart, 1995; Chen et al., 2005).

Four hundred independent runs of the algorithm were performed, as done with GA strategy, and the best historic value of objective function was registered after each run. With these results the values in Tables 4, 5 and 6 could be obtained. In these tables are presented the mean values, $\bar{x}$, standard deviation, σ, maximum and minimum values of the best particle for each configuration. As can be seen, in general, the lower is the sum $\phi_1 + \phi_2$ the best are the results; Strategies with Linear decreasing weight and $w = 0.7$ found the minimum value 0.0001 in seven of nine different configurations studied. In general, the maximum values found with $w = 0.7$ were smaller than the ones found with the others strategies. The best mean was obtained with $\phi_1 = 2.0$, $\phi_2 = 1.5$ and $w = 0.7$.

ϕ_1	ϕ_2	$\bar{x}$	σ	Minimum	Maximum
	3.5	0.2785	0.3026	0.0084	2.9783
0.5	3.0	0.1267	0.2251	0.0001	2.4088
	2.5	0.0975	0.1907	0.0001	1.2030
	3.0	0.1379	0.2302	0.0019	2.7456
1.0	2.5	0.0933	0.1747	0.0001	1.2499
	2.0	0.1082	0.2093	0.0001	0.9531
	2.0	0.0893	0.1672	0.0001	0.9537
2.0	1.5	0.0833	0.1606	0.0001	0.9511
	1.0	0.0928	0.1386	0.0001	0.9510

Table 4: Tests' results summary for PSO curve fitting with constant weight strategy ($w = 0.7$).

ϕ_1	ϕ_2	$\bar{x}$	σ	Minimum	Maximum
	3.5	3.1851	1.4674	0.2806	8.8549
0.5	3.0	2.7162	1.3379	0.3910	7.1626
	2.5	2.4817	1.3028	0.2746	7.4504
	3.0	2.7069	1.3378	0.2230	7.0968
1.0	2.5	2.3890	1.2761	0.1798	7.8184
	2.0	1.9271	1.0690	0.1272	6.0814
	2.0	2.3959	1.2491	0.1194	6.9791
2.0	1.5	1.8868	1.1035	0.0847	7.1447
	1.0	1.6088	1.0112	0.1623	8.2313

Table 5: Tests' results summary for PSO curve fitting with constant weight strategy ($w = 1.0$).

ϕ_1	ϕ_2	$\bar{x}$	σ	Minimum	Maximum
	3.5	0.4923	0.7008	0.0048	4.8828
0.5	3.0	0.1146	0.1910	0.0001	1.0175
	2.5	0.1334	0.2373	0.0001	1.9385
	3.0	0.1802	0.3148	0.0002	2.7079
1.0	2.5	0.1060	0.2289	0.0001	3.5339
	2.0	0.1056	0.1892	0.0001	1.3653
	2.0	0.1300	0.2613	0.0001	3.6992
2.0	1.5	0.1342	0.1645	0.0001	0.9526
	1.0	0.1423	0.2445	0.0001	2.0380

Table 6: Tests' results summary for PSO curve fitting with Linear decreasing weight strategy.

3.3 Artificial Neural Network curve fit

Artificial Neural Networks (ANN) represent an alternative computational method in which a problem's solution is learned from a set of examples. The idea of neural computing relies over a mathematical model of a biological neuron. This model receives an input signal, **x**; each component x_i is multiplied by a weight, w_i; all of these multiplied components and an additional parameter b, the bias, are summated, producing the υ signal then; the output, y, is generated applying υ in an activation function, $\varphi(\upsilon)$.

A single neuron is not able to learn and generalize complex problems, but it can be extended to an artificial neural network. Associating neurons accordingly to a specific architecture this structure over-comes this limitation. A relatively simple and widely used network architecture type is known as Multi Layer Perceptron (MLP). An MLP is a network organized in layers as illustrated in Figure 1, where each circle is a neuron and the lines represent the linkage between them. The MLP network illustrated has input signal with three parameters, four neurons on hidden layer, two in output layer and a two valued output signal. This layout can be extended indefinitely, adding more hidden layers. Cybenko (1989) had prof that an MLP is an universal approximator.

