

## Intelligent computing in inverse problems

Tadeusz Burczyński<sup>1,2</sup>, Witold Beluch<sup>1</sup>, Adam Długosz<sup>1</sup>,  
Piotr Orantek<sup>1</sup>, Antoni Skrobol<sup>1</sup>

<sup>1</sup> Department for Strength of Materials and Computational Mechanics  
Silesian University of Technology, Konarskiego 18a, 44-100 Gliwice, Poland

<sup>2</sup> Department of Artificial Intelligence, Institute of Computer Modelling  
Cracow University of Technology, Warszawska 24, 31-155 Kraków, Poland

(Received in the final form February 15, 2006)

This paper presents a review of intelligent computing techniques in solving inverse mechanics problems. These techniques are based on Evolutionary Algorithms (EAs) and the coupling of Evolutionary Algorithms (EAs) and Artificial Neural Networks (ANNs) in the form of Computational Intelligence Systems (CISs). The main attention was focused on the identification of the defects such as voids or cracks in structures on the basis of the knowledge about displacements, temperature and eigenfrequencies. The identification of the unknown number, position, size and kind of defects in the elastic structures is shown. The paper contains a lot of tests and numerical examples.

**Keywords:** intelligent computing (IC), evolutionary algorithms (EAs), artificial neural networks (ANNs), fuzzy inference system (FIS), computational intelligent system (CIS), finite element method (FEM), boundary element method (BEM)

### 1. INTRODUCTION

Most of catastrophic failures of mechanical structures were caused by the appearance of material defects such as voids, cracks or inclusions etc. There are several non-destructive methods, used for the identification of such defects but only few of them are capable of finding internal defects, which in some cases are very difficult to detect.

There are many methods that allow the identification of internal defects on the basis of the knowledge about boundary state fields such as displacements, temperature or natural frequencies. There are several approaches to identification problems. One group of methods is based on sensitivity analysis [3]. This approach is very elegant and strict from mathematical point of view but it sometimes fails because the minimization of identification functions leads to a local minimum.

Another group of methods is based on techniques which simulate or imitate biological systems and processes. These methods are sometimes called *intelligent techniques* because they have some features which fulfil some criteria of artificial intelligence. ANNs and EAs belong to this group. The Artificial Neural Networks [21] have been used for identification problems, [22, 26–28]. In this method there is a difficulty with the identification of a large number of different kinds of defects, especially when the number of defects is unknown. In the second very common approach the evolutionary algorithms are used in identification tasks, [5–11]. The EA enables the identification of multiple defects. It can distinguish different kinds of defects such as voids and cracks and a number of defects can be considered a design variable.

The main goal of this paper is to review various intelligent techniques, elaborated by authors, based on evolutionary computing and coupling the EA and the ANN in defect identification problems.

The paper is devoted to develop and examine solution intelligent techniques for non-destructive identification of multiple internal defects (crack and voids) in mechanical systems being under static, dynamical, thermal loads and in the free vibration state. These techniques are based on minimization approach performed by the evolutionary algorithms and, using the finite or boundary element methods, they solve the direct problem.

**2. FORMULATION OF THE INVERSE PROBLEM**

Consider an elastic body containing some defects in the form voids and/or cracks. The body occupies the domain  $\Omega$  bounded by boundary  $\Gamma$  (Fig. 1). Within the domain  $\Omega$  both the temperature  $T$  and displacements  $u_i$  are unknown. Appropriate differential equations defining the distribution of these two fields are described by linear thermoelasticity,

$$k T_{,ii} = 0, \quad \mathbf{x} \in \Omega, \tag{1}$$

$$G u_{i,jj}(\mathbf{x}) + \frac{G}{1-2\nu} u_{j,ji}(\mathbf{x}) + b_i(\mathbf{x}) = 0, \quad \mathbf{x} \in \Omega, \tag{2}$$

where  $k$  is thermal conductivity,  $G$  and  $\nu$  are the shear modulus and Poisson's ratio, respectively.

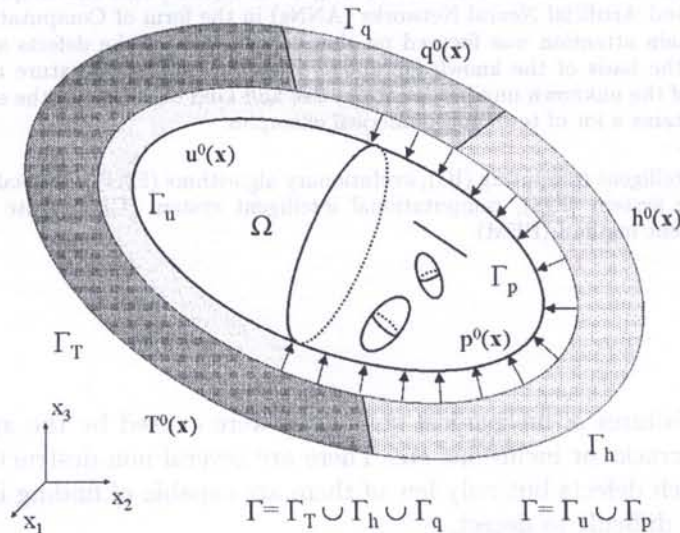


Fig. 1. The body with defects

The body force  $b_i$  due to the thermal expansion is defined as

$$b_i = -\frac{2G(1+\nu)}{1-2\nu} \alpha T_{,i} \tag{3}$$

where  $\alpha$  stands for the coefficient of thermal expansion.

The boundary conditions associated with heat conduction are defined on three sub-boundaries  $\Gamma_T$ ,  $\Gamma_q$  and  $\Gamma_h$ ,

$$\begin{aligned} T(\mathbf{x}) &= T^0(\mathbf{x}), & \mathbf{x} \in \Gamma_T, \\ q(\mathbf{x}) &= q^0(\mathbf{x}), & \mathbf{x} \in \Gamma_q, \\ q(\mathbf{x}) &= h [T(\mathbf{x}) - T_f(\mathbf{x})], & \mathbf{x} \in \Gamma_h, \end{aligned} \tag{4}$$

where  $q$  is the heat flux,  $T_f$  is the temperature of the fluid and  $h$  is heat transfer coefficient.

The topological sum of sub-boundaries  $\Gamma_T$ ,  $\Gamma_q$  and  $\Gamma_h$  forms the boundary  $\Gamma = \Gamma_T \cup \Gamma_q \cup \Gamma_h$ . The intersection of these sub-boundaries is empty,  $\Gamma_T \cap \Gamma_q = \emptyset$ ,  $\Gamma_T \cap \Gamma_h = \emptyset$ ,  $\Gamma_q \cap \Gamma_h = \emptyset$ .

Boundary conditions for elasticity are prescribed in the form

$$\begin{aligned} u_i(\mathbf{x}) &= u_i^0(\mathbf{x}), & \mathbf{x} \in \Gamma_u, \\ p_i(\mathbf{x}) &= p_i^0(\mathbf{x}), & \mathbf{x} \in \Gamma_p, \end{aligned} \tag{5}$$

where  $p_i$  are tractions. Sub-boundaries  $\Gamma_u$  and  $\Gamma_p$  fulfil the following conditions:  $\Gamma_u \cup \Gamma_p = \Gamma$ ,  $\Gamma_u \cap \Gamma_p = \emptyset$ .

In the crack case the displacements are allowed a jump across the boundary of the crack:  $[[u_i]] = u_i^+ - u_i^- \neq 0$ .

In the case of time-dependent thermoelastic problems boundary conditions should be completed by initial conditions.

The number, shapes and sizes of the defects are unknown. One should find them having information about measured displacements  $u_i$ , temperature  $T$  or natural frequencies.

The aim of the identification problem is to find the vector of parameters  $ch$ , describing the number, shape and position of the defects. The lack of this information is compensated by some knowledge about displacements, temperature or natural frequencies (*redundant data*). The usual approach to solve an identification problem is the minimization of some measure of distance between simulated numerically (by the FEM or the BEM) and measured experimentally displacements, temperature or natural frequencies. The identification problem is expressed as the minimization of the special objective function (or functional)  $J$  with respect to a design vector which is represented by a chromosome  $ch$ ,

$$\min_{ch} J. \tag{6}$$

The objective function  $J$  depends on state fields such as displacements, temperature or natural frequencies. A few typical objective functions can be expressed as follows:

- for displacements under static loading:

$$J_u = \int_{\Gamma} \varphi[\mathbf{u}(\mathbf{y})] d\Gamma, \tag{7}$$

- for temperature in a steady state:

$$J_T = \int_{\Gamma} \varphi[T(\mathbf{y})] d\Gamma, \tag{8}$$

- for displacements under dynamical loading:

$$J_{ut} = \int_0^{t_f} \int_{\Gamma} \psi[\mathbf{u}(\mathbf{y}, t)] d\Gamma dt, \tag{9}$$

- for transient heat conduction:

$$J_{Tt} = \int_0^{t_f} \int_{\Gamma} \psi[T(\mathbf{y}, t)] d\Gamma dt, \tag{10}$$

- for eigenfrequencies:

$$J_{\omega} = \sum_{j=1}^M (\omega_j - \hat{\omega}_j)^2, \tag{11}$$



where integrands are given as follows,

$$\begin{aligned}\varphi[\mathbf{u}(\mathbf{y})] &= [\mathbf{u}(\mathbf{y}) - \hat{\mathbf{u}}(\mathbf{y})]^2, \\ \varphi[T(\mathbf{y})] &= [T(\mathbf{y}) - \hat{T}(\mathbf{y})]^2, \\ \psi[\mathbf{u}(\mathbf{y}, t)] &= [\mathbf{u}(\mathbf{y}, t) - \hat{\mathbf{u}}(\mathbf{y}, t)]^2, \\ \psi[T(\mathbf{y}, t)] &= [T(\mathbf{y}, t) - \hat{T}(\mathbf{y}, t)]^2.\end{aligned}\quad (12)$$

The fields  $\hat{\mathbf{u}}(\mathbf{y})$ ,  $\hat{T}(\mathbf{y})$ ,  $\hat{\mathbf{u}}(\mathbf{y}, t)$ ,  $\hat{T}(\mathbf{y}, t)$  and  $\hat{\omega}_j$  are measured fields of displacements, temperature and natural frequency. In the case when these fields are measured in discrete sensor boundary points  $\mathbf{y}_i$  at the time  $t_k$  integrands have the forms

$$\begin{aligned}\varphi[\mathbf{u}(\mathbf{y})] &= \sum_{i=1}^N \delta(\mathbf{y} - \mathbf{y}_i) [\mathbf{u}(\mathbf{y}) - \hat{\mathbf{u}}(\mathbf{y})]^2, \\ \varphi[T(\mathbf{y})] &= \sum_{i=1}^N \delta(\mathbf{y} - \mathbf{y}_i) [T(\mathbf{y}) - \hat{T}(\mathbf{y})]^2, \\ \psi[\mathbf{u}(\mathbf{y}, t)] &= \sum_{i=1}^N \sum_{k=1}^M \delta(\mathbf{y} - \mathbf{y}_i) \delta(t - t_k) [\mathbf{u}(\mathbf{y}, t) - \hat{\mathbf{u}}(\mathbf{y}, t)]^2, \\ \psi[T(\mathbf{y}, t)] &= \sum_{i=1}^N \sum_{k=1}^M \delta(\mathbf{y} - \mathbf{y}_i) \delta(t - t_k) [T(\mathbf{y}, t) - \hat{T}(\mathbf{y}, t)]^2.\end{aligned}\quad (13)$$

It is possible to create the same combination of objective functions (7)–(11). The more general equivalent objective function has the form

$$J = \eta_u J_u + \eta_T J_T + \eta_{ut} J_{ut} + \eta_{Tt} J_{Tt} + \eta_\omega J_\omega \quad (14)$$

where  $\eta_u$ ,  $\eta_T$ ,  $\eta_{ut}$ ,  $\eta_{Tt}$ ,  $\eta_\omega$  are non-negative weights indicating the relative importance of each objective functions.

