JOURNAL OF THEORETICAL
AND APPLIED MECHANICS
42, 3, pp. 445-460, Warsaw 2004

FUZZY-NEURAL AND EVOLUTIONARY COMPUTATION IN IDENTIFICATION OF DEFECTS

Tadeusz Burczyński

Department for Strength of Materials and Computational Mechanics, Silesian University of Technology, Gliwice, Poland
Institute of Computer Modelling, Cracow University of Technology, Kraków, Poland
e-mail: tadeusz.burczynski@polsl.pl

Piotr Orantek
Antoni Skrobol

Department for Strength of Materials and Computational Mechanics, Silesian University of Technology, Gliwice, Poland
e-mail: piotr.orantek@polsl.pl; antoni.skrobol@polsl.pl

It is known that an elastic body contains some internal defects such as voids, cracks, additional masses, etc. This paper is devoted to a method based on computational intelligence for non-destructive defect identification. In the presented paper, an elastic body loaded statically is considered. The body contains an unknown number of internal defects. There are a lot of applications based on non-destructive methods. The Evolutionary Algorithm (EA) with the Boundary Element Method (BEM) is a very effective tool in the identification of internal defects. In this method, the fitness function is calculated for each chromosome in each generation by the BEM. The number of chromosomes in each generation is quite large, and the number of generations is also large, so the time needed to carry out the identification is very long.

Methods based on Artificial Neural Networks (ANN) find the position and shape of internal defects in a very short time. Because ANNs are usually trained using gradient methods, the risk that the solution is in a local optimum is one of disadvantages of such a method. There is also a problem when the ANN has to identify two or more different kinds of defects (cracks, voids and additional masses) in one body.

In the presented method, an EA is connected with the ANN in one system. This operation allows to avoid main disadvantages of these methods and to use their advantages. The evolutionary algorithm is applied to identify the number of defects and their parameters (position and size).

The identification of a defect in the body is performed by minimizing the fitness function which is calculated as a difference between measured and computed displacements in some sensor points on the boundary of the investigated structure. The fitness function is computed using an Artificial Neural Network (ANN).

Key words: fuzzy neural network, evolutionary algorithm, defect, identification, boundary element method

1. Introduction

The main target of this paper is to present a computational intelligence system in identification of defects in the form of cracks and voids in two-dimensional elastic systems. The computational intelligence system is composed of coupled an evolutionary algorithm (EA) with an artificial neural network (ANN) (Rutkowska, 1997). The identification process is realized on the basis of knowledge about displacements in some sensor points on the boundary of the body. There are several approaches to identification problems. One group of methods is based on sensitivity analysis (Bonnet *et al.*, 2002). This approach, from mathematical point of view, is very elegant and strict but sometimes fails because the minimization of identification functions leads to a local minimum.

Another group of methods is based on techniques which try to simulate (or imitate) biological systems. One approach which belongs to this group concerns artificial neural networks. The ANN has been used to identification problems by (Waszczyszyn and Ziemiański, 2001, 2003; Piątkowski and Ziemiański, 2003; Ziemiański and Piątkowski, 2000). In such a method there is a problem with the identification of a large number of different defects, especially when the number of defect is a unknown. The second very common approach is making use of evolutionary algorithms in identification tasks (Burczyński, 2002; Burczyński *et al.*, 2000; Nowakowski, 2000). An EA enables to find multiple defects. It can distinguish different kinds of defects as voids and cracks, and the number of defects can be considered as a design variable. An EA minimizes a fitness function which is formulated as a difference between measured displacements at sensor points x_i, $i = 1, 2, ..., n$ on the boundary of the investigated body and displacements computed for the assumed numerical model with defects

$$\min_{\boldsymbol{ch}} F(\boldsymbol{ch}) \qquad F(\boldsymbol{ch}) = \frac{1}{2}\sum_{i=1}^{n}[\widehat{u}(x_i) - u(x_i)]^2 \tag{1.1}$$

where $\widehat{u}(x_i)$ denotes the measured displacements at the sensor point x_i, $u(x_i)$ are computed displacements for the model in the same point x_i, $\boldsymbol{ch}$ is a vector of defect parameters which plays the role of a chromosome in the EA.

Usually, these computations need to solve a boundary-value problem using the boundary element method (BEM) or the finite element method (FEM) as it is shown in Fig. 1.