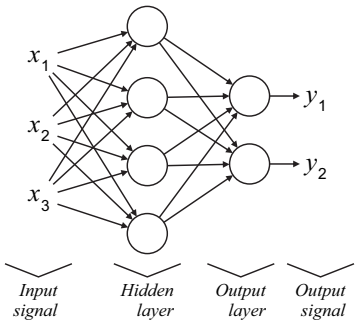


Figure 1: Multi Layer Perceptron network (Adapted from Haikyn (1994)).

This kind of neural network has its weights and biases determined with training algorithms. The general guidelines of these algorithms are: initialize the network, with random values for weights and biases; Present the input signals from the training group, the examples, to the network; Compares the outputs obtained with the desired outputs; Apply a correction over the weights and biases. This process is repeated until some convergence criterion is met.

Preliminary tests were performed in order to investigate the best configuration, in the same

way as done with GA and PSO strategies. Here, the ideal neuron distribution was pursued in order to determine the parameters of Equations (2) and (3). A data set consisting of 400 unique pairs of input signals and theirs corresponding desired output signals was used as training set. This training set was generated taking some of the outputs obtained during the investigation performed with GA and PSO algorithms.

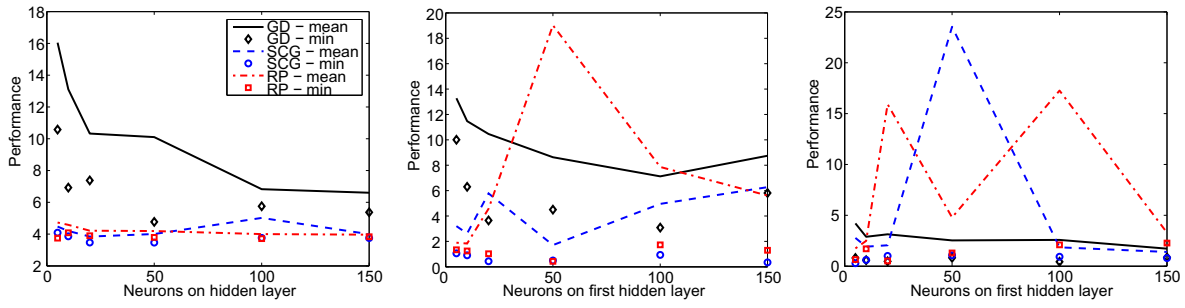
Each input signal has 10 elements, where the first half are the values of function $G'(\omega)$ and the last half are the values of function $\eta(\omega)$, both were evaluated at the same frequencies of characterization data from Table 1. The output signal has seven elements, where each of these elements are the curve parameters G^0, α_j, β_j and δ_j, with $j = 1 \ldots 2$.

There are several MLP training algorithms, three of the most utilized were used to train the networks: Gradient Descent Backpropagation (GD) (Rumelhart and McClelland, 1986); Scaled Conjugate Gradient Backpropagation (SCG) (Moller, 1993) and; Resilient Backpropagation (RP) (Riedmiller and Braun, 1993). The neurons' activation function, $\varphi(v)$, adopted was the hyperbolic tangent function, given by:

$$\varphi(v) = \frac{e^v - e^{-v}}{e^v + e^{-v}}. \tag{10}$$

Before the training sections start, each element of the data set (inputs and outputs) was normalized between -1 and 1, in order to avoid values of $\varphi(v)$ become near its asymptotes.

The trained MLP networks have seven neurons on output layer (one for each parameter) and the number of neurons on hidden layers were determined evaluating the results of 400 independent training sections for each configuration adopted. These results are summarized on Figure 2(a) for a MLP with 1 hidden layer, X-7; in Figure 2(b) for a MLP with 2 hidden layers, where there are 10 neurons on second hidden layer, X-10-7 and; in Figure 2(c) for a MLP with 2 hidden layers, with 20 neurons on second hidden layer, X-20-7. In these figures the lines represent the mean values and the points the minimum values obtained on each configuration considered. Traditionally a MLP performance is measured in terms of the Mean Squared Error (MSE) but, to allow comparisons between the three curve fitting approaches studied in this paper, the values presented at these figures were determined considering Equation (4) and adopting $\psi_1 = 1$ and $\psi_2 = 7.89 \times 10^{-7}$.



(a) X-7 configuration (b) X-10-7 configuration (c) X-20-7 configuration

Figure 2: MLP performance results for different layers configuration.