In order to evaluate the fitness functions (7)–(11) one should solve the direct problem, which is described by Eqs. (1) and (2). This can be done by the finite element method (FEM) [19, 29] or the boundary element method (BEM) [4]

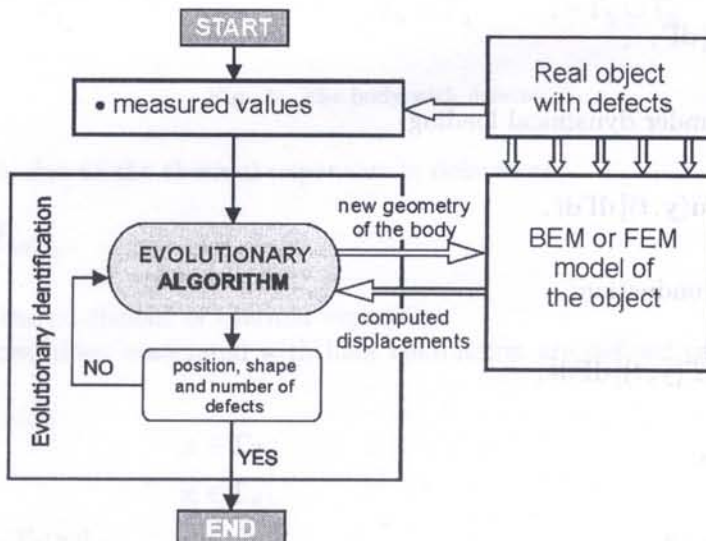


Fig. 2. The evolutionary identification using BEM or FEM to compute the fitness function

The problem of defect identification solved by means of the evolutionary algorithm and FEM or BEM is presented in Fig. 2.

### 3. GEOMETRICAL PARAMETERIZATION

The key problem in identification is the selection of design variables which enable the description of the shapes, positions, types and number of defects.

The defects in 2-D structures are modelled as: (i) a circular, (ii) an elliptical, (iii) any arbitrary shape by using the closed NURBS curve and (iv) cracks (Fig. 3). In the case of a 3-D structure the defect is modelled as: (i) a spherical, (ii) an ellipsoidal and (iii) any arbitrary shape using the closed NURBS surfaces are presented in Fig. 4 [20].

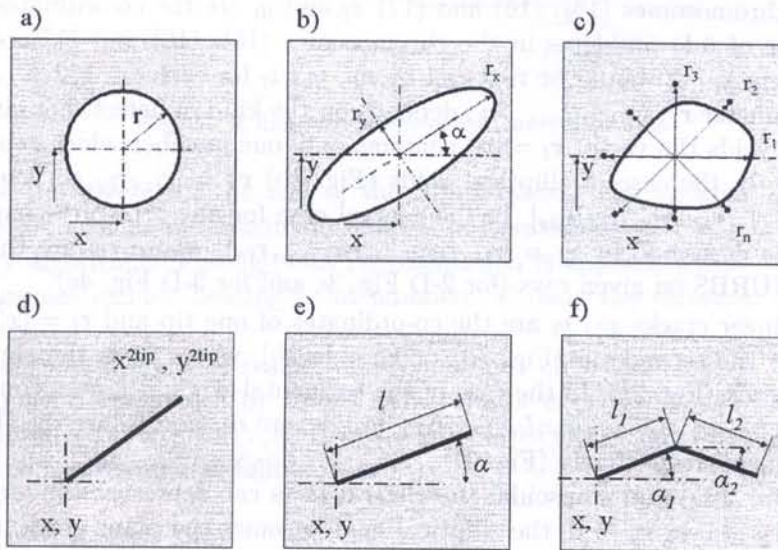


Fig. 3. The modelled forms of the defects (2-D)

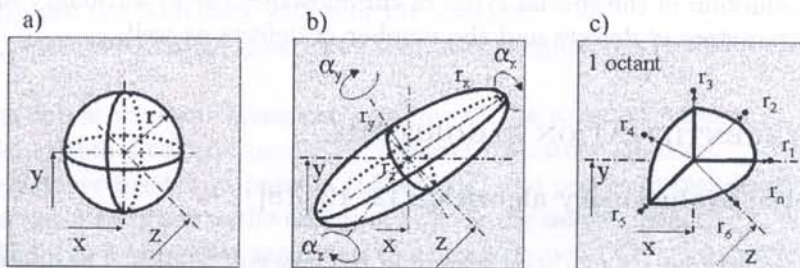


Fig. 4. The modelled forms of the defects (3-D)

There are a few possibilities of building chromosomes for defect identification problems in which the number of defects is unknown. It is necessary to assume the maximal number of defects  $n_{max}$  and the algorithm should find the actual number  $n \in \{0; n_{max}\}$  of defects and their positions, shape and kind.

Consider the 2-D case. The first possibility is to use the following type of the chromosome

$$ch = [n, x_1, y_1, r_1, x_2, y_2, r_2, \dots, x_l, y_l, r_l, \dots, x_{n_{max}}, y_{n_{max}}, r_{n_{max}}] \tag{15}$$

where the gene  $n \in \{0; n_{max}\}$  is responsible for the actual number of voids.

The second type of chromosomes is simpler

$$\mathbf{ch} = [x_1, y_1, \mathbf{r}_1, x_2, y_2, \mathbf{r}_2, \dots, x_l, y_l, \mathbf{r}_l, \dots, x_{n_{\max}}, y_{n_{\max}}, \mathbf{r}_{n_{\max}}] \quad (16)$$

and the actual number of defects is controlled by the conditions  $|\mathbf{r}_l| < r_{\min}$ , where  $r_{\min} (\rightarrow 0)$  is an admissible minimal value. If this condition is fulfilled, the  $l$ -th void does not exist.

The third type of chromosomes can be built as follows,

$$\mathbf{ch} = [w_1, w_2, \dots, w_l, \dots, w_{n_{\max}}, x_1, y_1, \mathbf{r}_1, x_2, y_2, \mathbf{r}_2, \dots, x_l, y_l, \mathbf{r}_l, \dots, x_{n_{\max}}, y_{n_{\max}}, \mathbf{r}_{n_{\max}}] \quad (17)$$

where a controlling parameter (indication)  $w_l = \{\text{true}, \text{false}\}$ , associated with the  $l$ -th void, shows if the void exists (*true*) or does not exist (*false*).

In all types of chromosomes (15), (16) and (17)  $x_l$  and  $y_l$  are the co-ordinates of the centres of defects. In the case of 3-D problems in the chromosomes (15), (16) and (17) co-ordinates of the centres of the defects  $x_l, y_l$  should be replaced by  $x_l, y_l, z_l$  for each  $l = 1, 2, \dots, n_{n_{\max}}$ .

The vector parameter  $\mathbf{r}_l, l = 1, 2, \dots, n$ , depends on the kind of defect. For circular (Fig. 3a) or spherical (Fig. 4a) voids the vector  $\mathbf{r}_l = [r_l]$  contains only one member which represents the radius  $r_l$  of the  $l$ -th void. In the case of elliptical voids (Fig. 3b)  $\mathbf{r}_l = [r_{lx}, r_{ly}, \alpha_l]$  and ellipsoidal voids (Fig. 4b)  $\mathbf{r}_l = [r_{lx}, r_{ly}, r_{lz}, \alpha_{lx}, \alpha_{ly}, \alpha_{lz}]$ . In the general case for any arbitrary shape of the void the vector parameter is described by  $\mathbf{r}_l = [r_{l1}, r_{l2}, \dots, r_{ls}, \dots, r_{ln}]$ , where  $r_{ls}$  are the positions of the control points of NURBS on given rays (for 2-D Fig. 3c and for 3-D Fig. 4c).

In the case of linear cracks  $x_l, y_l$  are the co-ordinates of one tip and  $\mathbf{r}_l = [x_l^{2\text{tip}}, y_l^{2\text{tip}}]$  contains the co-ordinates of the second tip (Fig. 3d) or  $\mathbf{r}_l = [\alpha_l, l_l]$ , where  $\alpha_l$  is the slope angle and  $l_l$  is the length of the crack (Fig. 3e). In the case of the segmental-straight cracks consisting of  $R$  linear segments  $\mathbf{r}_l = [\alpha_{l1}, l_{l1}, \alpha_{l2}, l_{l2}, \dots, \alpha_{lR}, l_{lR}, \dots, \alpha_{lR}, l_{lR}]$ , where  $\alpha_{lr}$  and  $l_{lr}$  are the slope angle and the length of each segment, respectively (Fig. 3f).

The elliptical (for 2-D) and ellipsoidal (for 3-D) defects can represent the cracks defects in the limit case. When  $r_x \rightarrow 0$  or  $r_y \rightarrow 0$ , the elliptical void becomes the plane crack. In the similar way the ellipsoidal void transforms into a spatial crack.

If the actual number of defects  $n$  is less than the maximal number  $n_{\max}$  ( $n < n_{\max}$ ), geometrical parameters  $x_l, y_l$  and  $\mathbf{r}_l$  associated with the non-generated defects, exist in the chromosome as not active genes.

Due to the introduction of the special types of chromosomes the evolutionary algorithm can find the geometrical parameters of defects and the number of defects as well.

## 4. INTELLIGENT IDENTIFICATION ALGORITHMS

### 4.1. The sequential evolutionary algorithm [2, 12, 18]

The evolutionary algorithms are methods searching the space of solutions basing on the analogy to the biological evolution of species. Like in biology, the term of an individual is used, and it represents a single solution. The evolutionary algorithms operate on populations of individuals, so while an algorithm works, all the time a set of problem solutions is dealt with. An individual consists of chromosomes. Usually, it is assumed that an individual has one chromosome. Chromosomes consist of genes which are equivalents of design variables in optimisation problems. The adaptation is computed by using the fitness function. All genes of an individual decide about the fitness function value. Figure 5 shows how the evolutionary algorithm works.

In the first step a initial population of individuals is created. Usually, the values of the genes of particular individuals are randomly generated. In the next step the individuals' fitness function value is computed. Then, evolutionary operators change genes of the parent population individuals, individuals are selected for the offspring population, which becomes a parent population and the

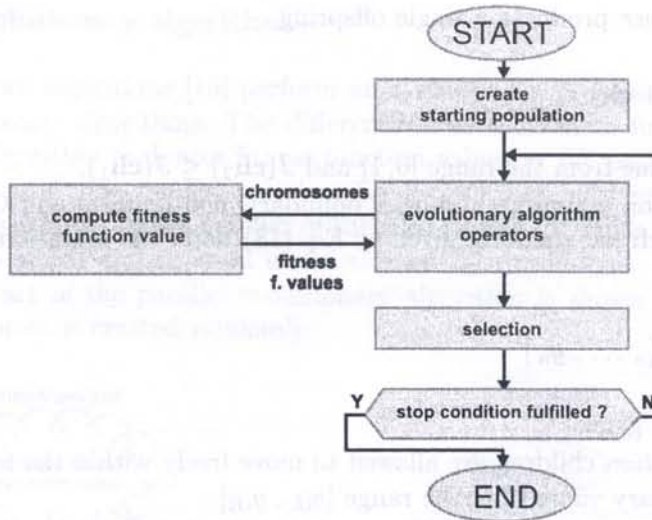


Fig. 5. A flowchart of the evolutionary algorithm

algorithm is done iteratively till the end of the computation. The termination condition of the computation is formulated as the maximum number of iterations.

In evolutionary algorithms floating-point representation is applied, which means that genes included in chromosomes include floating-point numbers. Usually the variation of the gene value is limited.

A single-chromosome individual (called a chromosome)  $\mathbf{ch}_k$ ,  $k = 1, 2, \dots, N$ , where  $N$  is the population size, may be presented by means of a column or a row matrix, whose elements are represented by genes  $g_i$ ,  $i = 1, 2, \dots, n$ ,  $n$  – the number of genes in a chromosome. The sample chromosome can be is presented as follows,

$$\mathbf{ch} = [g_1, g_2, \dots, g_i, \dots, g_n]. \tag{18}$$

The genes are real numbers on which constraints are imposed in the form

$$g_{iL} \leq g_i \leq g_{iR}; \quad i = 1, 2, \dots, n, \tag{19}$$

where  $g_{iL}$  and  $g_{iR}$  are the boundary values of the  $i$ -th gene.

Evolutionary operators change gene value like the biological mechanisms of crossing and mutation.

The crossover operator swaps some chromosome of the selected parents in order to create the offspring. Simple, arithmetical and heuristic crossover operators are used.

