Computer Assisted Mechanics and Engineering Sciences, **13**: 351–364, 2006.

An intelligent computing technique in identification problems

Piotr Orantek
Department for Strength of Materials and Computational Mechanics
Silesian University of Technology
Konarskiego 18a, 44-100 Gliwice, Poland

(Received March 31, 2005)

The paper is devoted to the application of the evolutionary algorithms, gradient methods and artificial neural networks to identification problems in mechanical structures. The special intelligent computing technique (ICT) of global optimization is proposed. The ICT is based on the two-stage strategy. In the first stage the evolutionary algorithm is used as the global optimization method. In the second stage the special local method which combines the gradient method and the artificial neural network is applied. The presented technique has many advantages: (i) it can be applied to problems in which the sensitivity is very hard to compute, (ii) it allows shortening the computing time. The key problem of the presented approach is the application of the artificial neural network to compute the sensitivity analysis. Several numerical tests and examples are presented.

1. Introduction

The identification problems belong to inverse problems and concern the determination of mechanical systems by finding same material, shape and topology parameters and boundary conditions from the knowledge of the responses to given excitations. Such problems are mathematically illposed. At present, the main challenges in solving such problems consist in elaborating new computational methods, experimental techniques, regularization techniques and the formulation of new objective funtionals [2]. An identification problem can be formulated as the minimization of some objective functionals (fitness functions) which depend on measured and computed state fields such as displacements, strains, eigenfrequencies or temperature. In order to obtain the unique solution of the identification problem one should find the global minimum of the objective functional. One of the global methods of optimization is the evolutionary algorithm [1, 13]. Evolutionary algorithms have been applied successfully to solve many identification problems [4–6]. The main disadvantage of this approach is very long computing time. In order to speed up this approach the hybrid evolutionary algorithm was proposed [7, 8] in which a special kind of gradient mutation was applied. It was based on the sensitivity of the objective functional. The gradient of the functional was computed by means of sensitivity analysis [11] or the finite difference method. The frequency of the gradient mutation was controlled by the artificial neural network.

In this paper a new intelligent computing technique based on the two-stage strategy is proposed. In the first stage the evolutionary algorithm is applied. The second stage is based on the gradient method in which the sensitivity analysis of the objective functional is performed by neuro-computing.

The problem of applications of ANN to the sensitivity analysis was considered in [12, 15]. In the paper [12] the sensitivity analysis as the method of the analysis of the effects of the ANN is described. The paper [15] is devoted to the sensitivity analysis, which defines a robust model of the neural network.

In the present paper the ANN is used as the computation tool of the sensitivity analysis. The method of local optimization, which uses the ANN to determinate the sensitivity analysis is shown.

This approach is more general than the methods described earlier, and is applied to an identification problem of mechanical structures.

The paper consists of 9 sections. In Sec. 2 the evolutionary algorithms are described. In Sec. 3 the idea of neural computing in the sensitivity analysis and the examination of this approach is presented. Section 4 is devoted to the local optimization method, which uses the neuro-computation of the sensitivity analysis. In Sec. 5 the intelligent computing technique is presented. Section 6 is devoted to the examination of the ICT for the minimization of a simple benchmark function. In Secs. 7 and 8 the applications of the ICT for the identification problems of 2-D and 3-D elastic structures were included respectively. In Sec. 9 general conclusions are presented.

2. Evolutionary algorithms

The evolutionary algorithms (Fig. 1) are one of the methods of artificial intelligent, and can be considered as a method of global optimization.

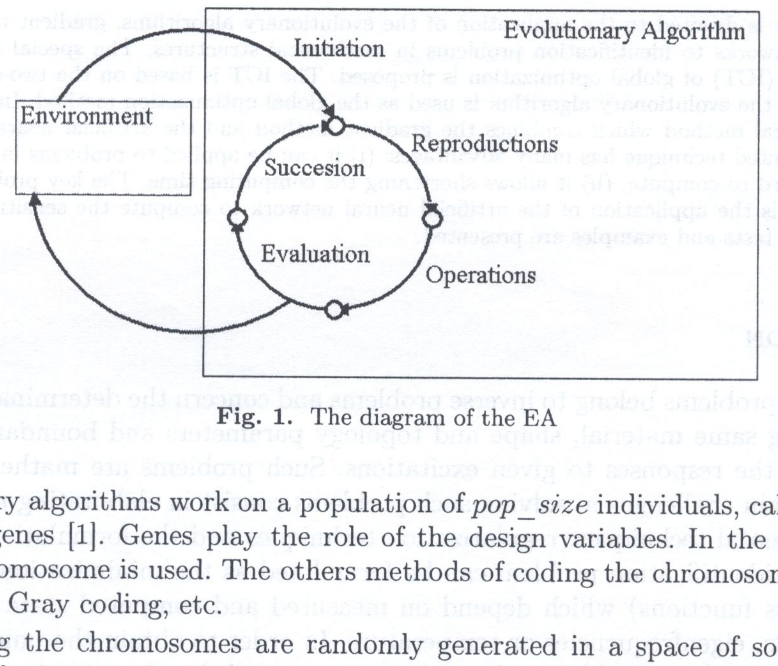

Fig. 1. The diagram of the EA

The evolutionary algorithms work on a population of *pop_size* individuals, called chromosomes, which consist of genes [1]. Genes play the role of the design variables. In the paper the case of the real-coded chromosome is used. The others methods of coding the chromosome are known: the binary coding, the Gray coding, etc.

At the begining the chromosomes are randomly generated in a space of solution (initation). Next, the EA works iteratively. In each iteration (generation) the chromosomes are modified by means of evolutionary operators: Gaussian mutation and arithmetical crossover. Many others kinds of mutation: uniform, non-uniform, boundary, etc. and kinds of crossover: simple, heuristic etc. are known. The mutation changes some values of randomly selected genes of the chromosome. The crossover creates children on the basis of two randomly selected parents. The frequency of working of the operators depends on the parameters, which describe the probability of the mutation *pro_mut* and the crossover *pro_cro*. The next step is the evaluation of the objective function for each chromosome (evaluation). The objective function plays the role of the environment to distinguish between good and bad solutions. The last step in each generation is the selection (succesion and reproduction). In this work the tournament selection was used. Many others types of selection: roulette-whell, rang, etc. are known. All steps are repeated until the stop condition is fulfilled. In many cases the stop condition is imposed to the number of generations.