4 EXPERIMENTAL EVALUATION

In this section, the strategies presented will be applied to a problem with data obtained from literature. Borges (2010) performed a wide program of laboratory studies. In these laboratory

studies, a viscoelastic material was characterized, the double face tape VHB 4955 made by 3M, and sandwich beams were tested with different layers configuration. The material was characterized applying the Direct Method (Faisca et al., 2001) for frequencies between 0 and 800 Hz, as seen in Table 7.

Frequency (Hz)	$G'(\omega)$ (MPa)	$\eta(\omega)$
11.17	0.895 ± 0.025	0.629 ± 0.047
62.58	1.204 ± 0.050	0.801 ± 0.094
171.88	2.468 ± 0.116	0.808 ± 0.249
538.38	2.687 ± 1.154	0.801 ± 0.374
800.50	3.334 ± 1.155	0.808 ± 0.305

Table 7: VHB 4955 characterization data. (Adapted from (Borges, 2010))

The mean values of this data set were presented to the three strategies, where the following parameters were adopted: 1) GA: Real coded chromosome, Heuristic crossover, initial population of 200 individuals, limit of 500 generations, 80% of the new population is created by crossover, 2 Elite individuals and 38 new individuals obtained by mutation; 2) PSO: $w = 0.7$, $\phi_1 = 2.0$, $\phi_2 = 1.5$, 50 particles and 500 iterations maximum; 3) ANN: 5-20-7 neurons configuration and SCG training algorithm. At all strategies were adopted $\psi_1 = 1$ and $\psi_2 = 2.4235 \times 10^{-7}$ on Equation (4). Each strategy was subjected to 400 independent runs and after each one the best result was registered. These results are summarized in Table 8 in terms of its statistical parameters. Analysing this table, one can see that PSO strategy presented the best performance.

Strategy	$\bar{x}$	σ	Minimum	Maximum
GA	0.3458	0.0363	0.3297	0.5273
PSO	0.3297	0.0000	0.3297	0.3299
ANN	9.4635	4.1245	2.7438	13.4552

Table 8: VHB 4955 curve fitting results summary.

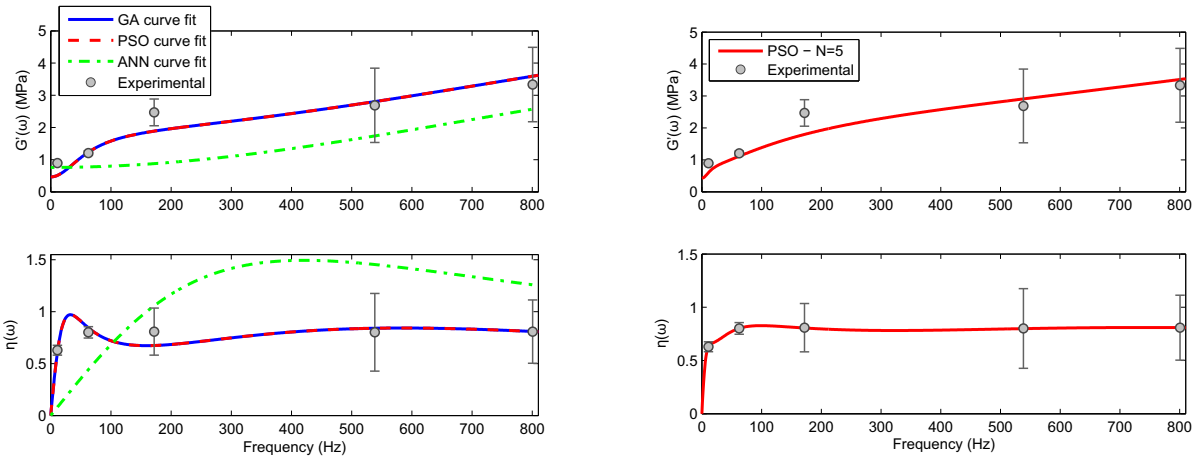
In Figure 3(a) one can see the best curves fitted with each strategy. GA and PSO curves fit are almost coincident and both of them present a good agreement and the curve fitted with the ANN strategy presented the worst agreement.

The parameters obtained for each of these curves are shown in Table 9. Despite the curve fits obtained with GA and PSO strategies are visually coincident and have the same performance, the determined parameters are not close.