The *simple crossover* needs two parents (p1 and p2) and produces two descendants (d1 and d2). The simple crossover may produce the offspring outside the design space. To avoid this, a parameter  $\alpha \in [0, 1]$  is applied. For a randomly generated crossing parameter  $i$  it works as follows (chromosomes  $\mathbf{ch}_1, \mathbf{ch}_2$  are parents),

$$\begin{aligned} p1: \quad \mathbf{ch}_1 &= [g_1, g_2, \dots, g_i, \dots, g_n] \\ p2: \quad \mathbf{ch}_2 &= [e_1, e_2, \dots, e_i, \dots, e_n] \end{aligned} \tag{20}$$

$$\begin{aligned} d1: \quad \mathbf{ch}'_1 &= [g_1, \dots, g_i, \alpha e_{i+1} + (1-\alpha)g_{i+1}, \dots, \alpha e_n + (1-\alpha)g_n] \\ d2: \quad \mathbf{ch}'_2 &= [e_1, \dots, e_i, \alpha g_{i+1} + (1-\alpha)e_{i+1}, \dots, \alpha g_n + (1-\alpha)e_n] \end{aligned} \tag{21}$$

The *arithmetical crossover* gives two descendants, which are a linear combination of two parents

$$\mathbf{ch}'_1 = \alpha \mathbf{ch}_1 + (1 - \alpha) \mathbf{ch}_2, \quad \mathbf{ch}'_2 = \alpha \mathbf{ch}_2 + (1 - \alpha) \mathbf{ch}_1. \tag{22}$$

The *heuristic crossover* produces a single offspring

$$\mathbf{ch}'_1 = r(\mathbf{ch}_2 - \mathbf{ch}_1) + \mathbf{ch}_2 \quad (23)$$

where  $r$  is a random value from the range  $[0, 1]$  and  $J(\mathbf{ch}_2) \leq J(\mathbf{ch}_1)$ .

Four kinds of mutation operators: uniform, boundary, non-uniform and Gaussian mutation, are used. If a chromosome  $\mathbf{ch}$  has the form given by Eq. (18), after the mutation it can be presented as follows,

$$\mathbf{ch} = [g_1, g_2, \dots, g'_i, \dots, g_n] \quad (24)$$

where  $g'_i$  is the mutated gene.

In the *uniform mutation* children are allowed to move freely within the feasible domain and the gene  $g'_i$  takes any arbitrary value from the range  $[g_{iL}, g_{iR}]$ .

In the *boundary mutation* the chromosome can take only the boundary values of the design space,  $g'_i = g_{iL}$  or  $g'_i = g_{iR}$ .

The operator of the *non-uniform mutation* depends on the generation number  $t$  and is employed in order to tune the system

$$g'_i = \begin{cases} g_i + \Delta(t, g_{iR} - g_i) & \text{if a random digit is 0,} \\ g_i - \Delta(t, g_i - g_{iL}) & \text{if a random digit is 1,} \end{cases} \quad (25)$$

where the function  $\Delta$  takes a value from the range  $[0, e]$ .

In the *Gaussian mutation* a gene  $g_i$  is mutated by adding a random variable  $\Delta g_i$  with the normal distribution

$$g'_i = g_i + \Delta g_i. \quad (26)$$

A special type of mutation, the so called *gradient mutation*, is used. This mutation is characterized by a genetic interference, which means the modification of a gene by adding the term  $\Delta g_i = \beta_i h_i$  which depends on the sensitivity of the fitness function  $h_i = h_i(\partial J / \partial g_i)$  and a coefficient  $\beta_i$  determining a step increment in a search direction  $h_i$ .

An important element of the evolutionary algorithm is the mechanism of selection. The probability of the individual's survival depends on the value of the fitness function. There are a few methods of selection: the roulette wheel selection, the tournament selection and the ranking selection. The ranking selection is performed in a few steps. First, the individuals are classified according to the value of the fitness function, then a rank value is attributed to each individual. It depends on the individual's number and the rank function. The best individuals obtain the highest rank value, the worst obtain the lowest one. In the final step individuals for the offspring generation are drawn, but the probability of drawing particular individuals is closely related to their rank value.

The sequential evolutionary algorithms are well known tools for global optimization but the number of fitness function evaluations during optimization is equal to thousands or even hundreds of thousands. The fitness function evaluation for most real life problems connected with mechanics or mechanical engineering takes a lot of time (from seconds to hours). The long computation time can be shortened when the parallel or distributed evolutionary is used.

The fitness function evaluation is done in a parallel way when the parallel evolutionary algorithms are used. The distributed evolutionary algorithms operate on many subpopulations. The parallelization of the distributed evolutionary algorithm leads to two cases: the first one in which each subpopulation uses a different processor, and the second one when the different processors can be used by each chromosome of the subpopulations.



## 4.2. The parallel evolutionary algorithm

The parallel evolutionary algorithms [15] perform an evolutionary process in the same manner as the sequential evolutionary algorithms. The difference is in the fitness function evaluation. The parallel evolutionary algorithm evaluates fitness function values in the parallel way. Theoretically, the maximum reduction of time needed to solve the optimization problem by means of parallel evolutionary algorithms is equal to the number of the used processing units. The maximum number of the processing units which can be used is constrained by the number of chromosomes in the population. The flowchart of the parallel evolutionary algorithm is shown in Fig. 6. The starting population of chromosomes is created randomly.

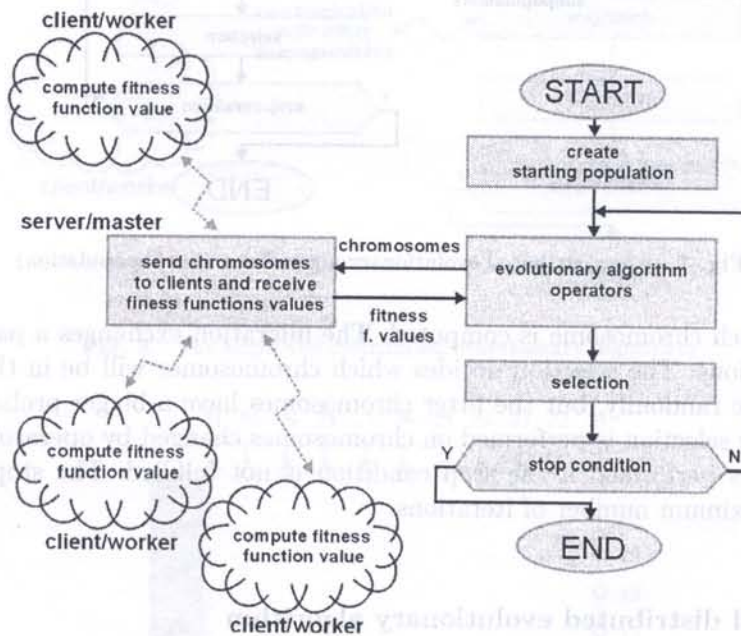


Fig. 6. The parallel evolutionary algorithm

The evolutionary operators change chromosomes and the fitness function value for each chromosome is computed. The server/master transfers chromosomes to clients/workers. The workers compute the fitness function and send it to the server. The workers operate on different processing units. The selection is performed after computing the fitness function value for each chromosome. The selection decides which chromosomes will be in the new population. The selection is done randomly, but the fitter chromosomes have a bigger probability to be in the new population. The next iteration is performed if the stop condition is not fulfilled. The stop condition can be expressed as the maximum number of iterations.

## 4.3. The distributed evolutionary algorithm

The distributed evolutionary algorithms (DEA) [1, 25] work similarly to many evolutionary algorithms operating on subpopulations. The evolutionary algorithms exchange chromosomes during a migration phase between subpopulations. The flowchart of the DEA is presented in Fig. 7. When the DEA is used, the number of fitness function evaluations can be lower in comparison with the sequential and parallel evolutionary algorithms. The DEA works in a parallel manner, usually. Each of the evolutionary algorithms in the DEA works on a different processing unit. The theoretical reduction of time could be bigger than the number of processing units. The starting subpopulation of chromosomes is created randomly. The evolutionary operators change chromosomes and the fitness

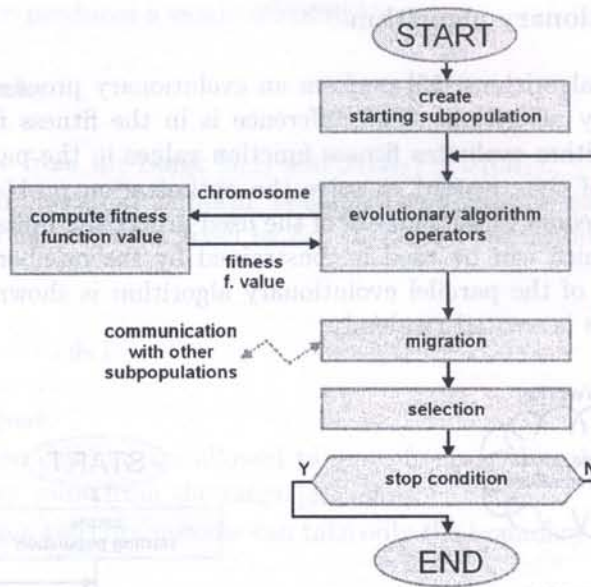


Fig. 7. The distributed evolutionary algorithm (one subpopulation)

function value for each chromosome is computed. The migration exchanges a part of chromosomes between subpopulations. The selection decides which chromosomes will be in the new population. The selection is done randomly, but the fitter chromosomes have a bigger probability to be in the new population. The selection is performed on chromosomes changed by operators and immigrants. The next iteration is performed if the stop condition is not fulfilled. The stop condition can be expressed as the maximum number of iterations.

#### 4.4. The improved distributed evolutionary algorithm

To improve the scalability of the distributed evolutionary algorithm, mechanisms from the parallel evolutionary algorithm can be used. The simplest improvement is computing the fitness function values in a parallel way. The maximum number of processing units which can be used is equal to the sum of chromosomes in subpopulations instead of the number of subpopulations. The flowchart of the modified distributed evolutionary algorithm is presented in Fig. 8.

### 5. THE TESTING OF THE IDENTIFICATION METHOD

#### 5.1. The selection of the sensor points in the identification problems

One of more important issues in an identification problem is the selection of the sensor points. The great number of sensor the points provides more information about the system but it can complicate the identification process. In order to check the influence of the number and positions of the sensor points on the identification process, a few tests have been carried out.

The aim of the presented test is to identify a circular defect in a 2-D structure (plane stress) (Fig. 9). The direct problem is described in the framework of elastodynamics. Boundary conditions in the form displacements  $u_i = u_i^0 = 0$  and tractions  $p_2 = p_2^0 H(t)$ ,  $p_2^0 = 10$  [kN/m],  $H(t)$  – Heaveside’s function, are prescribed and initial conditions are homogeneous. In order to identify the void, the objective function (9) was proposed. Measured displacements in 31 sensor boundary points were simulated numerically for the actual position and size of the void by using the BEM. Various 12 combinations of the sensor points were examined (see Table 1).

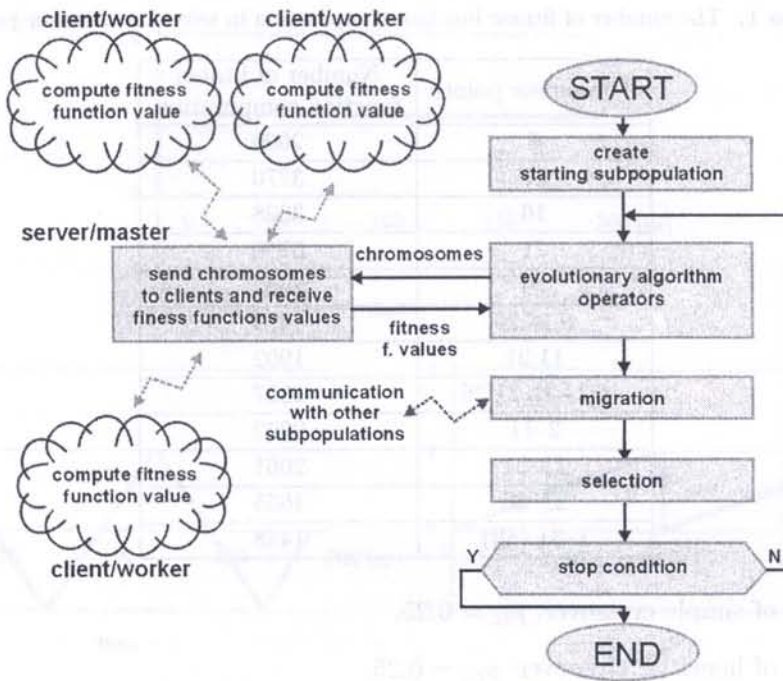


Fig. 8. The improved distributed evolutionary algorithm

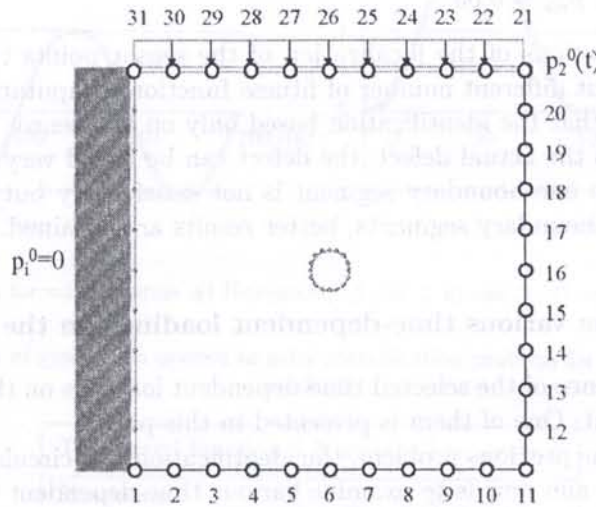


Fig. 9. The plate with the circular defect

In the test, for all combinations, 10 independent experiments were carried out. Each experiment started from randomly population. Finally, the results were averaged. The following parameters of the evolutionary algorithm were applied:

- the population size:  $pop\_size = 100$ ,
- the maximum number of generations:  $max\_life = 100$ ,
- the probability of uniform mutation:  $p_{um} = 0.25$ ,
- the probability of non-uniform mutation:  $p_{nm} = 0.35$ ,
- the probability of boundary mutation:  $p_{bm} = 0.05$ ,

**Table 1.** The number of fitness function computation in selection of sensor points

The sensor points	Number of fitness function computation
6	2600
11	3770
16	3808
21	3760
26	2632
6,16,26	1475
11,21	1992
6,11,16,21,26	1507
2-11	2202
11-21	2061
21-30	1655
1-31 (all)	1488

- the probability of simple crossover:  $p_{sc} = 0.25$ ,
- the probability of heuristic crossover:  $p_{hc} = 0.25$ ,
- the probability of arithmetical crossover:  $p_{ac} = 0.25$ ,
- the cloning probability:  $p_{scl} = 0.05$ .

For all selected combinations of the localization of the sensor points the identification process has finished successfully but different number of fitness function computation was required.

Obtained results show that the identification based only on one sensor point is very difficult. If the sensor point is close to the actual defect, the defect can be found very fast. The localization of many sensor points on the one boundary segment is not satisfactory but if the sensor points are localized on the middle of boundary segments, better results are obtained.

## 5.2. The influence of the various time-dependent loadings on the identification results

In order to check the influence of the selected time-dependent loadings on the identification process, many tests were carried out. One of them is presented in this paper.

Consider, similarly to the previous problem, the identification of a circular void in a elastic structure (Fig. 9) but the main aim now is to examine various time-dependent loadings  $p_2^0(t)$  (Fig. 10): a) Heaveside's; b) Saw 1; c) Saw 2; d) Sinusoidal 1; e) Sinusoidal 2. The sensor points are situated at nodes: 6, 11, 16, 21 and 26.

In the test, for all combination, 10 independent experiments were carried out. Each experiment started from a randomly population. Finally, numerical results were averaged.

The following parameters of the evolutionary algorithm were applied:

- the population size:  $pop\_size = 100$ ,
- the maximum number of generations:  $max\_life = 100$ ,
- the probability of uniform mutation:  $p_{um} = 0.25$ ,
- the probability of non-uniform mutation:  $p_{nm} = 0.35$ ,
- the probability of boundary mutation:  $p_{bm} = 0.05$ ,
- the probability of simple crossover:  $p_{sc} = 0.25$ ,

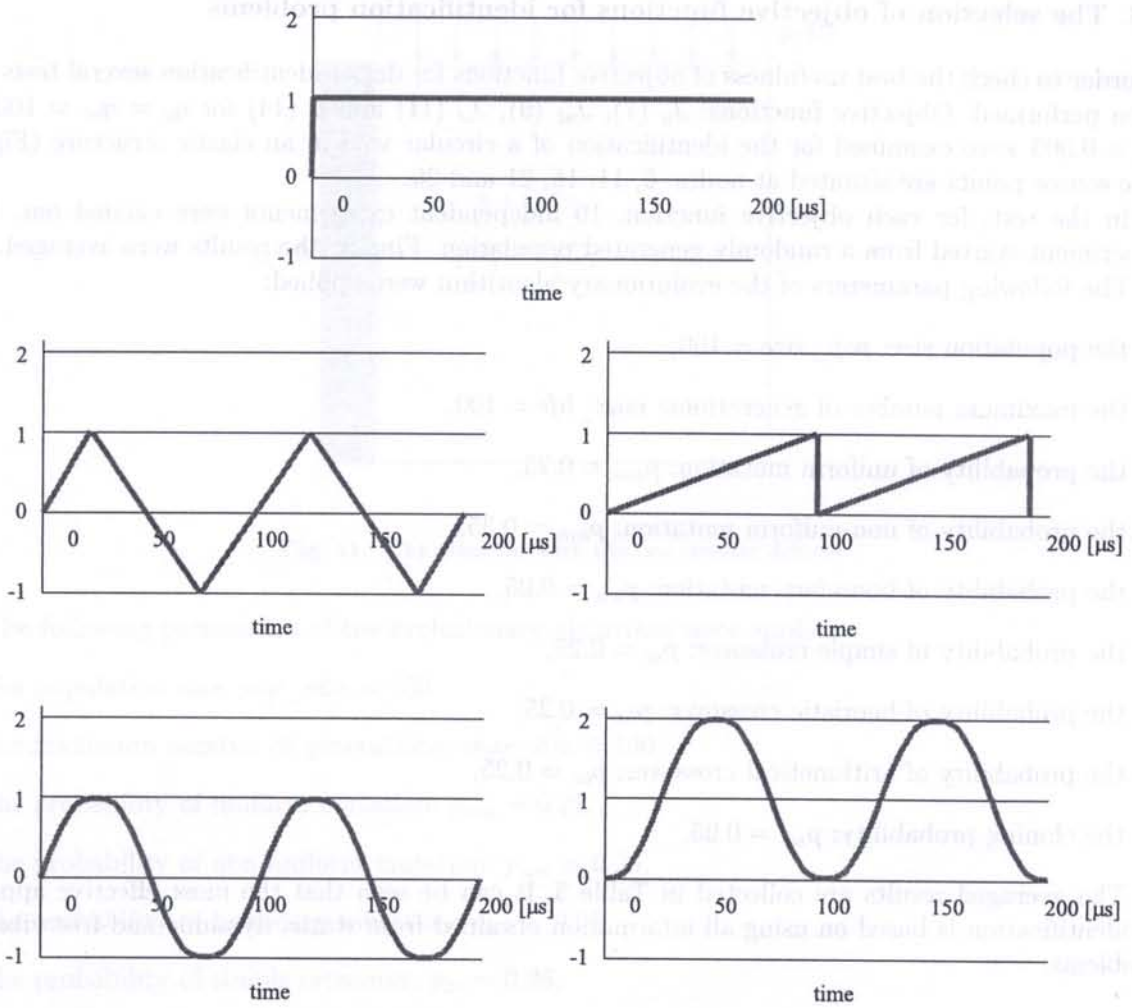


Fig. 10. The different forced functions: a) Heaveside, b) saw 1, c) saw 2, d) sinusoidal 1, e) sinusoidal 2

Table 2. The number of generation needed to solve identification problem for different time-dependent loadings

The forced function	Number of generation
Heaveside	20
Saw 1	39
Saw 2	16
Sinusoid 1	91
Sinusoid 2	72

- the probability of heuristic crossover:  $p_{hc} = 0.25$ ,
- the probability of arithmetical crossover:  $p_{ac} = 0.25$ ,
- the cloning probability:  $p_{scl} = 0.05$ .

The averaged results of the numbers of generations needed for the identification are presented in Table 2.

The obtained results show that the minimal number of generations, needed for obtaining good results of the identification, is for non-continuous time-dependent functions: (i) saw 2, (ii) Heaveside, (iii) saw 1. Unfortunately, practical realization of such loadings is not easy.

### 5.3. The selection of objective functions for identification problems

In order to check the best usefulness of objective functions for defect identification several tests have been performed. Objective functions:  $J_u$  (7),  $J_{ut}$  (9),  $J_\omega$  (11) and  $J$  (14) for  $\eta_u = \eta_{ut} = 100$  and  $\eta_\omega = 0.003$  were examined for the identification of a circular void in an elastic structure (Fig. 9). The sensor points are situated at nodes: 6, 11, 16, 21 and 26.

In the test, for each objective function, 10 independent experiments were carried out. Each experiment started from a randomly generated population. Finally, the results were averaged.

The following parameters of the evolutionary algorithm were applied:

- the population size:  $pop\_size = 100$ ,
- the maximum number of generations:  $max\_life = 100$ ,
- the probability of uniform mutation:  $p_{um} = 0.25$ ,
- the probability of non-uniform mutation:  $p_{nm} = 0.35$ ,
- the probability of boundary mutation:  $p_{bm} = 0.05$ ,
- the probability of simple crossover:  $p_{sc} = 0.25$ ,
- the probability of heuristic crossover:  $p_{hc} = 0.25$ ,
- the probability of arithmetical crossover:  $p_{ac} = 0.25$ ,
- the cloning probability:  $p_{scl} = 0.05$ .

The averaged results are collected in Table 3. It can be seen that the most effective approach to identification is based on using all information obtained from static, dynamic and free vibration problems.

**Table 3.** The number of fitness function computation for different objective functions

The information	Number of fitness function computation
The displacements in the static problem $J_u$	11850
The frequencies in eigenvalue problem $J_\omega$	11846
The displacements in the dynamic problem $J_{ut}$	6452
The all information $J = \eta_u J_u + \eta_\omega J_\omega + \eta_{ut} J_{ut}$	2176

### 5.4. Tests for the identification of multiple voids

The identification of multiple defects is much more complicated as for single defects. This problem is especially difficult when a distance between both voids is very small. In this section the applications of evolutionary computation and different objective functions are analysed.

Consider a 2-D elastic structure which contains two circular voids (Fig. 11). The distance between them is described by a parameter  $R$  which takes different values,  $R = 1, 3, 5, 7, 9$ . In the case when  $R = 1$  both voids overlap partly and create a new void with a complicated shape. To solve the identification problem four objective functions:  $J_u$  (7),  $J_{ut}$  (9),  $J_\omega$  (11) and  $J$  (14) for  $\eta_u = \eta_{ut} = 100$ ,  $\eta_\omega = 0.003$ , and  $\eta_T = \eta_{Tt} = 0$  were examined. The sensor points are situated at nodes: 6, 11, 16, 21 and 26.

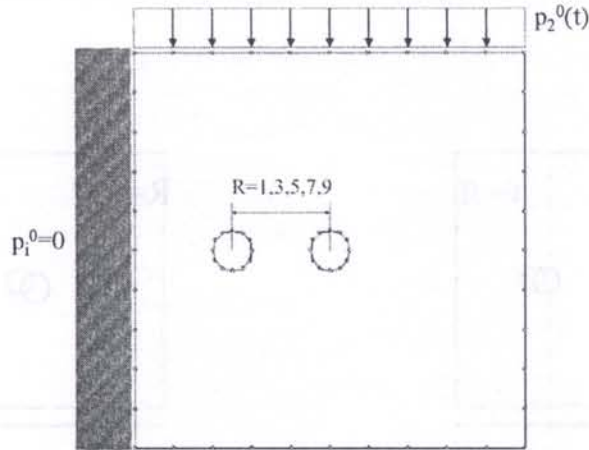


Fig. 11. The plate the with the two circular defects

The following parameters of the evolutionary algorithm were applied:

- the population size:  $pop\_size = 100$ ,
- the maximum number of generations:  $max\_life = 100$ ,
- the probability of uniform mutation:  $p_{um} = 0.25$ ,
- the probability of non-uniform mutation:  $p_{nm} = 0.35$ ,
- the probability of boundary mutation:  $p_{bm} = 0.05$ ,
- the probability of simple crossover:  $p_{sc} = 0.25$ ,
- the probability of heuristic crossover:  $p_{hc} = 0.25$ ,
- the probability of arithmetical crossover:  $p_{ac} = 0.25$ ,
- the cloning probability:  $p_{scl} = 0.05$ .

The results of the identification for different values of the parameter  $R$  and for various objective functions are presented in Figs. 12 to 15.