In many works the best chromosome is considered as a final solution of the evolutionary algorithm.

In the ICT a set of chromosomes is taken into account as the result of the first stage. There is a great probability that this set is close to the global optimum. Therefore, in the second part of the ICT the global optimum is found with great accuracy.

3. Sensitivity analysis by neural computing

The approximation problem is one of the well known applications of the ANN [10]. Consider the ANN shown in Fig. 2. The neurons $i_j = 1..I_j$ create j-th layer, where $j = 0..J$ is the number of the layers. Additionally, consider s_{ij} as the sum of signals in the i-th neuron in j-th layer, e_{ij} are the output values of i-th neurons in j-th layer. The output value of the neuron with the sigmoid activation function is expressed as:

$$e_{ik} = f(s_{ik}) = \frac{1}{1+e^{-s_{ik}}} \tag{1}$$

where:

$$\begin{aligned} s_{ik} &= e_{i-11}w_{i-11ik} + e_{i-12}w_{i-12ik} + \ldots + e_{i-1I_{i-1}}w_{i-1I_{i-1}ik} + ww_{ik} \\ &= \sum_{n=1}^{I_{i-1}} e_{i-1n}w_{i-1nik} + ww_{ik} \end{aligned} \tag{2}$$

e – output values from previous layer, w – weight.

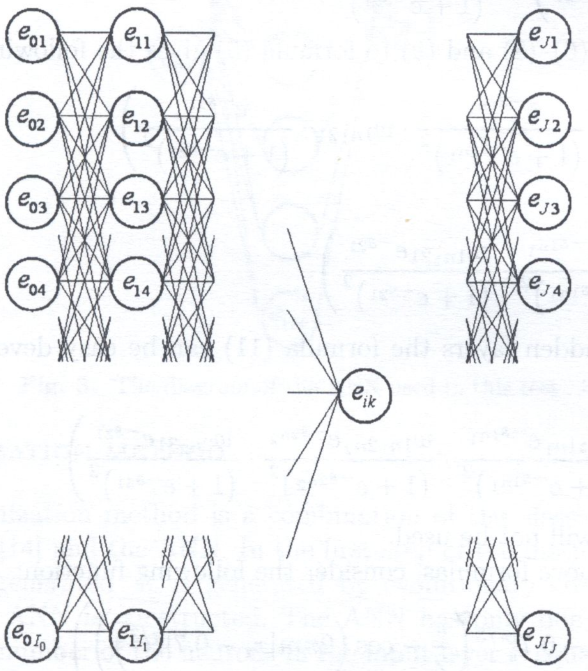

Fig. 2. The diagram of the ANN

Computing the sensitivity of the sigmoid active function is very easy. The sensitivity of the output signal e_{J1} with respect to some input value e_{0z} can be expressed as follows:

$$\frac{de_{J1}}{de_{0z}} = \sum_{n_1=1}^{I_1} \frac{ds_{1n_1}}{de_{0z}} \cdot \frac{de_{1n_1}}{ds_{1n_1}} \cdot \left(\sum_{n_2=1}^{I_2} \frac{ds_{2n_2}}{de_{1n_2}} \cdot \frac{de_{2n_2}}{ds_{2n_2}} \right) \left(\sum_{\ldots}^{\ldots} \ldots \right) \tag{3}$$

The fitness function close to the optimum is approximated by a parabolic function, therefore only one hidden layer in the ANN is sufficient.

The formula (3) can be reduced to the equation:

$$\frac{de_{21}}{de_{0z}} = \sum_{n_1=1}^{I_1} \frac{ds_{1n_1}}{de_{0z}} \cdot \frac{de_{1n_1}}{ds_{1n_1}} \cdot \left(\sum_{n_2=1}^{I_2} \frac{ds_{2n_2}}{de_{1n_2}} \cdot \frac{de_{2n_2}}{ds_{2n_2}} \right) \tag{4}$$

if the $n_1 \neq n_2$ the second component is equal to 0. It allows disregarding the second sum and simplification previous formula to:

$$\frac{\mathrm{d}e_{21}}{\mathrm{d}e_{0z}} = \sum_{n_1=1}^{I_1} \left(\frac{\mathrm{d}s_{1n_1}}{\mathrm{d}e_{0z}} \cdot \frac{\mathrm{d}e_{1n_1}}{\mathrm{d}s_{1n_1}} \cdot \frac{\mathrm{d}s_{2n_1}}{\mathrm{d}e_{1n_1}} \cdot \frac{\mathrm{d}e_{2n_1}}{\mathrm{d}s_{2n_1}} \right), \tag{5}$$

where:

$$\frac{\mathrm{d}s_{1n_1}}{\mathrm{d}e_{0z}} = \frac{\mathrm{d}}{\mathrm{d}e_{0z}} \left(\sum_{n_0=1}^{I_0} \left(e_{0n_0} \cdot w_{0n_0 1n_1} + ww_{1n_1} \right) \right) = w_{0z1n_1}, \tag{6}$$

$$\frac{\mathrm{d}e_{1n_1}}{\mathrm{d}s_{1n_1}} = \frac{\mathrm{d}}{\mathrm{d}s_{1n_1}} \left(\frac{1}{1+e^{-s_{1n_1}}} \right) = \frac{e^{-s_{1n_1}}}{\left(1+e^{-s_{1n_1}}\right)^2}, \tag{7}$$

$$\frac{\mathrm{d}s_{2n_1}}{\mathrm{d}e_{1n_1}} = \frac{\mathrm{d}}{\mathrm{d}e_{1n_1}} \left(\sum_{n_1=1}^{I_1} \left(e_{1n_1} \cdot w_{1n_1 21} + ww_{21} \right) \right) = w_{1n_1 21}, \tag{8}$$

$$\frac{\mathrm{d}e_{21}}{\mathrm{d}s_{21}} = \frac{\mathrm{d}}{\mathrm{d}s_{21}} \left(\frac{1}{1+e^{-s_{21}}} \right) = \frac{e^{-s_{21}}}{\left(1+e^{-s_{21}}\right)^2}. \tag{9}$$