Strategy	E_0 ($\times 10^5$)	α_1 ($\times 10^6$)	α_2 ($\times 10^6$)	β_1 ($\times 10^7$)	β_2 ($\times 10^7$)	δ_1 ($\times 10^9$)	δ_2 ($\times 10^9$)
GA	4.6456	1.4942	6.3910	4.1609	0.6532	2.5819	8.8884
PSO	4.6456	6.3896	1.4942	0.3304	6.2715	4.4947	3.8916
ANN	7.5936	4.5511	4.3988	1.1478	0.4539	12.9530	13.7570

Table 9: GHM model fitted parameters for VHB 4955 considering 2 GHM terms.

In order to investigate the influence of the parameters number, an additional curve fit was made using PSO strategy, considering 5 GHM terms (16 parameters total) and the same pa-



(a) Curves obtained considering 2 GHM terms (b) Curves obtained considering 5 GHM terms

Figure 3: Materials curves obtained for VHB 4955 material.

rameters pointed before. The best curve fit was achieved with $E_0 = 4.2619 \times 10^5$, $\alpha_1 = 4.1188 \times 10^6$, $\alpha_2 = 8.0078 \times 10^5$, $\alpha_3 = 2.9600 \times 10^6$, $\alpha_4 = 4.3845 \times 10^5$, $\alpha_5 = 9.6721 \times 10^5$, $\beta_1 = 1.1218 \times 10^6$, $\beta_2 = 5.6458 \times 10^6$, $\beta_3 = 1.4915 \times 10^6$, $\beta_4 = 7.1739 \times 10^7$, $\beta_5 = 1.3015 \times 10^7$, $\delta_1 = 2.1199 \times 10^9$, $\delta_2 = 5.9126 \times 10^8$, $\delta_3 = 3.1934 \times 10^9$, $\delta_4 = 1.0334 \times 10^9$ and $\delta_5 = 3.3495 \times 10^9$. One can see a comparison of the fitted curves obtained with these parameters and the experimental results in Figure 3(b). The adjusted curves are practically coincident with experimental data, specially the $\eta(\omega)$ curve which is visually coincident to the experimental mean values. However, the models matrices size increased and, consequently, the computational effort to solve the Finite Element model increased too.

Borges (2010) tested two specimens of viscoelastic sandwich beams. These beams have two elastic layers, made of aluminium, (a base beam and clamped restraining layer) and one viscoelastic core layer. They have rectangular cross section and 1140 mm length; the beams working as elastic base structure have 16.1mm height; the viscoelastic layers have 2.0mm height; and the elastic constraining layers have 3.17mm height. The first three experimental damping rates are shown in Table 10 in comparison of the ones obtained with a Finite Element model using the parameters found with PSO strategy considering 2 and 5 GHM terms. These models have 9,660 dof (model with 2 GHM terms) and 14,844 dof (model with 5 GHM terms). As can be seen, the results improved 8.15% for first vibration mode, 1.74% for second mode and 17.86% for third mode. These improvements are about 9.25% on average but systems matrices enlarged 153.7%.

Strategy	ξ_1 (%)	ξ_2 (%)	ξ_3 (%)
Experimental	4.98	4.90	4.39
	4.44	4.32	3.28
PSO (N=2)	4.17	4.60	2.24
PSO (N=5)	4.51	4.68	2.64

Table 10: Experimental and numerical damping ratios comparisons.

5 CONCLUSIONS

Three nature inspired strategies were presented to curve fit viscoelastic materials: Genetic Algorithm, Particle Swarm Optimization and Artificial Neural Network; and the influence of some parameters on each of them was investigated. Genetic Algorithm and Particle Swarm Optimization showed a good performance during preliminary tests and on real data example. The Artificial Neural Network had poor performance in comparison on the other ones. An study about the damping rates obtained with GHM Finite Element models, when the number of GHM terms are increased and its impact over the computational cost, was briefly presented.

The best performance of GA curve fit was obtained using real coded chromosomes and Heuristic crossover operator; In case of PSO strategy the best configuration was $w = 0.7$, $\phi_1 = 2.0$ and $\phi_2 = 1.5$. Among the preliminary tests and the real data example the PSO presented a slightly better performance.

The number of terms and the performance of the model are strictly related but, due to the characteristics of GHM model, the size of elemental matrices increases linearly. On models with a great number of viscoelastic elements it could be prohibitive due to the processing time required and the memory space required.