In the cases, in which the distance  $R$  between defects is large, the objective functions  $J_u$  (7) and  $J_{ut}$  (9) are sufficient. The application of only  $J_\omega$  (11) is not satisfactory. For a small distance  $R$  the applications of all single objective functions  $J_u$  (7),  $J_{ut}$  (9),  $J_\omega$  (11) does not give good results. Only the mixed functional  $J$  (14) for  $\eta_u = \eta_{ut} = 100$ ,  $\eta_\omega = 0.003$ , and  $\eta_T = \eta_{Tt} = 0$  provides very good results of the identification.

## 6. THE IDENTIFICATION OF VOIDS

Numerical tests of identification were carried out for 2-D and 3-D structures with internal defects.

### 6.1. The identification of a void with arbitrary shape

A 2-D elastic structure (Fig. 16), loaded dynamically by tractions field  $p = p_0 \sin \omega ft$ , contains an unknown void which shape is parameterised by a NURBS curve (see Fig. 3c).

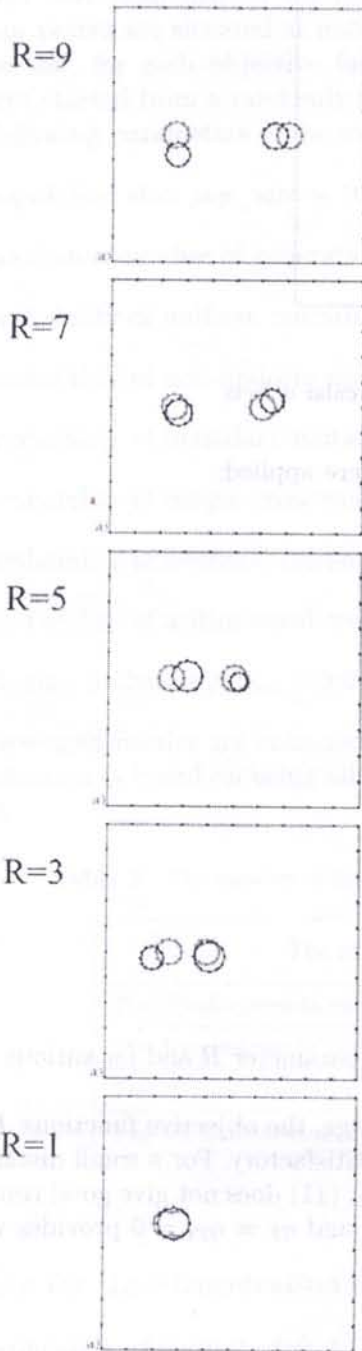


Fig. 12. Identification of two circular defects for the objective function  $J_u$  (7)

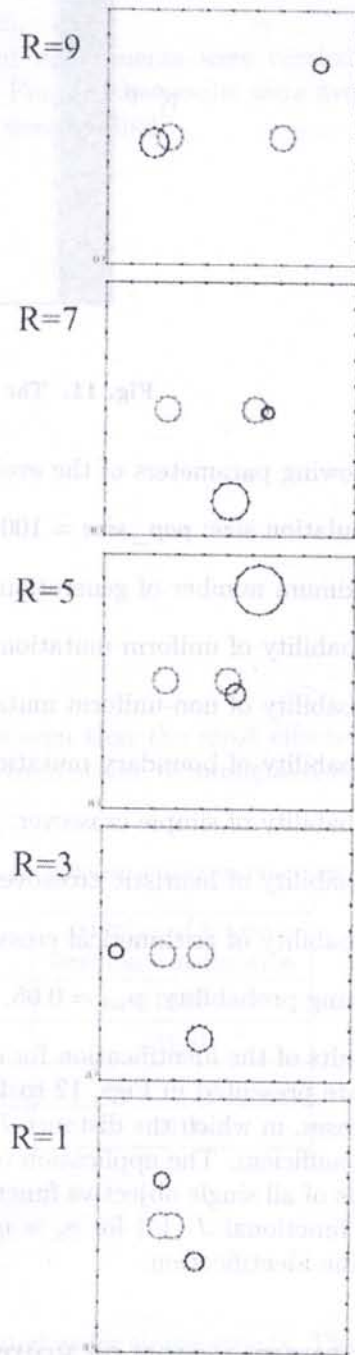


Fig. 13. Identification of two circular defects for the objective function  $J_w$  (11)



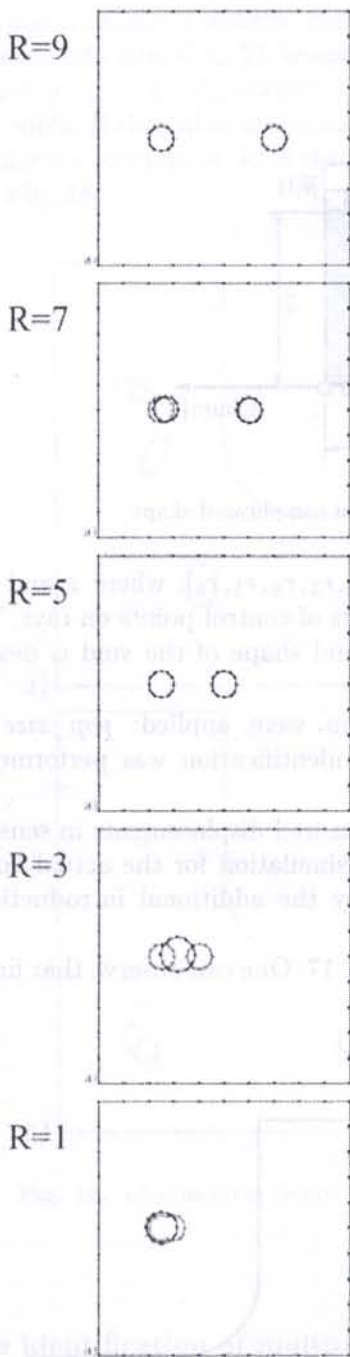


Fig. 14. Identification of two circular defects for the objective function  $J_{ut}$  (9)

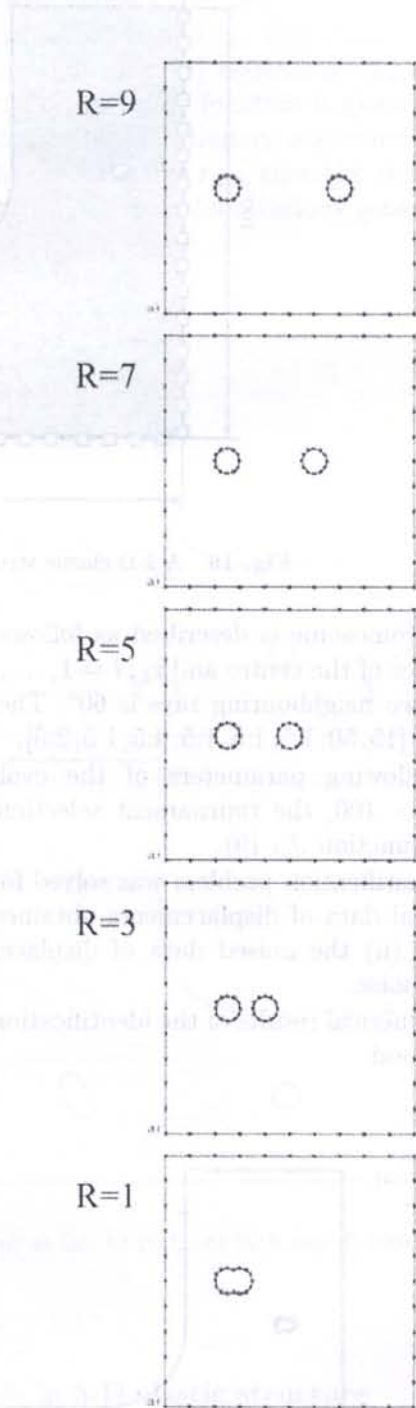


Fig. 15. Identification of two circular defects for the objective function  $J = \eta_u J_u + \eta_w J_w + \eta_{ut} J_{ut}$  (14)

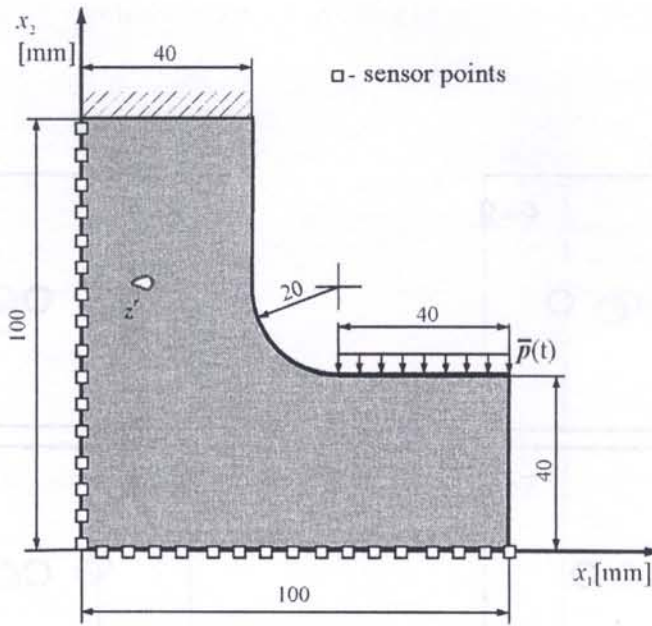


Fig. 16. A 2-D elastic structure with the void of complicated shape

The chromosome is described as follows:  $\mathbf{ch} = [x, y, r_1, r_2, r_3, r_4, r_5, r_6]$ , where  $x$  and  $y$  are the co-ordinates of the centre and  $r_l, l = 1, \dots, 6$ , are the positions of control points on rays. The angle between two neighbouring rays is  $60^\circ$ . The actual position and shape of the void is described by  $z^r \equiv \mathbf{ch} = [15, 50, 1.5, 1.5, 1.5, 4.5, 1.5, 2.5]$ .

The following parameters of the evolutionary algorithm were applied:  $pop\_size = 600$ ,  $max\_life = 100$ , the tournament selection. The process of identification was performed for the objective function  $J_{ut}$  (9).

The identification problem was solved for two kinds of measured displacements in sensor points: (i) the ideal data of displacements obtained from numerical simulation for the actual void by the BEM and (ii) the noised data of displacements obtained by the additional introduction of the Gaussian noise.

The numerical results of the identification are shown in Fig. 17. One can observe that final results are very good.

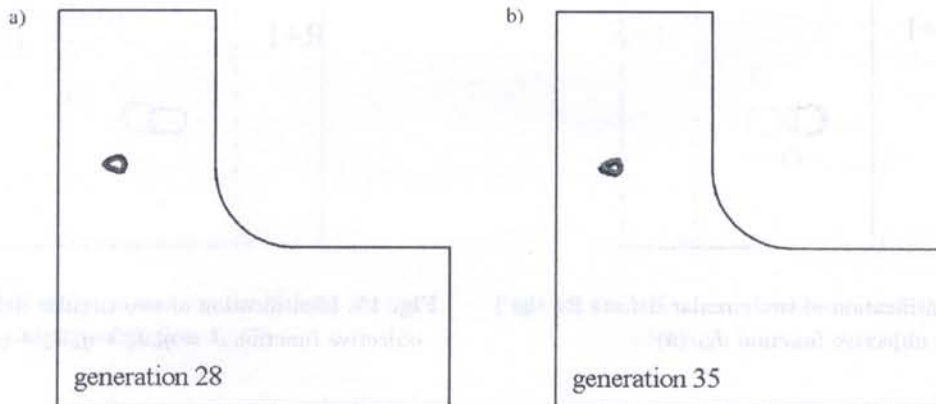


Fig. 17. Identification of the void of complicated shape: a) for ideal data of measured displacements, b) for noised data of measured displacements

## 6.2. The identification of multiple voids with different shapes

A 2-D elastic structure, showed in Fig. 16 contains two circular voids and an elliptical one instead one void. All voids are parameterised by the elliptical description. Their actual shape parameters are as follow:  $\mathbf{z}^1 = [70, 20, 3, 3, 0]$ ;  $\mathbf{z}^2 = [20, 70, 2, 2, 0]$ ;  $\mathbf{z}^3 = [20, 20, 6, 3, 1]$ . The identification task is to find the number of defects, their size and coordinates having: (i) eigenvalues  $\omega_i$ ,  $i = 1, 2, 3$ ; (ii) displacements  $\mathbf{u}(x, t)$  in 21 boundary sensor points. The objective function is given as follows,  $J = \eta_u J_u + \eta_\omega J_\omega$ . The chromosome had 15 genes, because the evolutionary algorithm could find  $n_{\max} = 3$  voids. If the value of the radius is less than the critical value  $r_{\min}$  then the void vanishes. The population consists of 3000 chromosomes. The best solutions in four various generations are shown in Fig. 18.