Substituting formulas (6)–(8) and (9) to formula (5) gives the following expression:

$$\frac{\mathrm{d}e_{21}}{\mathrm{d}e_{0z}} = \sum_{n_1=1}^{I_1} \left(w_{0z1n_1} \cdot \frac{e^{-s_{1n_1}}}{\left(1+e^{-s_{1n_1}}\right)^2} \cdot w_{1n_1 21} \cdot \frac{e^{-s_{21}}}{\left(1+e^{-s_{21}}\right)^2} \right) \tag{10}$$

or:

$$\frac{\mathrm{d}e_{21}}{\mathrm{d}e_{0z}} = \sum_{n_1=1}^{I_1} \left(\frac{w_{0z1n_1} e^{-s_{1n_1}}}{\left(1+e^{-s_{1n_1}}\right)^2} \cdot \frac{w_{1n_1 21} e^{-s_{21}}}{\left(1+e^{-s_{21}}\right)^2} \right). \tag{11}$$

In the case of more hidden layers the formula (11) can be easy developed. For example, for 2 hidden layers one obtains:

$$\frac{\mathrm{d}e_{31}}{\mathrm{d}e_{0z}} = \sum_{n_1=1}^{I_1} \sum_{n_2=1}^{I_2} \left(\frac{w_{0z1n_1} e^{-s_{1n_1}}}{\left(1+e^{-s_{1n_1}}\right)^2} \cdot \frac{w_{1n_1 2n_2} e^{-s_{2n_2}}}{\left(1+e^{-s_{2n_2}}\right)^2} \cdot \frac{w_{2n_2 31} e^{-s_{31}}}{\left(1+e^{-s_{31}}\right)^2} \right). \tag{12}$$

However, in this work it will not be used.

In order to examine above formulas, consider the following function:

$$f = (x_n) = \sum_{i=1}^{n} \left(\frac{1}{\pi} |x_i - 0.7|^{\pi/3} \left(\frac{\pi}{2} - \cos\left(2\pi m |x_i - 0.7|^{\pi/3} \right) \right) \right), \tag{13}$$

where: n is a number of variables, m defines the number of local optimas by each variable.

In the test n is equal to 2 and the parameter m equals 1. This function is relatively simple, differentiable and has one optimum (global). A more complicated function is not necessary to verify the gradient computation using ANN, because above formulas are used in the algorithm of local optimization method.

The global optimization and its verification for more complicated, multimodal functions will be described in the following sections.

The aim of this test is the comparison of the actual sensitivity of the function in selected points with the sensitivity computed by means of the ANN. The ANN with 11 neurons (2-8-1) is used. The ANN for 18 randomly generated (in domain) training vectors is learned. For another set of the vectors the comparison can give similar results. In the Table 1 the values of sensitivities selected from 100 randomly generated points are shown.

The verification of neural sensitivity analysis for other functions (with one optimum also) has been carried out. The results are similar.

Table 1. The results of the verification of the sensitivity of the ANN

result	$\partial f/\partial x_1$	$\partial f/\partial x_2$
min error	0.01%	0.02%
max error	8.32%	7.85%
average error	2.87%	3.02%

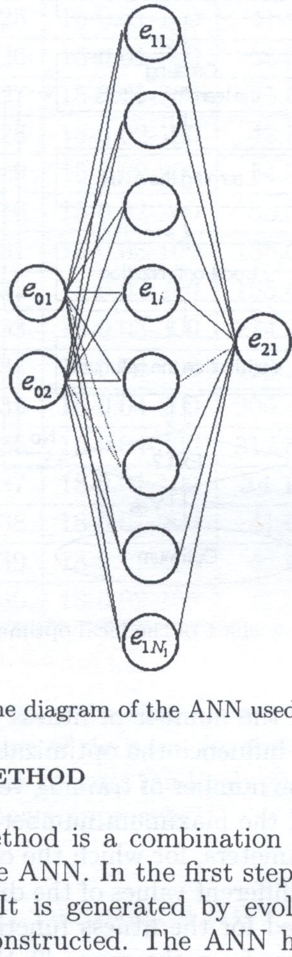

Fig. 3. The diagram of the ANN used in this test

4. The local minimization method

The proposed local minimization method is a combination of the classical gradient method (the steepest descent method) [14] and the ANN. In the first step of the method a set (cloud) of points in a function domain is generated. It is generated by evolutionary computing. To perform the optimization process, the ANN is constructed. The ANN has only one hidden layer, which was explained previously. The number of the neurons in the input layer is equal to the number of design variables of the minimizing function. The number of neurons in the hidden layer is equal to the number of the design variables of the minimizing function multiplied by 2. In the output layer there is only one neuron (which plays the role of the approximation of the fitness function). The starting number of training vectors is equal to 3 powered by the number of design variables of the minimizing function (simple mathematical dependence).

In each iteration of the optimization method a few steps are performed (Fig. 4). In the first step a set of training vectors of the ANN is created. In the first iteration the set is created on the basis of the cloud of points. The coordinates of the points play the role of the input values of the ANN, the fitness values in the points play the role of output value of the ANN. Both values (input and output) are transformed to the range [0; 1], which is necessary in the case of the sigmoid activation function of neurons. In the second step the ANN is trained. In the next, third step, the optimization process is carried out. The fitness function approximation is performed by the ANN. The steepest descent method of optimization is used. To compute the gradient the formula (11) is employed. The fitness function is computed by means of the ANN. For the point, which is the result of the optimization (found in step 3), the actual fitness function is computed. In the last step the stop condition is checked. If the condition is true, the point is treated as the result of the optimization process. If the condition is false, this point is added to the training vector set and the next iteration is carried out (go to step 1).