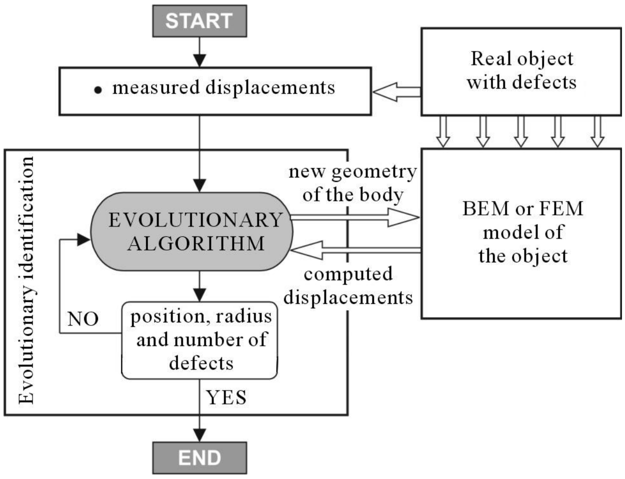

Fig. 1. The evolutionary identification using BEM or FEM to compute the fitness function

This part of the identification process is very time consuming because the fitness function has to be computed for each chromosome in every generation. The second disadvantage of such an approach is that the time needed for solving the identification problem depends on geometry of the model (Nowakowski, 2000). The more complicated shape of the examined object the longer time for computation is needed.

One way to speed up the identification process is to improve the evaluation of the fitness function. It can be done by replacing the BEM or FEM solution to the boundary-value problem by an approximate solution ehich is obtained by using an ANN. As a result of coupling the EA with ANN, a computational intelligence system is obtained (Fig. 2).

It can be said that the artificial neural network is an approximator of a boundary-value problem for different kinds and positions of defects. The EA will find the number, shapes and positions of internal defects based on the results obtained using the ANN.

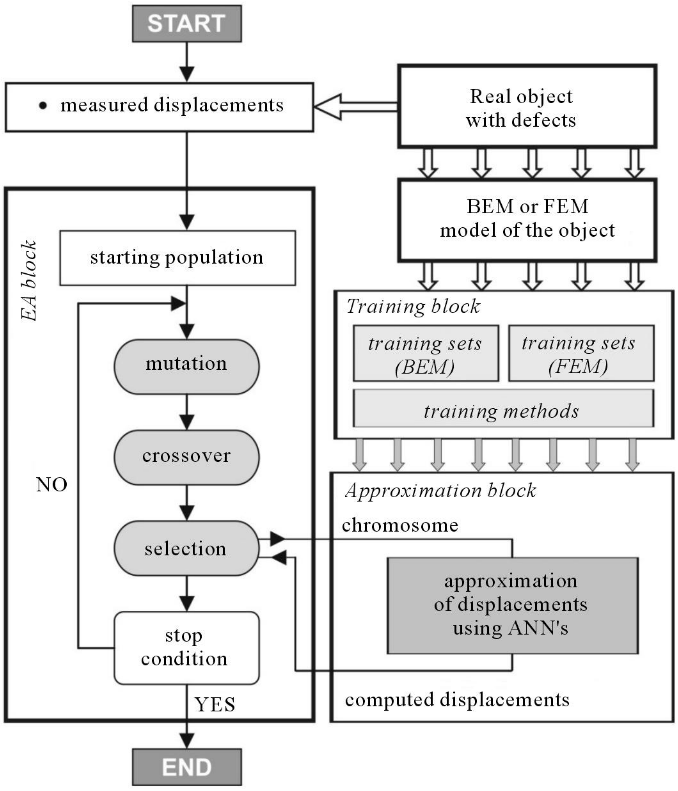

Fig. 2. The computational intelligence system for defect identification

To approximate the boundary-value problem, a fuzzy neural network (FNN) is chosen.

2. Fuzzy modelling

A fuzzy system is a system that uses a collection of fuzzy membership functions and rules, instead of conventional (Boolean) logic, to reason about data. Usually, the form of rules is following (Osowski, 1996)

$$\text{IF} \;\; x_1 = A_1 \;\; \text{AND} \;\; x_2 = A_2 \;\; \text{AND} \;\; x_n = A_n \;\; \text{THEN} \; y = B \tag{2.1}$$

where x_i is the input variable, y is the output variable, A_i is the fuzzy subset of rules premise, B is the fuzzy set of rules conclusion. The rules are collected in one set called the rule base or knowledge base.

A typical fuzzy system consists of four parts (Rutkowska, 1997) as it is shown in Fig. 3.