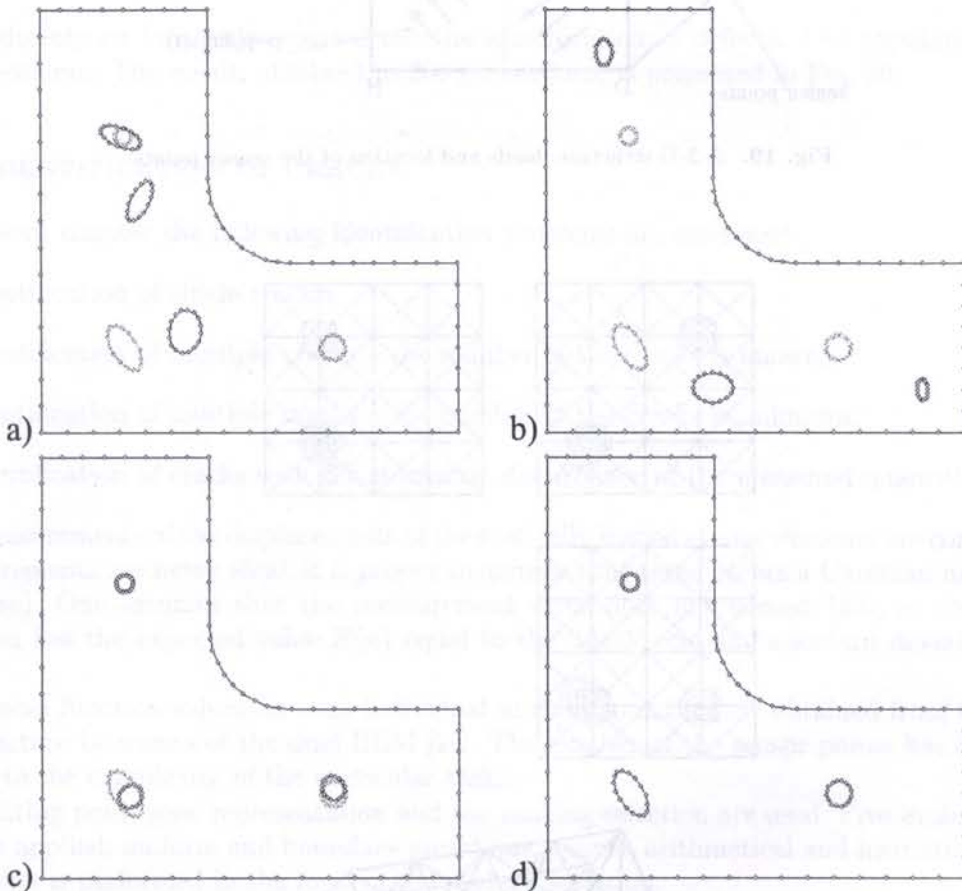


Fig. 18. Identification results for generation number: a) 1st, b) 10th, c) 50th and d) 100th

## 6.3. The identification of multiple spherical voids in 3-D elastic structure

A 3-D structure – the cube with a 20 [cm] side, showed Fig. 19, has one wall supported, while the opposite one is subjected to the harmonic load  $p = p_0 \sin \omega t$ . The load is uniformly distributed on the wall and has a different direction in each quarter ( $p_0 = 15000$  [N/m<sup>2</sup>],  $\omega = 31$  [rad/s]). The mass density of the structure is  $\rho = 100$  [kg/m<sup>3</sup>], the shear modulus  $G = 1 \cdot 10^6$  [Pa] and the Poisson's ratio  $\nu = 0.25$ . The structure contains two internal defects in a form of spherical voids, which parameters – the coordinates of centres and radii – are given in the Table 4 as the actual parameters. The evolutionary algorithm, using the values of amplitudes in 64 sensor points,

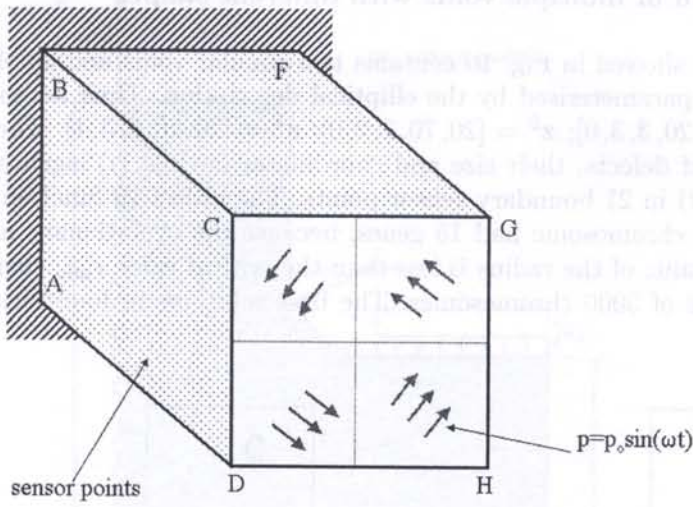


Fig. 19. A 3-D structure: loads and location of the sensor points

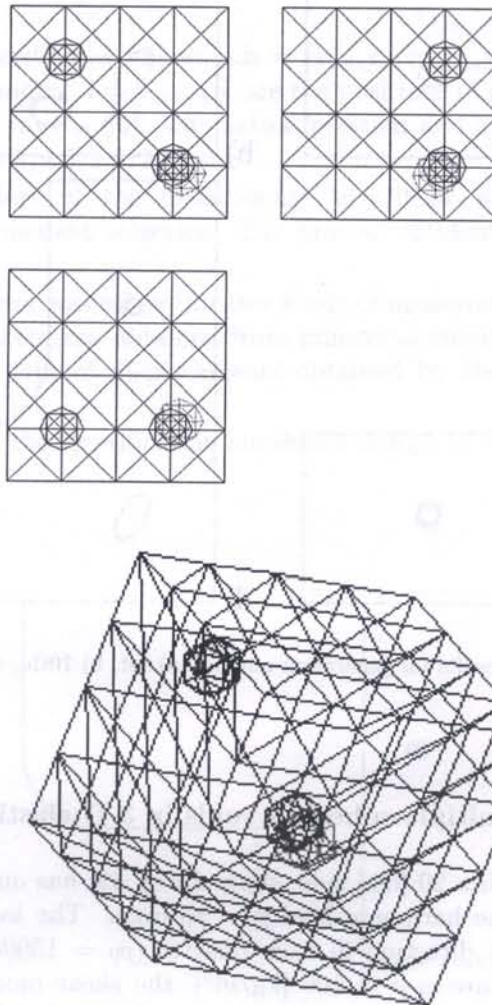


Fig. 20. Identification results of two spherical defects in the 3-D structure

**Table 4.** Parameters of two spherical voids in 3-D structure

defect parameter	actual	final
$x_1(1)$	5.00	4.99
$x_2(1)$	15.00	14.98
$x_3(1)$	15.00	14.98
$r(1)$	2.00	2.00
$x_1(2)$	15.00	15.83
$x_2(2)$	5.00	4.25
$x_3(2)$	15.00	14.22
$r(2)$	2.00	2.12

placed uniformly on four walls, carried out the identification of defects. The population contains 200 chromosomes. The result, obtained in 200 generations, is presented in Fig. 20.

## 7. THE IDENTIFICATION OF CRACKS

In the present chapter the following identification problems are examined:

- the identification of single cracks;
- the identification of multiple cracks – the number of the cracks is known;
- the identification of multiple cracks – the number of the cracks is unknown;
- the identification of cracks with the stochastic disturbance of the measured quantities.

The measurements of the displacements of the statically loaded elastic elements are considered. As the measurements are never ideal, it is proper to assume that there occurs a Gaussian measurement error (noise). One assumes that the measurement error does not exceed 10%, so the Gaussian distribution has the expected value  $E(x)$  equal to the “ideal” one and standard deviation  $\sigma(x) = E(x)/30$ .

The fitness function values for each individual in the population are obtained from the analysis of the structure by means of the dual BEM [23]. The number of the sensor points has been chosen according to the complexity of the particular task.

The floating point gene representation and the ranking selection are used. Five evolutionary operators are applied: uniform and boundary mutations, simple, arithmetical and heuristic crossovers. The selection is performed in the form of the ranking selection.

### 7.1. The identification of single crack

A structural element of the shape and dimensions presented in Fig. 21 containing a single crack of a known shape is considered. One should find the size and the position of the crack having measured displacements at 37 boundary sensor points. 6 design variables represent co-ordinates of the 1st crack tip, length of segments and slope angles of segments, respectively.

The parameters of the EA are:

- the number of variables  $N_v = 6$ ;
- the number of individuals  $N_i = 100$ ;
- the maximum number of generations  $N_g = 1000$ ;

- the probability of the uniform mutation  $p_{um} = 0.01$ ;
- the probability of the boundary mutation  $p_{bm} = 0.05$ ;
- the probability of the simple crossover  $p_{sc} = 0.1$ ;
- the probability of the arithmetic crossover  $p_{ac} = 0.1$ ;
- the probability of the heuristic crossover  $p_{hc} = 0.1$ .

The actual and final positions of the crack are shown Fig. 21. The actual and final values of variables are presented in Table 5.

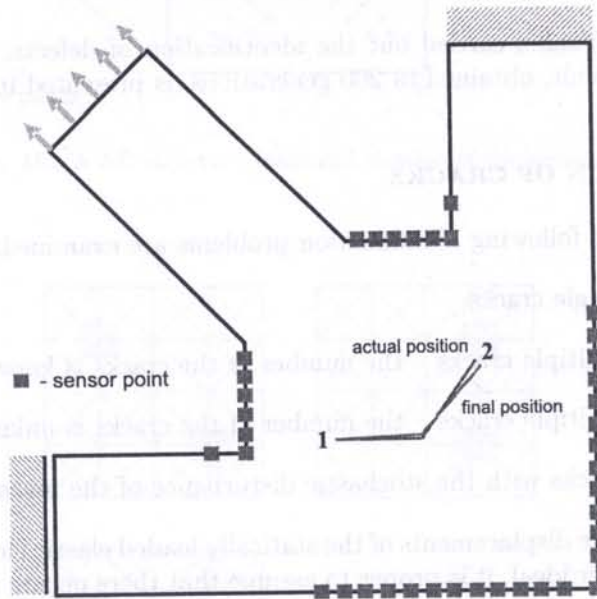


Fig. 21. The identification of one crack

Table 5. Identification of single crack

Variable number	Actual value	Range	Final value	Error [%]
1	0.00	-0.20; 0.90	0.000	-
2	-0.04	-0.20; 0.90	-0.0401	0.25
3	0.04	0.0; 0.1	-0.0413	3.25
4	0.0	-90; 90	0.0	-
5	0.058	0.0; 0.1	0.061	5.17
6	62.0	-90; 90	61.1	1.46

## 7.2. The identification of two cracks

A structural element containing two linear cracks is considered (Fig. 22). The aim is to find the positions of the cracks represented by co-ordinates of crack tips. Displacements at 81 boundary sensor points are measured.

The parameters of the EA are:

- the number of variables  $N_v = 8$ ;
- the number of individuals  $N_i = 100$ ;
- the maximum number of generations  $N_g = 3500$ ;
- the probability of the uniform mutation  $p_{um} = 0.01$ ;
- the probability of the boundary mutation  $p_{bm} = 0.05$ ;
- the probability of the simple crossover  $p_{sc} = 0.1$ ;
- the probability of the arithmetic crossover  $p_{ac} = 0.1$ ;
- the probability of the heuristic crossover  $p_{hc} = 0.1$ .

The actual and final positions of the cracks are shown in Fig. 22. The parameters of the evolutionary algorithm and the numerical results are presented in Table 6.