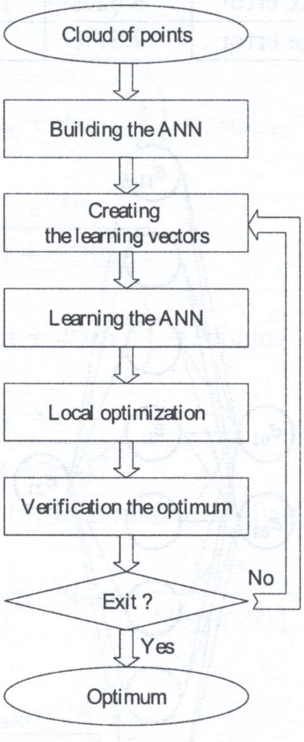

Fig. 4. The flow chart of the local optimization method

The optimization time depends on the number of fitness function computation. There are parameters of algorithm, which strongly influence the optimization time. It is readily noticeable that the most important parameters are: the number of training vectors (in the first iteration), the RMS error of the ANN (learning error) and the maximum number of iterations. The aim of the tests is to choose the best values of these parameters, for which the computation time takes the minimum.

The local optimization method for different values of the discussed parameter was carry out. The test of the minimization was performed for the fitness function (13) ($n = 2$, $m = 1$). The design variables x_1 and x_2 can take the values from the range [0; 1]. The actual minimum of the fitness function is equal to [0.7; 0.7], the fitness value in the optimum is equal to 0. The stop condition for the algorithm concerns the finding point. If both design variables x_1 and x_2 belong to the range [0.68; 0.72], the optimization process is stopped.

The number of training vectors takes the values: 6; 12; 15; 18 and 21. The RMS error takes the values: 0.01; 0.02; 0.03 and 0.04. The maximum number of iterations takes the values: 100, 200 and 300. All combinations (60) of the parameters were checked.

For each combination 100 tests were applied, and then, the results (the number of fitness function computation) were averaged.

The mean results obtained in the tests are shown in the Table 2.

The optimal parameters are: the number of training vectors is equal to 18, the RMS error take the value: 0.01 and the maximum number of iterations is equal to 100.

In the next tests the optimal parameters were used.

The optimal number of training vectors are relatively low. The ANN works as the local approximation tool, and the ANN should map the function in the relatively small environment (region of the local optimum). The ANN does not have to map the function in whole domain.

Table 2. The results of testing (parameters: the number of training vectors, the RMS-error, the maximum number of iterations, results: the number of fitness function computation)

No	parameters	results	No	parameters	results	No	parameters	results
1	6-0.01-100	75.22	21	12-0.03-300	287.12	41	18-0.02-200	51.36
2	6-0.01-200	98.95	22	12-0.04-100	105.24	42	18-0.02-300	55.33
3	6-0.01-300	148.33	23	12-0.04-200	220.42	43	18-0.03-100	97.72
4	6-0.02-100	108.22	24	12-0.04-300	334.78	44	18-0.03-200	175.12
5	6-0.02-200	141.07	25	15-0.01-100	41.95	45	18-0.03-300	218.01
6	6-0.02-300	174.47	26	15-0.01-200	58.00	46	18-0.04-100	102.96
7	6-0.03-100	98.26	27	15-0.01-300	45.68	47	18-0.04-200	209.55
8	6-0.03-200	193.84	28	15-0.02-100	42.39	48	18-0.04-300	314.87
9	6-0.03-300	311.06	29	15-0.02-200	54.32	49	21-0.01-100	120.28
10	6-0.04-100	127.53	30	15-0.02-300	58.33	50	21-0.01-200	41.49
11	6-0.04-200	254.99	31	15-0.03-100	135.09	51	21-0.01-300	39.44
12	6-0.04-300	376.32	32	15-0.03-200	195.47	52	21-0.02-100	60.85
13	12-0.01-100	51.11	33	15-0.03-300	274.21	53	21-0.02-200	60.26
14	12-0.01-200	64.83	34	15-0.04-100	112.27	54	21-0.02-300	58.07
15	12-0.01-300	66.24	35	15-0.04-200	200.44	55	21-0.03-100	113.04
16	12-0.02-100	37.71	36	15-0.04-300	312.88	56	21-0.03-200	174.70
17	12-0.02-200	52.33	37	18-0.01-100	**34.14**	57	21-0.03-300	283.92
18	12-0.02-300	77.63	38	18-0.01-200	41.54	58	21-0.04-100	112.62
19	12-0.03-100	89.66	39	18-0.01-300	42.66	59	21-0.04-200	219.66
20	12-0.03-200	171.07	40	18-0.02-100	47.90	60	21-0.04-300	332.35

In the first iterations the ANN gives results that are not very equal. The results are more equal in the next iterations due to the addition of the point (to the training vectors), which is the result of the local optimization. The introduction of more randomly generated vectors to the training vectors, gives worse results than previously shown approach.

5. The intelligent computing technique (ICT)

The main idea of the creating the intelligent computing technique (ICT) (Fig. 5) as the two-stages strategy is the coupling of the advantages of evolutionary and gradient optimization methods aided by neuro-computing. The evolutionary algorithms can find the global optimum, but it is very time consuming. The gradient methods can find the optimum precisely, but they need information about sensitivity of the objective function.

The ICT (in first stage) uses some properties of the evolutionary algorithms (EA). Those algorithms are procedures to search the optimum in the feasible space of solutions. The EA generates clusters of points. The clusters are positioned closely to the optimum. There is a great possibility that the optimum is the global optimum.

There is a risk that the points are located close to more than one optimum. In this case the second stage (local method) can work unstably. It can be solved in a few ways.