We could state that a good practice is investigating the performance of curve fits with different numbers of GHM terms in a small Finite Element model and then, with the curve fit which presents the best cost-effectiveness relationship, proceed the analysis with a larger model.

ACKNOWLEDGEMENTS Authors would like to thank: CNPq (Conselho Nacional de Desenvolvimento Científico e Tecnológico); UFJF (Federal University of Juiz de Fora); FAPEMIG (Fundação de Amparo à Pesquisa do Estado de Minas Gerais) and CAPES (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior) for financial supports.

REFERENCES

Adi D. and Shamsuddin A. Particle swarm optimization for nurbs curve fitting. *Proc., Sixth International Conference on Computer Graphics, Imaging and Visualization*, 2009.

ASTM. Standard test method for measuring vibration-damping properties of materials. 1993.

Baker J.E. Reducing bias and inefficiency in the selection algorithm. *Proceedings of the Second International Conference on Genetic Algorithms and their Application*, pages 14–21, 1987.

Barbosa F. and Farage M. A finite element model for sandwich viscoelastic beams: Experimental and numerical assessment. *Journal of Sound and Vibration*, 317:91–111, 2008.

Borges F.C. *Análise do comportamento dinâmico de vigas sanduíche com múltiplas camadas*. M.sc. thesis, COPPE/UFRJ, Rio de Janeiro, Brasil, 2010.

Chen J., Quin Z., Liu Y., and Lu J. Particle swarm optimization with local search. *Proc. of the IEEE Int. Conf. Neural Networks and Brain*, pages 481–484, 2005.

Cybenko G. Approximation by superposition of a sigmoidal function. *Math. Control Signals Syst.*, 2:303–314, 1989.

Deep K. and Thakur M. A new crossover operator for real coded genetic algorithms. *Applied Mathematics and Computation*, 188(1):895 – 911, 2007.

DeJong K. *An analysis of the behavior of a class of genetic adaptive systems*. Ph.d. thesis, University of Michigan, 1975.

Dumitrescu D., Lazzerini B., Jain L., and Dumitrescu A. *Evolutionary Computation*. CRC Press, 2000.

Eiben A.E. and Smith J.E. *Introduction to Evolutionary Computing*. Springer-Verlag, 2003.

Faisca R.G., Magluta C., and Roitman N. Experimental characterization of viscoelastic materials as vibration dampers. *Journal of Engineering Mechanics*, 127:959–962, 2001.

Felippe W., Borges F., Barbosa F., Toledo E., Roitman N., and Magluta C. Numerical and experimental evaluation of dynamical behavior of sandwich structures. *Proc., XXXV Jornadas Sul-americanas de Engenharia Estrutural*, 2012.

Friswell M., Dutt J., Adhikari S., and Lee. Time domain analysis of a viscoelastic rotor using internal variable models. *International Journal of Mechanical Sciences*, 52:1319–1324, 2010.

Gálvez A. and Iglesias A. Efficient particle swarm optimization approach for data fitting with free knot b-splines. *Computer-Aided Design*, 43:1683–1692, 2011.

Golla D. and Hughes P. Dynamics of viscoelastic structures - a time-domain, finite element formulation. *Journal of Applied Mechanics*, 52:897–906, 1985.

Gulsen M., Smith A., and Tate D. A genetic algorithm approach to curve fitting. *International Journal of Production Research*, 33:1911–1223, 1995.

Haikyn S. *Neural networks: A comprehensive foundation*. Prentice Hall, 1994.

Hasançebi O. and Erbatur F. Evaluation of crossover techniques in genetic algorithm based optimum structural design. *Computers & Structures*, 78(1–3):435 – 448, 2000.

Herrera F., Lozano M., and Verdegay J. Dynamic and heuristic fuzzy connectives based crossover operators for controlling the diversity and convergence of real-coded genetic algorithms. *International Journal of Intelligent Systems*, 11:1013–1040, 1996.

Hillström L., Mossberg B., and Lundberg B. Identification of complex modulus from measured strains on axially impacted bar using least squares. *Journal of Sound and Vibration*, 230:689–707, 2000.

Holland J. *Adaptation in Natural and Artificial Systems*. Michigan Press, 1975.