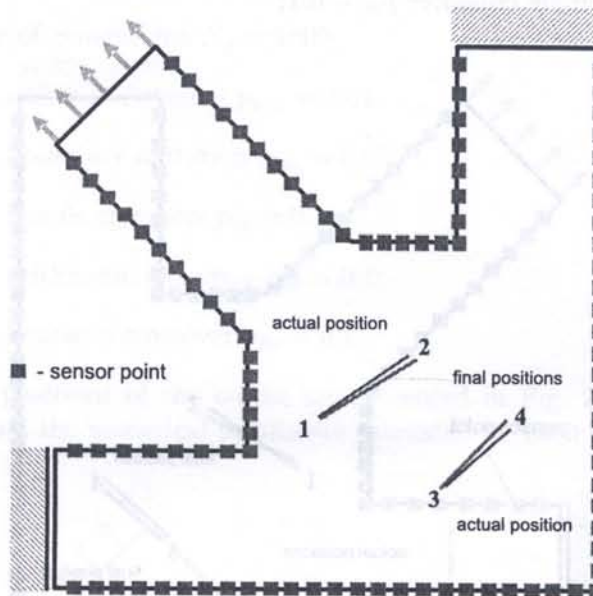


Fig. 22. The identification of two cracks

Table 6. Identification of two cracks

Variable number	Actual value	Range	Final value	Error [%]
1	0.00	-0.20; 0.90	0.0	-
2	-0.04	-0.20; 0.90	-0.041	2.5
3	0.04	-0.20; 0.90	0.04	0.0
4	-0.02	-0.20; 0.90	-0.023	15
5	0.04	-0.20; 0.90	0.044	10
6	-0.07	-0.20; 0.90	-0.062	11.4
7	0.07	-0.20; 0.90	0.069	14.2
8	-0.04	-0.20; 0.90	-0.033	17.5

### 7.3. The identification of an unknown number of cracks

It is assumed that the maximal number of linear cracks in a structure (Fig. 23) is  $n_{\max} = 5$ . The actual number of the cracks is  $n = 2$ . The number and positions of cracks should be found. The chromosome is of the type (15), the first gene represents the number of the cracks  $n \in [1, 5]$ , remaining design variables represent the co-ordinates of the crack tips. Displacements at 81 boundary sensor points are measured.

The parameters of the EA are:

- the number of variables  $N_v = 21$ ;
- the number of individuals  $N_i = 200$ ;
- the maximum number of generations  $N_g = 3500$ ;
- the probability of the uniform mutation  $p_{um} = 0.01$ ;
- the probability of the boundary mutation  $p_{bm} = 0.05$ ;
- the probability of the simple crossover  $p_{sc} = 0.1$ ;

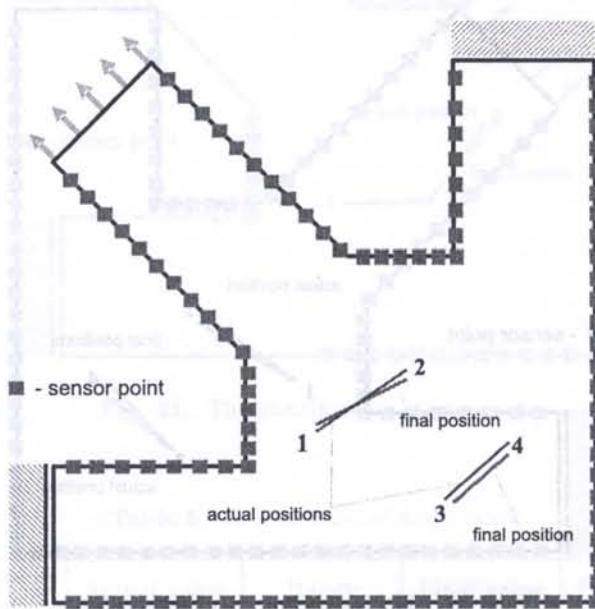


Fig. 23. The identification of 1–5 cracks

Table 7. Identification of 1–5 cracks

Variable number	Actual value	Range	Final value	Error [%]
1	2	1, ..., 5	2	0
2	0.00	-0.20; 0.90	0.01	-
3	-0.04	-0.20; 0.90	-0.041	2.5
4	0.04	-0.20; 0.90	0.04	0
5	-0.02	-0.20; 0.90	-0.019	5
6	0.04	-0.20; 0.90	0.042	5
7	-0.07	-0.20; 0.90	-0.073	4.28
8	0.07	-0.20; 0.90	0.072	2.85
9	-0.04	-0.20; 0.90	-0.043	7.5



- the probability of the arithmetic crossover  $p_{ac} = 0.1$ ;
- the probability of the heuristic crossover  $p_{hc} = 0.1$ .

The actual and final positions of the cracks are presented Fig. 23. The parameters of the evolutionary algorithm and the numerical results are presented in Table 7.

#### 7.4. The influence of the measurement errors on crack identification

The influence of the stochastic disturbance of the measured quantities on the identification of the crack is considered. The identification task becomes a non-deterministic one.

A structural element containing a linear crack is considered. The displacements are measured in 45 sensor points.

The parameters of the EA are:

- the number of variables  $N_v = 4$ ;
- the number of individuals  $N_i = 100$ ;
- the maximum number of generations  $N_g = 1000$ ;
- the probability of the uniform mutation  $p_{um} = 0.01$ ;
- the probability of the boundary mutation  $p_{bm} = 0.05$ ;
- the probability of the simple crossover  $p_{sc} = 0.1$ ;
- the probability of the arithmetic crossover  $p_{ac} = 0.1$ ;
- the probability of the heuristic crossover  $p_{hc} = 0.1$ .

The actual and final positions of the cracks are presented in Fig. 24. The parameters of the evolutionary algorithm and the numerical results are presented in Table 8.

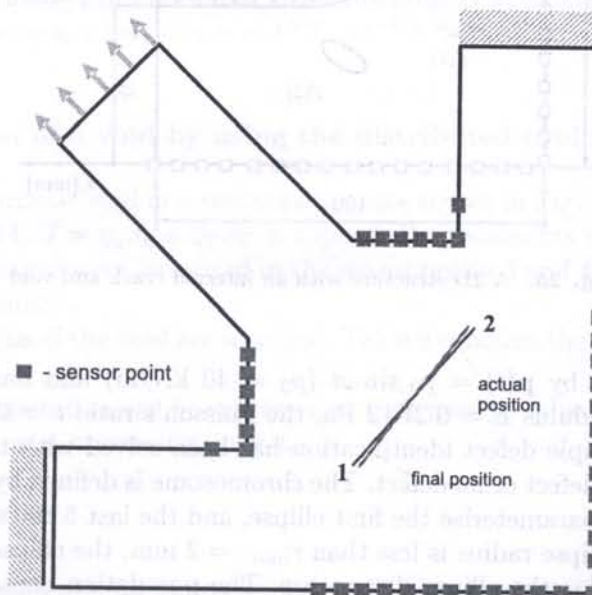


Fig. 24. The influence of the measurement error — identification 1 crack

Table 8. The influence of the measurement error

Variable number	Actual value	Range	Final value	Error [%]
1	0.05	-0.20; 0.90	0.0486	2.8
2	-0.01	-0.20; 0.90	-0.0959	4.1
3	0.01	-0.20; 0.90	0.0112	12
4	-0.05	-0.20; 0.90	-0.0518	3.6

## 8. THE IDENTIFICATION OF VOIDS AND CRACKS

A 2-D structure (Fig. 25) contains two different internal defects: a crack and an elliptical void. Both defects are parameterised by an elliptical description. The actual parameters of the elliptic void are:  $\mathbf{z}^2 = \mathbf{z}(2) = [50, 25, 5, 2.5, 2.5]$ , where the first two parameters are the co-ordinates of the ellipse centre, next ones – the two radii of the ellipse and the last one – the angle between the  $x_1$  axis and first radius. The actual crack parameters are:  $\mathbf{z}^1 = \mathbf{z}(1) = [20, 30, 5, 0, 0.25]$  and are defined in the same way as for the ellipse. The aim of the identification problem is to find the number of the defects  $n \in [0, n_{\max}]$  where  $n_{\max} = 2$ , the kind of the defects and their shape having measured displacements  $\hat{\mathbf{u}}(\mathbf{x}, t)$  in 33 sensor points (Fig. 25). The objective function  $J_{ut}$  (9) was applied.

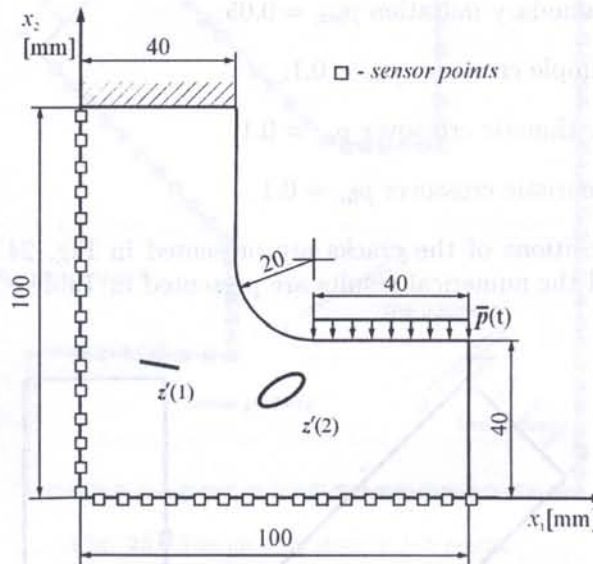


Fig. 25. A 2D structure with an internal crack and void

The structure is loaded by  $\mathbf{p}(t) = p_0 \sin \omega t$  ( $p_0 = 40$  kN/m) and has the following material properties: the Young's modulus  $E = 0.2E12$  Pa, the Poisson's ratio  $\nu = 0.3$  and the mass density  $\rho = 7800$  kg/m<sup>3</sup>. The multiple defect identification has been solved with the assumption, that the body contains: 2 defects, 1 defect or no defect. The chromosome is defined by Eq. (16) and consists of 10 genes, where the first 5 parameterise the first ellipse, and the last 5 the second ellipse. When one of the genes, which is an ellipse radius is less than  $r_{\min} = 2$  mm, the ellipse becomes a crack, when both radii are less than  $r_{\min}$  the ellipse disappears. The population contains 2000 chromosomes. The tournament method of selection was used. The solution was obtained for the case with no noise in 100 generations and for noisy data in 120 generation. Figure 26 presents the best solution of the first and the last generation.

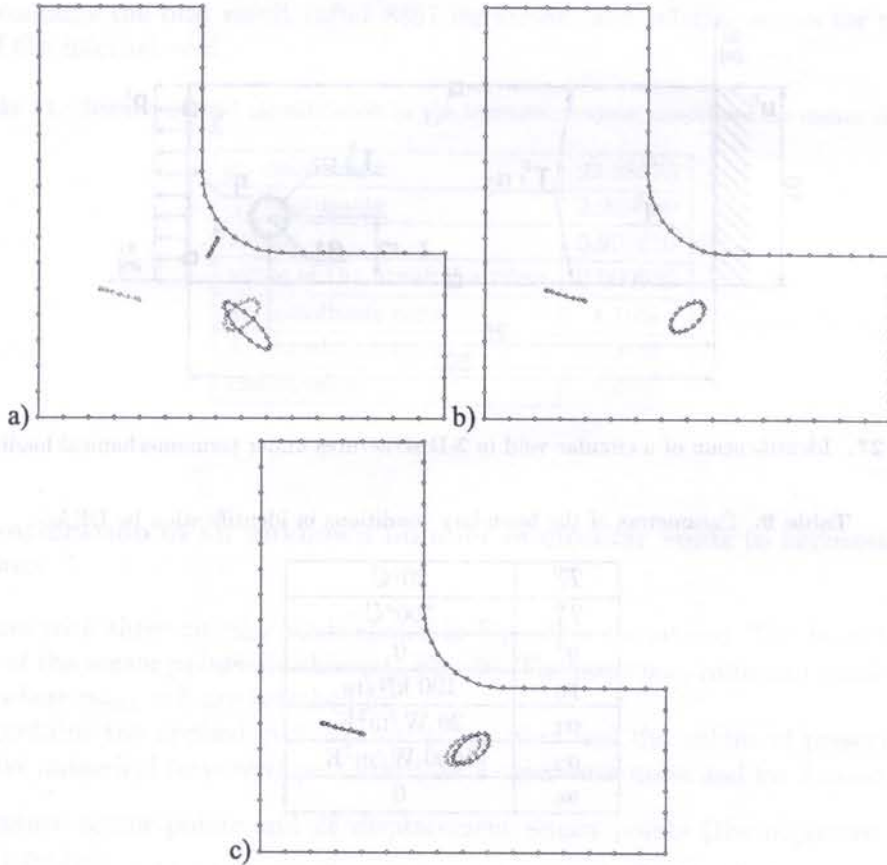


Fig. 26. Identification results: a) 1st generation, b) 100th generation, c) 120th generation (noisy data)

## 9. THE IDENTIFICATION IN THE TERMOMECHANICAL CONDITIONS

Two examples of the identification of voids in 2D structures (plain strain) are considered in this chapter. Structures are made from material: shear modulus  $G = 80$  GPa, Poisson's ratio  $\nu = 0.23$  and the coefficient of thermal expansion  $\alpha = 12.5 \cdot 10^{-6}$   $1/^\circ\text{C}$ .