One of the possibilities is to introduce the parameter which describes the maximum size of the cluster. The parameter can be expressed by the radius of the region in domain. The center of the region is equal to the best solution of the EA. All points which are inside the region, belong to the cloud of points. This approach is characterized by a variable number of training vectors. In this case an alternative parameter is introduced. The parameter defines the maximum number of the points

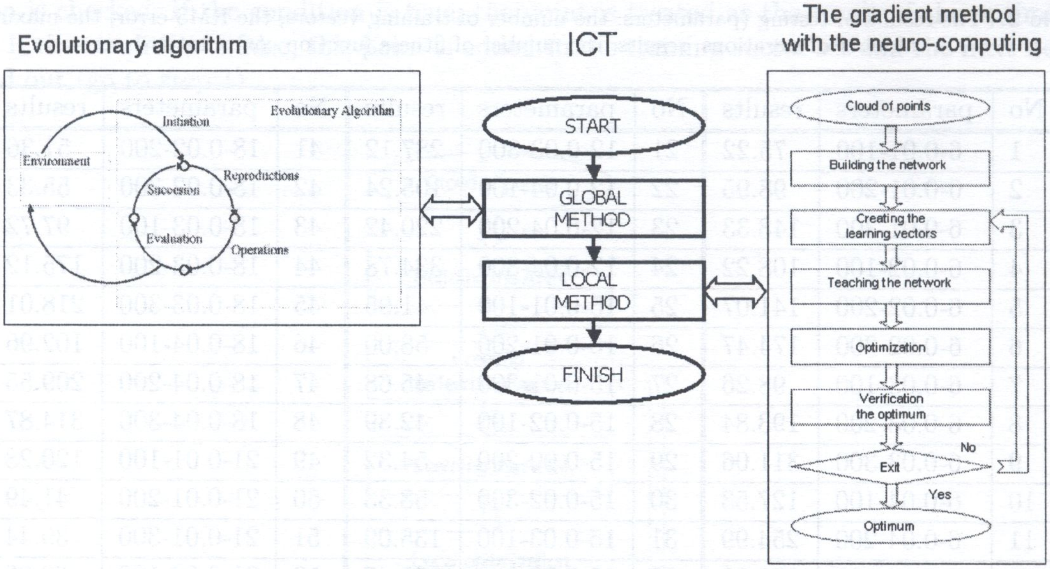

Fig. 5. The diagram of the intelligent computational technique (ICT)

in the cloud. In the case where the EA has generated fewer points inside the region it is necessary to generate additional ones randomly.

In the second stage of the ICT several best points in this region are selected. Then, these points play the role of the cloud and as previously shown, the local method is beginning. This method is based on the gradient method, but the sensitivity analysis is evaluated by the neuro-computing. Therefore, the ICT combines advantages of previous described methods, and avoids the disadvantages.

The crucial problem is the moment of the transition from the first stage to the second one. Some experience allows taking parameters of the ICT, for which the ICT can find the optimum earlier than the EA, which is used separately (see Secs. 6, 7 and 8).

In the general case the moment of transition can depend on some parameters of the first stage: (i) – the changes of fitness function of the best chromosome, (ii) – the size of the clusters of chromosomes, (iii) – the diversification of the population and many others.

6. The examination of intelligent computing technique (ICT)

The aim of the test is to find the minimum of the multimodal function (13) ($m = 3$, $n = 2$). The design variables can take the values from the range $[0; 1]$. The actual minimum of the fitness function is equal to $[0.7; 0.7]$, the fitness function value in the optimum equals 0.

If both design variables x_1 and x_2 belong to range $[0.68; 0.72]$, the optimization process is stopped. The minimization process for two cases is carried out. The introduction of two cases allows assessing the proposed method and the comparison to another known method – the EA. In the first case only the EA is applied. The parameters of the EA are as follows: the probability of the mutation: $pro\_mut = 0.2$, the probability of the crossover: $pro\_cro = 0.2$, population size: $pop\_size = 7$ (the parameters are optimized by the minimal number of fitness function computations of the EA). The number of fitness function computation is equal to 251 (the mean value from 1000 experiments). In the second case the two-stage global method is applied. In the first stage the EA is used. The parameters of the EA are the same as in the first case. The number of generations is equal to 30. The number of fitness function computations equals 95 (the mean value from 1000 experiments).

The number of generations of the EA (first stage) depends on the number of fitness function computations in the case, in which the EA is used only. In the case of the EA working separately (as the optimization method), the optimum was found in 251 fitness function computation. In case of the EA working as the method of preparation the cloud, the time will be shorten, therefore the 30 number of generations were assumed.

If the EA is used to generate the cloud of points only, its working time equals 40% of working time in the case, when the EA is used as the optimization method.

In the second stage the best 9 points (the best one and 8 closest points) in domain as the points of the cloud are selected. The ANN with 11 neurons (2-8-1) is used (Fig. 3).

The results obtained in the next iterations of the second stage are shown in Table 3.

Table 3. The solutions obtained in next steps of the minimization process

No.	x	y	$fitness function$
1	0.521	0.850	0.241
2	0.504	0.793	0.188
3	0.539	0.733	0.124
4	0.587	0.739	0.070
5	0.633	0.740	0.030
6	0.678	0.697	0.004
7	0.692	0.665	0.008
8	0.660	0.673	0.013
9	0.629	0.722	0.028
10	0.616	0.727	0.038
11	0.602	0.732	0.053
12	0.594	0.695	0.055
13	0.612	0.715	0.039
14	0.600	0.687	0.050
15	0.626	0.715	0.028
16	0.606	0.696	0.043
17	0.638	0.668	0.024
18	0.667	0.649	0.019
19	0.664	0.676	0.011
20	0.666	0.686	0.009
21	0.673	0.710	0.006
22	0.693	0.721	0.004
23	0.687	0.731	0.008
24	0.687	0.727	0.007
25	0.679	0.736	0.011
26	0.681	0.731	0.009
27	0.682	0.723	0.007
28	0.679	0.705	0.004
29	0.678	0.700	0.004
30	0.680	0.706	0.004
31	0.703	0.695	0.001

In Table 3 the results from only one experiment are presented, but the results from other experiments are very similar.