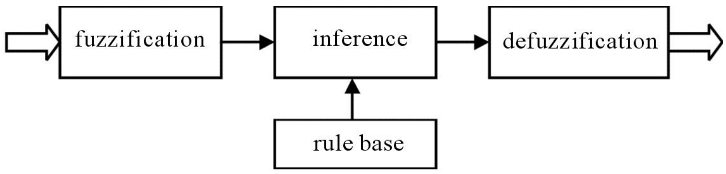

Fig. 3. A scheme of a typical fuzzy system

The inference process proceeds in three steps: fuzzification, inference and defuzzification.

FUZZIFICATION block – the degree of membership of every input variable for each rule premise is determined.

INFERENCE block – the membership degrees are applied to the conclusion part of each rule and the one fuzzy subset for each rule is obtained.

In the presented fuzzy system, the fuzzy subset A is calculated by the following formula

$$\mu_A(\boldsymbol{x}) = \mu_{A1}(x_1)\mu_{A2}(x_2)\dots\mu_{An}(x_n) \tag{2.2}$$

where $\mu_A(\boldsymbol{x})$ is the membership function of the conclusion of the rule A for input vector $\boldsymbol{x}$, $\mu_{Ai}(x_i)$ is the degree of membership of every input variable for A rule premise.

After that all fuzzy subsets are combined together to create one fuzzy set.

DEFUZZIFICATION block – the output fuzzy set is converted to a crisp number. In this paper, the centroid method is considered. In this method the output value (the crisp value) is computed by finding the value of the centre of gravity of the membership function of the output set

$$y = \frac{\sum\limits_{l=1}^{M} c_l \mu_{A^{(l)}}(\boldsymbol{x})}{\sum\limits_{l=1}^{M} \mu_{A^{(l)}}(\boldsymbol{x})} \tag{2.3}$$

where c_l is the centre of the output set for the rule $A^{(l)}$, $\mu_A^{(l)}(\boldsymbol{x})$ is the membership function calculated in the inference step, $l = 1, 2, \dots, M$ is the rule number.

Applying the above described methods an arbitrary continuous function can be represented. Using (2.2) and (2.3) the following formula is obtained

$$f(\boldsymbol{x}) = \frac{\sum\limits_{l=1}^{M} c_l \Big(\prod\limits_{i=1}^{N} \mu_{A_i^{(l)}}(x_i) \Big)}{\sum\limits_{l=1}^{M} \prod\limits_{i=1}^{N} \mu_{A_i^{(l)}}(x_i)} \tag{2.4}$$

where $l = 1, 2, \dots, M$ is the number of rule, $I = 1, 2, \dots, N$ is the number of input, c_l is the centre of the fuzzy output set.

3. Fuzzy neural network

To approximate the fitness value, a fuzzy neural network (FNN) is considered. The FNN should realize a multi-variable function using the sum of single-variable fuzzy functions. These fuzzy functions are characterized by the membership function $\mu(\boldsymbol{x})$. The Gaussian description of the membership function for every input in every rule is assumed

$$\mu_A(\boldsymbol{x}; \boldsymbol{c}, \boldsymbol{\sigma}) = \exp\Big(- \Big[\frac{\boldsymbol{x} - \boldsymbol{c}}{\boldsymbol{\sigma}} \Big]^2 \Big) \tag{3.1}$$

In this case, formula (2.4) can be presented as follows

$$f(\boldsymbol{x}) = \frac{\sum\limits_{l=1}^{M} W_l \prod\limits_{i=1}^{N} \exp\Big(- \Big[\frac{x_i - c_i^{(l)}}{\sigma_i^{(l)}} \Big]^2 \Big)}{\sum\limits_{l=1}^{M} \prod\limits_{i=1}^{N} \exp\Big(- \Big[\frac{x_i - c_i^{(l)}}{\sigma_i^{(l)}} \Big]^2 \Big)} \tag{3.2}$$

where W_l corresponds to the centre c_l in equation (2.3). In this formula, $c_i^{(l)}$ and $\sigma_i^{(l)}$ are centres and widths of part "IF" in each rule, and W_l is the centre of part "THEN" in each rule.

This function can be described by making use of a multi-layer structure called the fuzzy neural network (Fig. 4).