Islam M., Moniruzzaman M., Misran N., and Shakib M. Curve fitting based particle swarm optimization for uwb patch antenna. *Jounal of Electromagnetics Waves and Applications*, 23:2421–2432, 2009.

Kennedy J. and Eberhart R. Particle swarm optimization. *Proc., IEEE International conference on neural networks*, 1995.

Lesieutre G. Finite elements for dynamic modeling of uniaxial rods with frequency dependent material properties. *International Journal of Solids and Structures*, 29:1567–1579, 1992.

Lesieutre G. and Bianchini E. Time domain modeling of linear viscoelasticity using anelastic displacement fields. *Journal of Vibration Acoustical*, 117:424–430, 1993.

Lesieutre G. and Govindswamy K. Finite element modeling of frequency-dependent and temperature-dependent dynamic behavior of viscoelastic material in simple shear. *International Journal of Solids and Structures*, 33:419–432, 1996.

Lesieutre G. and Lee U. A finite element for beams having segmented active constrained layers with frequency-dependent viscoelastics. *Smart Materials and Structures*, 5, 1996.

Lesieutre G. and Mingori D. Finite element modeling of frequency-dependent materials properties using augmented thermodynamics fields. *Journal of Guidance Control and Dynamics*, 13:1040–1050, 1990.

Manela M., Thornhill N., and Campbell J. Fitting spline functions to noisy data using a genetic algorithm. *Proc., Fifth International Conference on Genetic Algorithms*, 1993.

Masterson P.A. and Miles R.N. Method for the determination of complex shear modulus of viscoelastic adhesives. *Proc., 2nd Annu. International Symposium on Smart Structures and Materials, Society of Photo-optical Instrumentation Engineers (SPIE)*, 2445:276–281, 1995.

McTavish D. and Hughes P. Modeling of linear viscoelastic space structures. *Journal of Vibration and Acoustics*, 115:103–113, 1993.

Moller M. A scaled conjugate gradient algorithm for fast supervised learning. *Neural Networks*, 6:525–533, 1993.

Morbiducci U., Tura A., and Grigioni M. Genetic algorithms for parameter estimation in mathematical modeling of glucose metabolism. *Computers in Biology and Medicine*, 35:862–874, 2005.

Mousavi S., Nicolas D., and Lundberg B. Identification of complex moduli and poisson's ratio from measured strains on an impacted bar. *Journal of Vibration and Vibration*, 227:971–986, 2004.

Pendharkar P. and Rodger J. An empirical study of impact of crossover operators on the performance of non-binary genetic algorithm based neural approaches for classification. *Computers & Operations Research*, 31(4):481 – 498, 2004.

Poli R., Kennedy J., and Blackwell T. Particle swarm optimization: An overview. *Swarm Intelligence*, 1:1683–1692, 2007.

Riedmiller M. and Braun H. A direct adaptive method for faster backpropagation learning: The rprop algorithm. *IEEE International Conference on Neural Networks*, pages 586–591, 1993.

Roy H., Dutt J., and Datta P. Dynamics of a viscoelastic rotor shaft using augmenting thermodynamic fields—a finite element. *International Journal of Mechanical Sciences*, 50:845–853, 2008.

Rumelhart D. and McClelland J. *Parallel Distributed Processing: Explorations in the Microstructure of Cognitio*. MIT Press, 1986.

Shi Y. and Eberhart R. A modified particle swarm optimizer. *Evolutionary Computation Proceedings*, pages 69–73, 1998.

Soula M., Vinh T., Chevalier Y., Beda T., and Esteoule C. Measurements of isothermal complex moduli of viscoelastic materials over a large range of frequencies. *Journal of Sound and Vibration*, 205:167–184, 1997.

Wang G., Veeramani S., and Wereley N. Analysis of sandwich plates with isotropic face plates and a viscoelastic core. *Journal of Vibration and Acoustics*, 112:305–312, 2000.

Wang Y. and Inman D. Finite element analysis and experimental study on dynamic properties of a composite beam with viscoelastic damping. *Journal of Sound and Vibration*, 332:6177–6191, 2013.

Xin J., Chen G., and Hai Y. A particle swarm optimizer with multistage linearly-decreasing inertia weight. *Proceedings of the Second International Joint Conference on Computational Sciences and Optimization*, pages 505–508, 2009.