The application of the ICT allows decreasing the number of fitness function computation. In the case of using the EA, the number of fitness function computations was equal to 251. In the case of

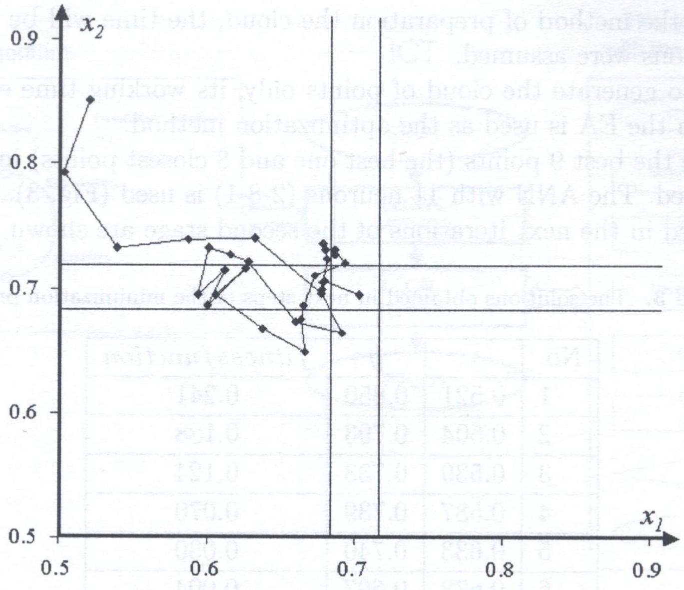

Fig. 6. The graphical representation of Table 3

using the ICT the number of fitness function computation equals 126 (95 + 31). In this case, the computational time was shortened by 49%.

7. Identification of a circular defect in 2-D structure using ICT

Consider a two-dimensional elastic body (plane strains) with a circular hole (Fig. 7).

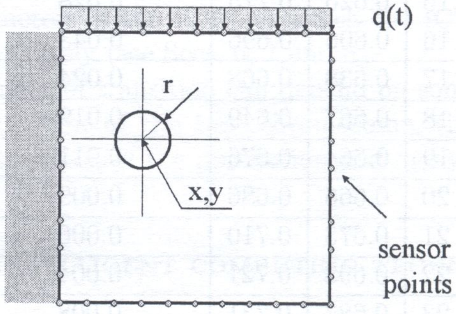

Fig. 7. The plate with the circular defect

The body (1 cm × 1 cm) is loaded dynamically by a traction field $q(t) = qH(t)$, where $q = 100$ kN, $H(t)$ – Heaveside function. The coordinates x, y and the radius r of the hole are unknown. The aim of the test is the identification of the parameters of the hole through the minimization of the objective functional:

$$f = \sum_{i=1}^{n} \int_{\Gamma} \int_{T} \left(\left| \mathbf{u}(\underline{\mathbf{x}}, t) - \hat{\mathbf{u}}(\underline{\mathbf{x}}, t) \right| \right)^2 \delta\left(\underline{\mathbf{x}} - \underline{\mathbf{x}}^i \right) \mathrm{d}t\mathrm{d}\Gamma \tag{14}$$

where: $\mathbf{u}$ and $\hat{\mathbf{u}}$ – measured and computed displacements in boundary sensor points $\underline{\mathbf{x}}^i$ respectively, n – the number of sensor points, δ – Dirac function, Γ – the boundary of the structure, $T = [0, t_f]$ – the time of the analysis, t_f – terminal time.

Displacements $\hat{\mathbf{u}}$ were computed by the boundary element method (BEM) [3]. The actual parameters of the hole are: $x = 0.3$, $y = 0.6$ and $r = 0.1$. If the design variable x belongs to range

[0.28; 0.32], y belongs to range [0.58; 0.62] and r belongs to range [0.08; 0.12], the optimization process is stopped. As in the previous test, the minimization process for two cases is carried out. The introduction of two cases allows assessing the proposed method and the comparison to another known method – the EA, too. In the first case only the EA is applied. The parameters of the EA are as follows: the probability of the mutation: $pro\_mut = 0.2$, the probability of the crossover: $pro\_cro = 0.2$, population size: $pop\_size = 15$ (the parameters are optimized by the minimal number of fitness function computations of the EA). The number of fitness function computation is equal to 213 (the mean value from 25 experiments). In the second case the ICT is applied. In the first stage the EA is used. The parameters of the EA are the same as in the first case. The number of generations is equal to 15. The number of fitness function computations equals 122 (the mean value from 25 experiments).

If the EA is used to generate the cloud of points only, its working time equals 60% of working time in the case, when the EA is used as the optimization method.

In the second stage the best 27 points (the best one and 26 closest points) in the domain as the points of the cloud are selected. The ANN with 10 neurons (3-6-1) is used.

The results obtained in the next iterations of the second stages are shown in the Table 4. In Table 4 the results from only one experiment are presented, but the results from other experiments are very similar.

Table 4. The solutions obtained in the next steps of the minimization process

No.	x [cm]	y [cm]	r [cm]	$fitness function$
1	0.537	0.565	0.229	18.733
2	0.300	0.697	0.122	3.102
3	0.560	0.750	0.130	2.056
4	0.453	0.760	0.098	1.295
5	0.354	0.606	0.233	25.165
6	0.345	0.340	0.220	20.147
7	0.822	0.345	0.086	3.472
8	0.232	0.765	0.123	5.185
9	0.286	0.500	0.092	0.909
10	0.582	0.483	0.139	2.077
11	0.371	0.557	0.132	2.674
12	0.439	0.558	0.123	1.532
13	0.419	0.558	0.116	1.060
14	0.305	0.541	0.111	0.816
15	0.285	0.643	0.110	1.132
16	0.315	0.561	0.106	0.400
17	0.347	0.608	0.105	0.430
18	0.309	0.674	0.103	0.773
19	0.284	0.532	0.102	0.351
20	0.324	0.591	0.102	0.125
21	0.304	0.594	0.101	0.074

The application of the ICT allows decreasing the number of fitness function computations. In the case of using the EA, the number of fitness function computation was equal to 213. In the case of using the ICT the number of fitness function computation was equal to 133. In this case, the computational time was shortened by 38%.