During the training process the parameters W_l, $c_i^{(l)}$ and $\sigma_i^{(l)}$ should be found. In a gradient optimisation the learning process depends on the minimization of the square error which can be presented as follows

$$E = \frac{1}{2} [f(\boldsymbol{x}) - d]^2 \tag{3.3}$$

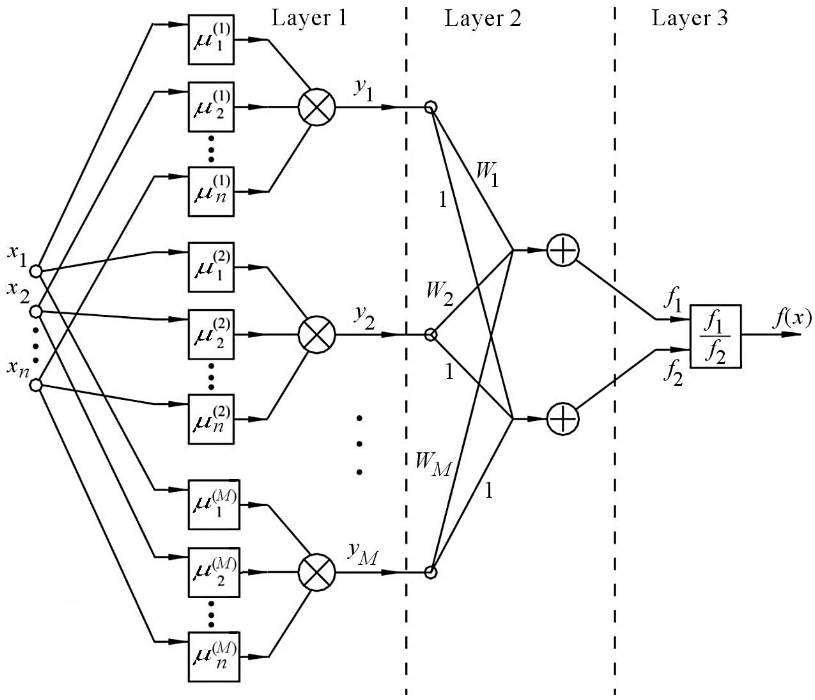

Fig. 4. The scheme of the fuzzy neural network with one output

where $\boldsymbol{x}$ is the input vector, $f(\boldsymbol{x})$ is the value approximated by the fuzzy neural network and d is the desirable answer of the FNN for the input vector $\boldsymbol{x}$.

When the training process is carried out by making use of the gradient method, the knowledge about the gradient vector ∇E is very important. When the function $f(\boldsymbol{x})$ is in the form shown in formula (3.2) and the error is defined as it is presented in (3.3), the gradient vector ∇E has three components

$$\begin{aligned}
\frac{\partial E}{\partial W_l} &= [f(\boldsymbol{x}) - d]\frac{y_l}{f_2} \\
\frac{\partial E}{\partial c_i^{(l)}} &= 2\frac{f(\boldsymbol{x}) - d}{f_2} y_l [W_l - f(\boldsymbol{x})] \frac{x_i - c_i^{(l)}}{(\sigma_i^{(l)})^2} \\
\frac{\partial E}{\partial \sigma_i^{(l)}} &= 2\frac{f(\boldsymbol{x}) - d}{f_2} y_l [W_l - f(\boldsymbol{x})] \frac{(x_i - c_i^{(l)})^2}{(\sigma_i^{(l)})^3}
\end{aligned} \tag{3.4}$$

for every input $i = 1, 2, \ldots, N$ and each rule $l = 1, 2, \ldots, M$.

The change of parameters is proceeding according to the method shown below

$$p(s+1) = p(s) - \eta \frac{\partial E}{\partial p} + \alpha \Delta p(s-1) \tag{3.5}$$

where p is the parameter put to optimisation, s – number of the iteration step, η – learning rate value, α – momentum rate, $\Delta p(s-1)$ – parameter increment in the $(s-1)$ step.

4. Formulation of an identification problem

A two dimensional elastic body with n internal defects in the form of circular holes is considered. The EA should identify the number of defects and their parameters based on information about displacements in m sensor points on the boundary of the body. The unknown parameters of a defect are coordinates of the hole centre (X_i, Y_i) and its size R_i, $i = 1, 2, \ldots, n$.

The defects are specified by a chromosome

$$\boldsymbol{ch} = [X_1, Y_1, R_1, X_2, Y_2, R_2, \ldots, X_i, Y_i, R_i, \ldots, X_n, Y_n, R_n] \tag{4.1}$$

where X_i, Y_i and R_i, $i = 1, 2, \ldots, n$, play the role of genes, n is the number of a defect. The evolutionary algorithm sends the chromosome with suggestion values of positions and radii of defects to the approximation block (Fig. 2). In the case when $R_i < R_{min}$, the program assumes that the genes X_i, Y_i, R_i are inactive genes and

$$R_i = 0 \qquad \forall R_i < R_{\min} \tag{4.2}$$

Condition (4.2) controls the number of defects. The number of input values, which are sent from the EA to the fuzzy neural networks, depends on the number of active genes. Thus, in the approximation block there are several fuzzy neural networks with different numbers of input neurons (Fig. 5). Every FNN is responsible for approximation of displacements on the boundary of the model with a different number of internal defects.

Genes with information about the position and shape of defects are sent to the inputs of FNNs. The number of active genes defines indirectly the number of internal defects. Approximated displacements in several sensor points on the boundary of the model are obtained on the outputs of FNNs. They are sent back to EA where the fitness function of each chromosome is computed.

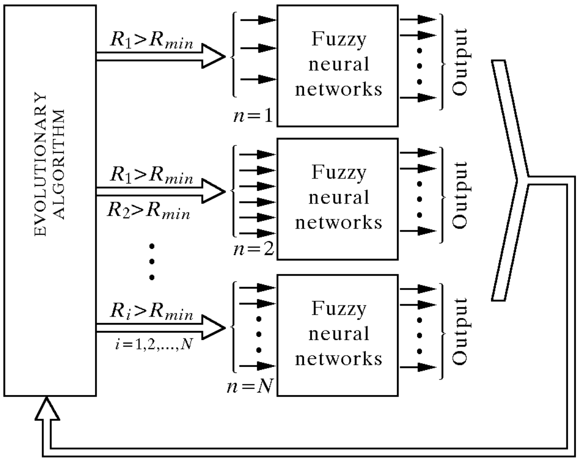

Fig. 5. Approximation of displacements for one, two or more internal defects

5. Numerical tests

A two-dimensional elastic rectangle in the plane stress under statical load is considered. The body contains one or two defects in the form of a circular hole. The considered structure with one defect is presented in Fig. 6a, the structure with two defects is shown in Fig. 6b. One should find the number, position and size of internal defects. To solve the problem, an evolutionary algorithm coupled with a fuzzy neural network is applied. The fuzzy neural network is chosen because of its good approximation abilities (Osowski, 1996) and the short time needed for learning (see Table 3). The learning time of such a network is much shorter than the time needed to learn BPNN (Burczyński *et al.*, 2003).

In both cases the defects are described by a chromosome with six genes

$$\boldsymbol{ch} = [X_1, Y_1, R_1, X_2, Y_2, R_2] \tag{5.1}$$

The evolutionary algorithm sends the chromosome with suggestion of positions and values of radii of two defects: X_i, Y_i, R_i, $i = 1, 2$ to the approximation block (Fig. 2). In the event when one of the values of radii is less than R_{min} this value equals zero. In such a case the three input values are sent to fuzzy neural networks with three inputs. When both R_1 and R_2 are bigger than R_{min} then the input vector with six elements is sent to other fuzzy neural networks with six inputs.

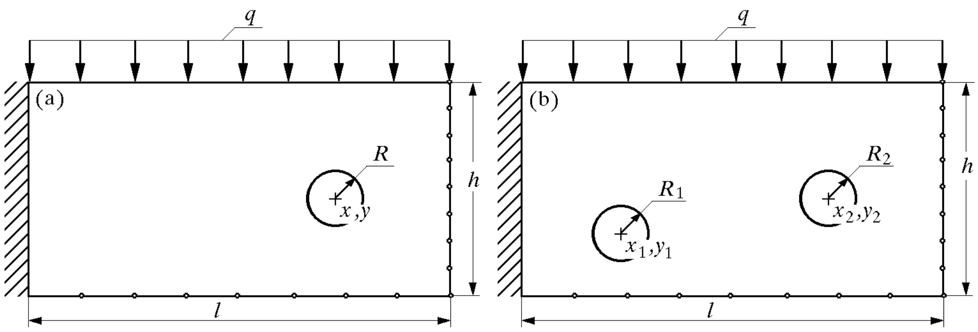

Fig. 6. The structure with (a) one internal defect (b) two defects

Input neurons get three or six values – the radius of the hole, X and Y coordinates of the hole centre for one or two defects. The number of sensor points is $m = 20$, so the output values are twenty displacements in the OX direction and twenty displacements in the OY direction. These are the displacements in sensor points on the boundary of the model with hole parameters proposed by the chromosome. Because the fuzzy neural network with only one input is used, in order to approximate displacements in two directions in 20 sensor points on the boundary of the body a set of forty fuzzy neural networks with 3 inputs and one output and the set of 40 FNN's with 6 inputs and one output has to be built. Each FNN is responsible for displacement approximation in one direction in only one sensor point.