8. Identification of a spherical defect in 3-D structure using ICT

Consider a three-dimensional elastic body with a spherical hole (Fig. 8).

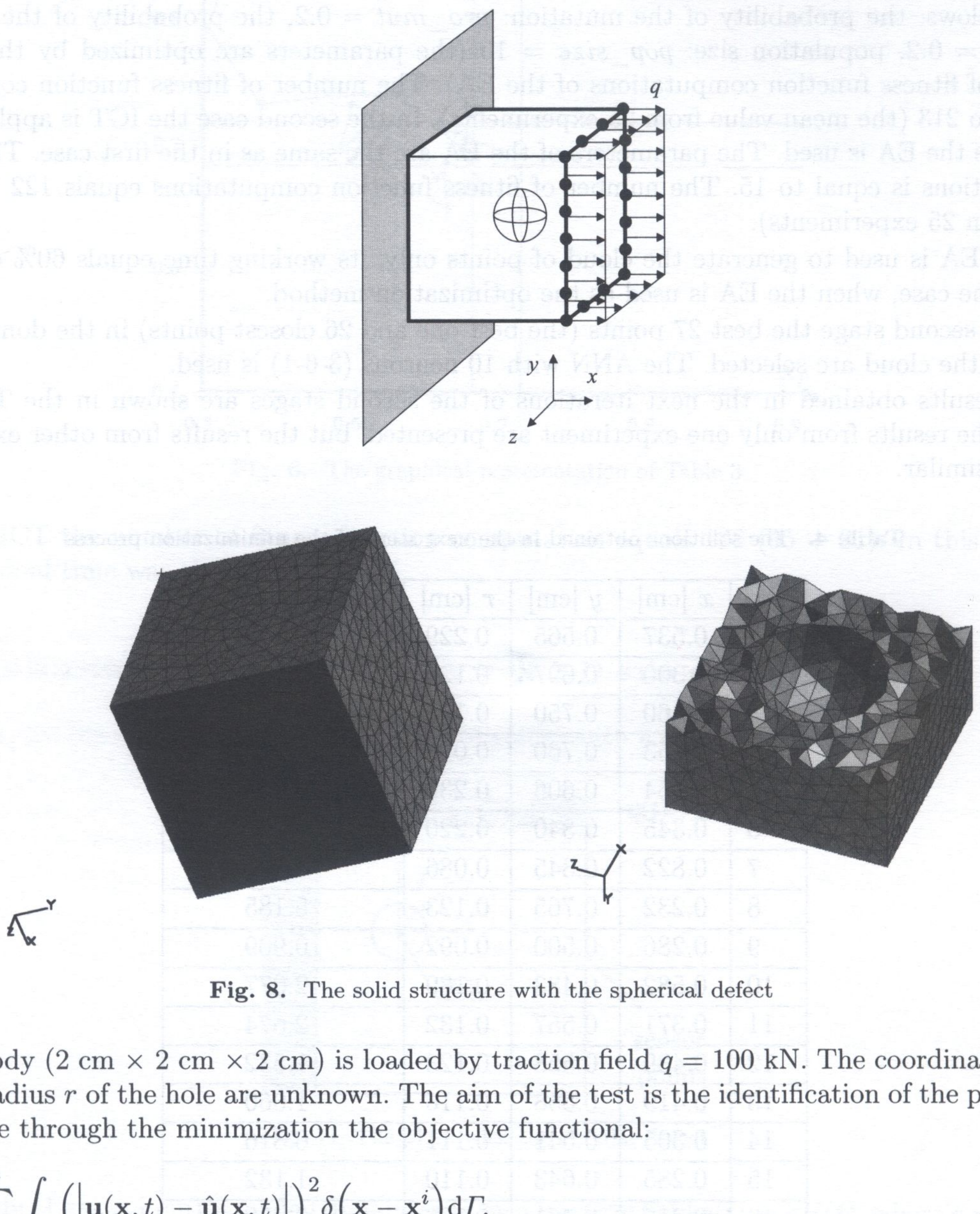

Fig. 8. The solid structure with the spherical defect

The body (2 cm × 2 cm × 2 cm) is loaded by traction field $q = 100$ kN. The coordinates x, y, z and the radius r of the hole are unknown. The aim of the test is the identification of the parameters of the hole through the minimization the objective functional:

$$f = \sum_{i=1}^{n} \int_{\Gamma} \left(\left| \mathbf{u}(\underline{\mathbf{x}}, t) - \hat{\mathbf{u}}(\underline{\mathbf{x}}, t) \right| \right)^2 \delta\left(\underline{\mathbf{x}} - \underline{\mathbf{x}}^i \right) \mathrm{d}\Gamma, \tag{15}$$

where: $\mathbf{u}$ and $\hat{\mathbf{u}}$ – measured and computed displacements in boundary sensor points $\underline{\mathbf{x}}^i$ respectively, n – the number of sensor points, δ – Dirac function, Γ – the boundary of the structure.

Displacements $\hat{\mathbf{u}}$ were computed by the finite element method (FEM) [11].

The actual parameters of the hole are: $x = 0.0$, $y = 0.0$, $z = 0.0$ and $r = 0.05$. The stop condition for the algorithm concerns the best finding point. If design variables x, y and z belong to range $[-0.02; 0.02]$ and r belongs to range $[0.04; 0.06]$, the optimization process is stopped. Like in the previous test the minimization process for two cases is carried out. In the first case (only the EA) the number of fitness function computation was equal to 424. In the second case (two stages strategy) the number of fitness function computation was equal to 262 $(240 + 22)$.