The artificial neural network was learned and tested on values obtained by making use of the boundary element method for the 2D problem of elastostatics (Burczyński, 1995). The set of fuzzy-neural networks with three input neurons (FNN-3) was trained by 2374 vector pairs, and for 231 pairs was verified. The set of FNNs with six input neurons (FNN-6) was trained using 5032 vector pairs, and for 184 was verified. The procedure of training by the back propagation method with momentum was applied. The error was computed in the following way

$$Er = \frac{1}{2T} \sum_{t=1}^{T} \sum_{u=1}^{U} (f^{(u)}_{(\boldsymbol{x}_t)} - d^{(u)}_t) \tag{5.2}$$

where T is the number of training pairs, U is the number of outputs, $f(\boldsymbol{x})$ is the value given by the FNN and d is the desirable answer for the input vector $\boldsymbol{x}_t$.

The error of training set (Er_l) and testing set (Er_t) for different fuzzy neural network sets with different numbers of rules are given in Table 1 for the set of fuzzy neural networks with three inputs (FNN-3), and in Table 2 for the set of fuzzy neural networks with six inputs (FNN-6).

Table 1. Error values for FNN-3 with different number of rules in the experimental training

FNN-3					
No. of rules	**3**	**5**	**7**	**10**	**13**
Er_l	0.00052	0.00052	0.00052	0.00052	0.00052
Er_t	0.00619	0.00586	0.00568	0.00582	0.00555
No. of iterations	20	20	20	20	10
No. of rules	**15**	**17**	**20**	**25**	**40**
Er_l	0.00052	0.00053	0.00053	0.00052	0.00053
Er_t	0.00557	0.00559	0.00549	0.00526	0.00513
No. of iterations	10	10	10	10	10

Table 2. Error values for FNN-6 with different number of rules in the experimental training

FNN-6					
No. of rules	**3**	**5**	**7**	**10**	**15**
Er_l	0.00030	0.00030	0.00030	0.00030	0.00030
Er_t	0.00640	0.00582	0.00643	0.00685	0.00619
No. of iterations	10	10	10	10	10
No. of rules	**17**	**20**	**25**	**35**	**40**
Er_l	0.00030	0.00030	0.00030	0.00031	0.00030
Er_t	0.00599	0.00631	0.00602	0.00580	0.00599
No. of iterations	10	10	10	10	10

The starting parameters W, c, σ were random values. Based on Table 1 and Table 2, two fuzzy neural network architectures and two sets of the starting parameters W, c, σ were chosen for further training. Finally, the following fuzzy neural networks were obtained (Table 3).

Table 3. Architecture and training parameters

	FNN-3	FNN-6
Architecture		
No. of inputs	3	6
No. of outputs	40	40
No. of rules per input	40	35
Training parameters		
No. of iterations	9	14
Er_l	0.00052	0.00030
Er_t	0.00504	0.00571
Learning rate value (η)	0.2	0.2
Momentum rate (α)	0.9	0.9
Time of training [s]	35	29
The number of learning pairs	2374	5032
The number of testing pairs	231	184

The following evolutionary parameters were applied (Table 4).

Table 4. The parameters of the evolutionary algorithm

Number of chromosomes	300
Number of iterations	100
Number of design parameters	6
Probability of uniform mutation	0.25
Probability of arithmetic crossover	0.25
Probability of cloning	0.05
Selection coefficient	0.75

In this paper, only two examples are presented. The geometrical and material parameters of the body with one and two defects are described in Table 5.

The actual and found defects using the EA with BEM are shown in Fig. 7a (one defect) and in Fig. 7b (two defects). The deffects determined by the computational intelligence system are presented in Fig. 7c (one defect) and in Fig. 7d (two defects). It is seen that in both cases the evolutionary algorithms have found actual numbers of defects.