If the EA is used to generate the cloud of points only, its working time equals 60% of working time in the case, when the EA is used as the optimization method.

The results obtained in the next iterations of the second stages are shown in the Table 5.

Table 5. The solutions obtained in next steps of the minimization process

No.	x [cm]	y [cm]	z [cm]	r [cm]	*fitnessfunction*
1	0.917	−0.859	−0.224	0.872	678.375
2	−0.348	0.826	0.766	−0.255	459.342
3	0.797	0.658	−0.282	−0.665	180.966
4	0.308	−0.528	−0.015	0.109	330.086
5	0.490	−0.408	−0.084	0.230	342.055
6	0.324	−0.146	0.019	0.431	183.236
7	−0.187	−0.178	0.036	0.364	51.921
8	−0.335	0.014	−0.163	−0.137	144.331
9	−0.046	−0.176	−0.117	0.246	59.003
10	0.160	0.081	−0.009	−0.070	46.073
11	0.005	0.054	−0.098	−0.132	22.476
12	−0.098	−0.142	−0.120	0.092	39.734
13	0.122	0.033	−0.144	−0.048	40.022
14	−0.045	−0.113	0.060	−0.011	62.204
15	−0.121	0.042	0.033	−0.042	61.173
16	−0.075	0.000	−0.055	0.107	30.029
17	0.063	0.020	−0.035	−0.043	21.277
18	−0.017	0.071	−0.074	−0.009	32.943
19	0.026	0.045	−0.034	−0.003	27.299
20	−0.005	−0.051	0.008	0.008	19.156
21	−0.025	0.010	0.020	−0.009	13.092
22	0.006	−0.004	−0.020	−0.049	8.052

The application of the ICT allows decreasing the number of fitness function computations from 424 to 262. In this case, the computational time was shortened by 38% (like in previous section).

9. Conclusions

In the paper a new approach of global optimization method, named the intelligent computational technique (ICT) has been presented. The ICT combines classical techniques with the methods of artificial intelligence. The ICT is based on the two-stage strategy. In the first stage the evolutionary algorithm is used as the global optimization method. In the second stage the special local method which combines the gradient method and the artificial neural network is applied. Due to finding the optimal parameters of the second stage of the ICT, the computation time of whole ICT is minimized.

Several numerical tests and examples have been presented. The ICT gives better results than when the EA is used alone. The computational time is decreased even to 49%. In the cases, in which the time of computing of the fitness function is meaningful (the BEM or the FEM analysis) this method is particularly useful.

The crucial problem of the ICT is the moment of transition from the first stage to the second one. This problem will be examined in details in next works.

References

[1] J. Arabas. *Evolutionary Algorithms Lectures* (in Polish). WNT, 2001.

[2] H. D. Bui. *Inverse Problems in the Mechanics of Materials: An Instroduction.* CRC Press, Bocca Raton, 1994.

[3] T. Burczyński. *Boundary Elements Method in Mechanics* (in Polish). WNT, Warszawa 1995.

[4] T. Burczyński, W. Beluch, A. Długosz, P. Orantek, M. Nowakowski. Evolutionary methods in inverse problems of engineering mechanics. In: M. Tanaka and G. S. Dulikravich, eds., *Inverse Problems in Engineering Mechanics II,* pp. 553–562, Elsevier, 2000.

[5] T. Burczyński, W. Beluch, A. Długosz, W. Kuś, M. Nowakowski, P. Orantek. Evolutionary computation in optimization and identification. *Computer Assisted Mechanics and Engineering Sciences,* **9**: 3–20, 2002.

[6] T. Burczyński, E. Majchrzak, W. Kuś, P. Orantek, M. Dziewoński. Evolutionary computation in inverse problems. In: T. Burczyński and A. Osyczka, eds., *Evolutionary Methods in Mechanics,* pp. 33–46, Kluwer, 2004.

[7] T. Burczyński, P. Orantek. *Evolutionary algorithms aided by sensitivity information. Artificial Neural Nets and Genetic Algorithms.* V. Kurkowa, N. Stelle, R. Neruda, M. Karny, eds., Springer Computer Science, pp. 272–275, Wien, 2001.

[8] T. Burczyński, P. Orantek. *Application of Neural Networks in Controlling of Evolutionary Algorithms. Computational Mechanics.* S. Valliappan, N. Khalili, eds., Elsevier, pp. 1271–1276, London 2001.

[9] T. Burczyński, P. Orantek. Application of Artificial Neutal Network in Computational sensitivity analysis. Recent Developments In: *Artificial Intelligence Methods,* T. Burczyński, W. Cholewa, W. Moczulski, AI-METH Series pp. 37–38, Gliwice, November, 2004.

[10] W. Duch, J. Korbicz, L. Rutkowski, R. Tadeusiewicz. *Neural Networks* (in Polish), vol. 6, Exit, Warsaw 2000.

[11] M. Kleiber et al. *Parameter Sensitivity in Nonlinear Mechanics: Theory and Finite Element Computations.* J. Wiley&Sons, New York, 1997.

[12] J. J. Montano, A. Palmer, Numeric sensitivity analysis applied to feedforward neural networks. *Neural Computing & Application,* **12**: 119–125, Springer-Verlag, London, 2003.

[13] R. Schaefer. *The basis of the genetic global optimization* (in Polish). Wydawnictwo Uniwersytetu Jagiellońskiego, Kraków, 2002.

[14] J. Seidler, A. Badach, W. Molisz. *The methods of solving the optimization problems* (in Polish.) WNT, Warszawa, 1980.

[15] Xiaotong Wang, Chih-Chen Chang, Fang Du. Achieving a more robust neural network model for control of a MR damper by signal sensitivity analysis. *Neural Computing & Application,* **10**: 330–338, Springer-Verlag, London, 2002.