The evolutionary algorithm, with fitness function values approximated by the FNN, found the best solution in 2 min. 20 sec. in the case of the body with two internal defects and in 2 min. 25 sec. for the body with one defect. In the

Table 5. Geometrical and material parameters of the examined objects

Geometrical and material parameters	The structure with one defect	The structure with two defects
l [m]	4.0	4.0
h [m]	2.0	2.0
q [N/m]	3750	3750
E [MPa]	$2 \cdot 10^5$	$2 \cdot 10^5$
ν	0.3	0.3
X_1 [m]	0.92	0.7
Y_1 [m]	1.54	1.1
R_1 [m]	0.16	0.07
X_2 [m]	–	1.55
Y_2 [m]	–	0.6
R_2 [m]	–	0.07
R_{min} [m]	0.0314	

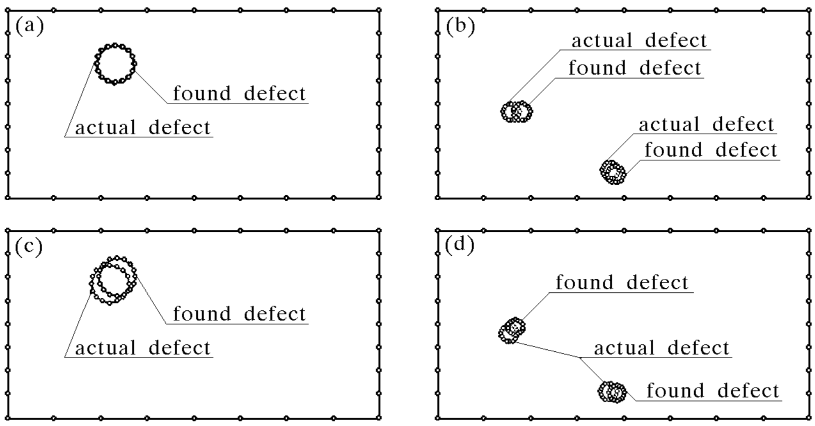

Fig. 7. Actual and found defects using EA with BEM: (a) one defect (b) two defects; using EA with FNN: (c) one defect, (d) two defects

case of the evolutionary algorithm with BEM the CPU time was 22 min. 25 sec. and 11 min. 40 sec., respectively. It can be said then that the evolutionary algorithm with the fitness function approximated by using the fuzzy neural network is much faster than an evolutionary algorithm with the boundary element method (Fig. 8).

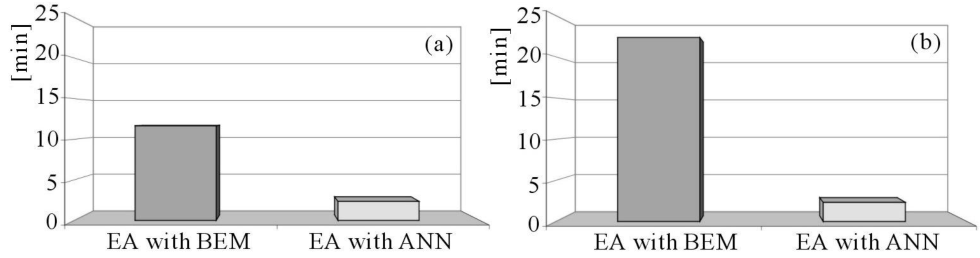

Fig. 8. CPU time using EA with BEM and EA with FNN for a body with (a) one defect (b) two defects

6. Conclusions

The presented tests confirm that the evolutionary algorithm with the artificial neural network identifies the number, positions and radii of circular holes in a 2D body under static load.

This approach is less accurate but much faster than the evolutionary algorithm with the boundary element method. In the case of identification of two internal defects, the computing time using the computational intelligence system is about 90% shorter than the computing time consumed by the EA with BEM. The more complicated geometry of the examined body the longer time for the identification through EA with BEM is needed. In the proposed approach the time of computations does not depend on geometry of the body.

The advantage of employing the FNN instead of BPNN is the much shorter time needed for the FNN training (Burczyński *et al.*, 2003). When fuzzy neural networks are applied, there is also a possibility of containing some knowledge about a problem before the training process (Jang *et al.*, 1997).

The time of computation with the EA and FNN used does not take into account the time needed to learn the FNN and the time needed to prepare the learning and testing sets. The computational intelligence system is worth using when the defect identification has to be done in many structures with the same shape. In such a case the time needed to prepare the learning and testing sets and to train the FNN is not significant.

References

1. Bonnet M., Burczyński T., Nowakowski M., 2002, Sensitivity analysis for shape perturbation of cavity or internal crack using BIE and adjoint variable approach, *International Journal of Solids and Structures*, **39**, 2365-2385
2. Burczyński T., 1995, *The Boundary Element Method in Mechanics*, WNT Warszawa
3. Burczyński T. (edit.), 2002, *Computational Sensitivity Analysis and Evolutionary Optimization of Systems with Geometrical Singularities*, ZN KWMiMKM, Gliwice
4. Burczyński T., Beluch W., Długosz A., Orantek P., Nowakowski M., 2000, Evolutionary methods in inverse problems of engineering mechanics, In: *Inverse Problems in Engineering Mechanics II* (eds. M. Tanaka and G.S. Dulikravich), Elsevier, 553-562
5. Burczyński T., Orantek P., Skrobol A., 2003, Application of computational intelligence system for defect identification, *Proc. ECCOMAS Symposium on Artificial Intelligence AI-METH*, Gliwice
6. Jang J.R., Sun Ch., Mizutani E., 1997, *Neuro-Fuzzy and Soft Computing. A Computational Approach to Learning and Machine Intelligence*, Prentice-Hall, Upper Saddle River
7. Nowakowski M., 2000, Analiza wrażliwości i identyfikacja kształtu brzegów wewnętrznych drgających układów mechanicznych przy zastosowaniu metody elementów brzegowych, Rozprawa doktorska, Politechnika Śląska, Gliwice
8. Osowski S., 1996, *Sieci neuronowe w ujęciu algorytmicznym*, WNT, Warszawa
9. Piątkowski G., Ziemiański L., 2003, Neural network identification of a circular hole in the rectangular plate, In: *Neural Networks and Soft Computing* (eds L. Rutkowski, J. Kacprzyk), Heidelberg, Physica-Verlag Springer, 778-783P
10. Rutkowska D., 1997, *Computational Intelligent Systems*, Akademicka Oficyna Wydawnicza PLJ, Warszawa (in Polish)
11. Waszczyszyn Z., Ziemiański L., 2001, Neural networks in mechanics of structures and materials – new results and prospects of applications, In: *Computer and Structures*, **79**, 2261-2276
12. Waszczyszyn Z., Ziemiański L., 2003, Neural networks in the identification analysis of structural mechanics problems, *CISM Advanced School on Parameter Identification of Materials and Structures*, Udine
13. Ziemiański L., Piątkowski G., 2000, Use of neural networks for damage detection in structural elements using wave propagation, In: Topping B.H.V. (Edit.), *Computational Engineering using Metaphors from Nature*, Edinburgh: Civil-Comp Press, 25-30

Neuronowo-rozmyte oraz ewolucyjne obliczenia w identyfikacji defektów

Streszczenie

Obiekty techniczne jako układy mechaniczne zawierają różne defekty wewnętrzne takie jak pustki, pęknięcia itp. Artykuł jest poświęcony nieniszczącym metodom identyfikacji defektów opartym na inteligencji obliczeniowej. Rozważane jest ciało sprężyste znajdujące się pod wpływem obciążenia statycznego zawierające nieznaną liczbę defektów wewnętrznych. Istnieje wiele nieniszczących metod identyfikacji defektów wewnętrznych. Jedną z nich jest metoda oparta na Algorytmach Ewolucyjnych (AE) połączonych z Metodą Elementów Brzegowych (MEB). W tej metodzie dla każdego chromosomu w każdym pokoleniu obliczana jest za pomocą MEB funkcja przystosowania. Ponieważ liczba chromosomów w epoce oraz liczba epok jest dosyć duża, zatem czas potrzebny do przeprowadzenia identyfikacji jest znaczący.

Metody bazujące na Sztucznych Sieciach Neuronowych (SSN) identyfikują położenie oraz kształt defektów wewnętrznych w bardzo krótkim czasie. SSN są zazwyczaj uczone z wykorzystaniem metod gradientowych. Istnieje zatem spore ryzyko, że uzyskane rozwiązanie utknęło w minimum lokalnym. Wykorzystując SSN napotykamy na spore trudności również w przypadku identyfikacji dwóch lub więcej różnych rodzajów defektów (pęknięć, pustek itp.), które występują jednocześnie w identyfikowanym układzie,

W metodzie opisywanej w niniejszym artykule połączono AE oraz SSN w jeden system. Operacja ta pozwoli ustrzec się przed głównymi wadami i uwypuklić zalety obydwu metod. AE identyfikuje liczbę, położenie oraz wymiary defektów. Identyfikacja następuje przez minimalizację funkcji przystosowania, która jest mierzona jako różnica pomiędzy zmierzonymi i obliczonymi przemieszczeniami na brzegu modelu obiektu w punktach kontrolnych. Funkcja przystosowania jest obliczana z wykorzystaniem SSN.

Manuscript received March 3, 2004; accepted for print April 26, 2004