### 9.1. The identification of a void by using the distributed evolutionary algorithm

The identification of a circular void in a rectangular plate shown in Fig. 27 is considered. The fitness function given by Eq. (14)  $J = \eta_u J_u + \eta_T J_T$  is applied. Displacements were measured in the sensor points 1 and 2. Temperatures were measured in the sensor points 3 and 4. The boundary was divided into 48 boundary elements.

The position and radius of the void are searched. Table 9 contains the parameters of the boundary conditions.

The speedup  $s$  of computation can be expressed as time need to solve the problem on 1 processing unit  $t_1$  divided by time on  $n$ -processing units  $t_n$ ,

$$s = \frac{t_1}{t_n}. \quad (27)$$

The number of processing units varies from 1 to 3. Two computers with two SMP processing units are used. Table 10 contains the parameters of the distributed evolutionary algorithm.

The speedup of the improved distributed evolutionary algorithm is shown in Fig. 28. The linear speedup is the theoretical maximal speedup of the parallel evolutionary algorithm.

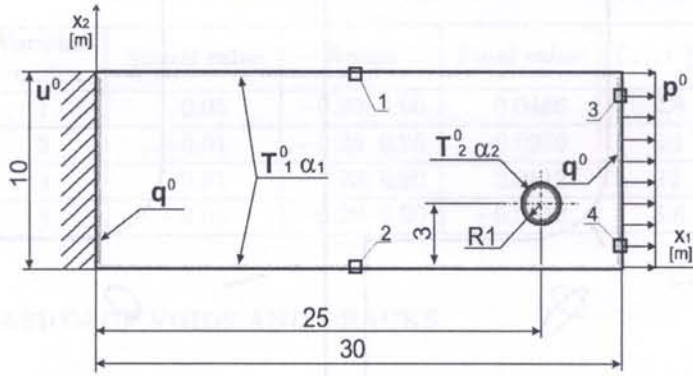


Fig. 27. Identification of a circular void in 2-D structures under termomechanical loadings

Table 9. Parameters of the boundary conditions in identification by DEA

$T_1^0$	20°C
$T_2^0$	500°C
$q^0$	0
$p_0$	100 kN/m
$\alpha_1$	20 W/m <sup>2</sup> K
$\alpha_2$	1000 W/m <sup>2</sup> K
$u_0$	0

Table 10. Parameters of distributed evolutionary algorithm

Number of subpopulations	2
Number of chromosomes	10
Number of genes	3
Constrain on gene 1 ( $x_1$ coordinate)	0.5 ÷ 29.5
Constrain on gene 2 ( $x_2$ coordinate)	0.5 ÷ 9.5
Constrain on gene 3 (radius)	0.5 ÷ 3.0

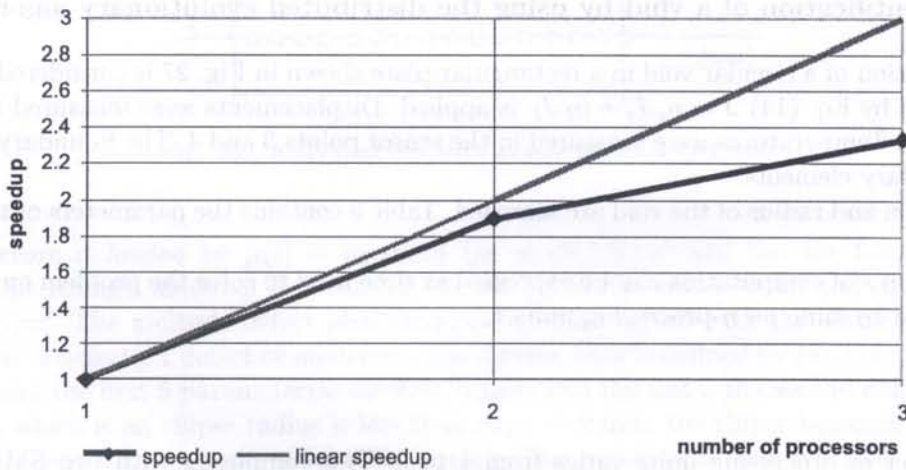


Fig. 28. The speedup of improved distributed evolutionary algorithm

Table 11 contains the best result (after 8897 iterations) and relative errors for the coordinates and radius of the internal void.

**Table 11.** Results of void identification in the termomechanical conditions by means of DEA

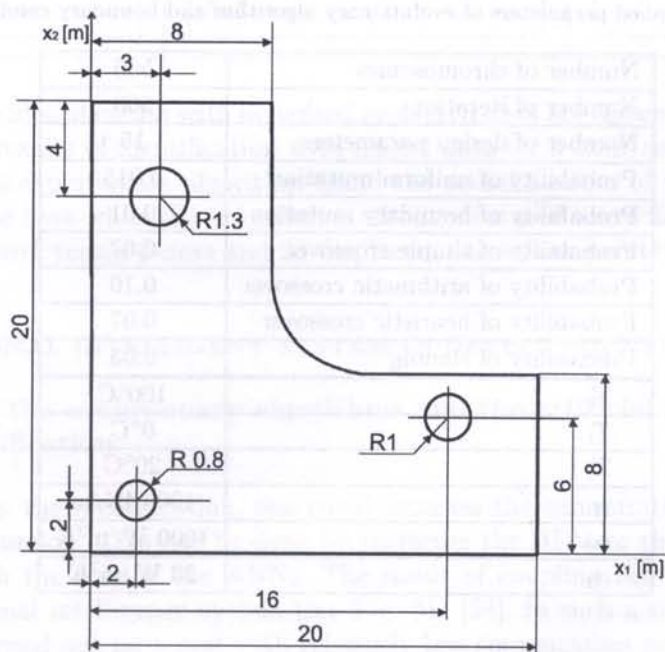
$x_1$ coordinate	25.28978
$x_2$ coordinate	2.989090
radius	0.997610
value of the fitness function	0.000030
$x_1$ coordinate error	1,16%
$x_2$ coordinate error	0,36%
radius error	0,24%

## 9.2. The identification of an unknown number of circular voids in termomechanical conditions

A 2-D structure with three circular voids shown in Fig. 29 is considered. The boundary conditions and positions of the sensor points are shown in Fig. 30. The positions, radii and number of the voids  $n \in [0, n_{\max}]$  where  $n_{\max} = 5$  are searched for.

Table 12 contains the applied evolutionary parameters and the values of prescribed boundary conditions. Five numerical tests were performed for 3 cases with noise and for 3 cases with no noise:

- 30 temperature sensor points and 28 displacement sensor points (the objective function (14),  $J = \eta_u J_u + \eta_T J_T$ ),
- 58 temperature sensors – located in every sensor point (the objective function (8),  $J_T$ ),
- 58 displacement sensors – located in every sensor point (the objective function (7),  $J_u$ ).



**Fig. 29.** A 2-D structure with three circular voids under termomechanical conditions

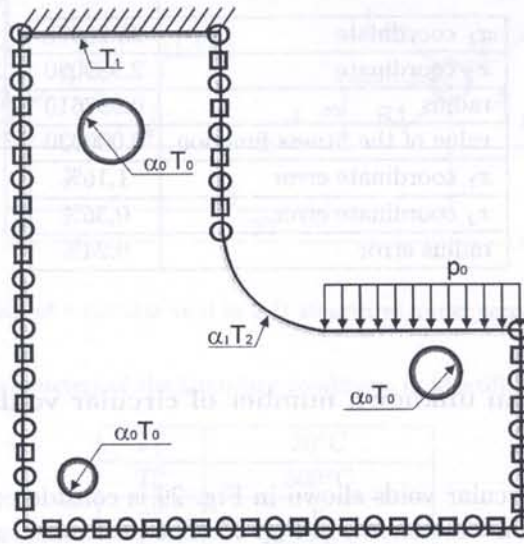
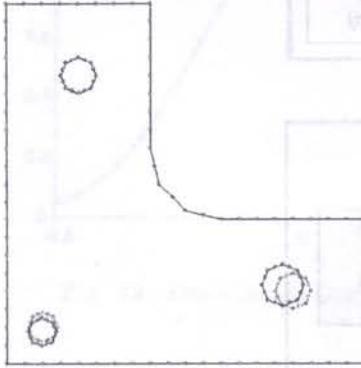
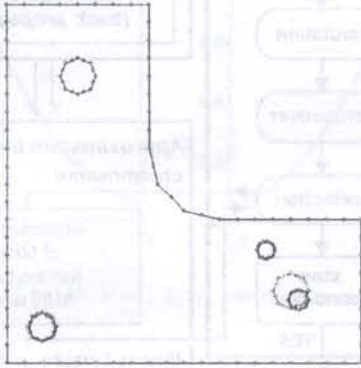
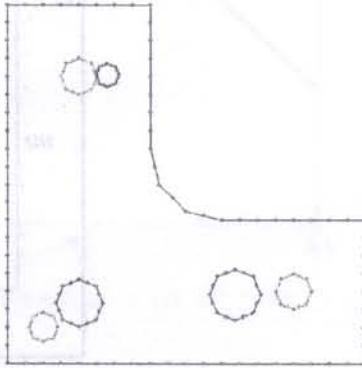


Fig. 30. Boundary conditions and location of the sensor points for the structure

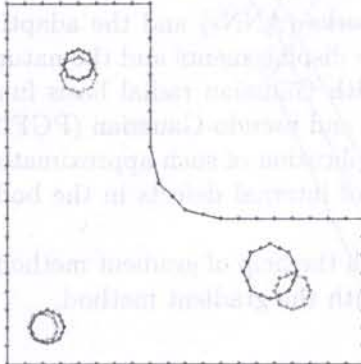
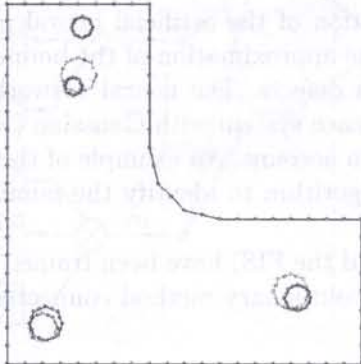
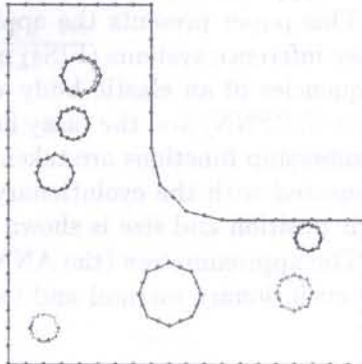
Table 12. Applied parameters of evolutionary algorithm and boundary conditions values

Number of chromosomes	500
Number of iterations	300
Number of design parameters	15
Probability of uniform mutation	0.015
Probability of boundary mutation	0.01
Probability of simple crossover	0.07
Probability of arithmetic crossover	0.10
Probability of heuristic crossover	0.07
Probability of cloning	0.03
$T_0$	100°C
$T_1$	0°C
$T_2$	20°C
$p_0$	100 MN/m
$\alpha_0$	1000 W/m <sup>2</sup> K
$\alpha_1$	20 W/m <sup>2</sup> K

**Table 13.** Results of numerical tests with no noise

30 temperature sensor points and 28 displacement sensor points	58 temperature sensor points	58 displacement sensor points
		

**Table 14.** Results of numerical tests with no noise

30 temperature sensor points and 28 displacement sensor points	58 temperature sensor points	58 displacement sensor points
		

The results of the identification with no noised measured data are shown in Table 13, and Table 14 contains numerical results of identification with noised data for 3 cases mentioned above.

It is seen that the evolutionary algorithm found the actual number of voids and the best results were obtained in the case when the information about measured state fields had mixed character (i.e. for 30 temperature sensor points and 28 displacements sensor points).

## 10. COMPUTATIONAL INTELLIGENT SYSTEM IN DEFECT IDENTIFICATION

### 10.1. Coupling of the evolutionary algorithms and the artificial neural network in defect identification

In order to speed up the identification, one could increase the computation power or improve the fitness function evaluation. This can be done by replacing the BEM or the FEM solutions by their approximations with the help of the ANNs. The result of coupling the EA and the ANN or the FIS is a computational intelligence system (see Fig. 31) [24]. In such a situation the identification process could be carried out on a unit with relatively low computation power in a very short time. It can be said that the artificial neural network or the fuzzy inference system is an approximator of a boundary-value problem for the different number, shapes and positions of defects. The EA will

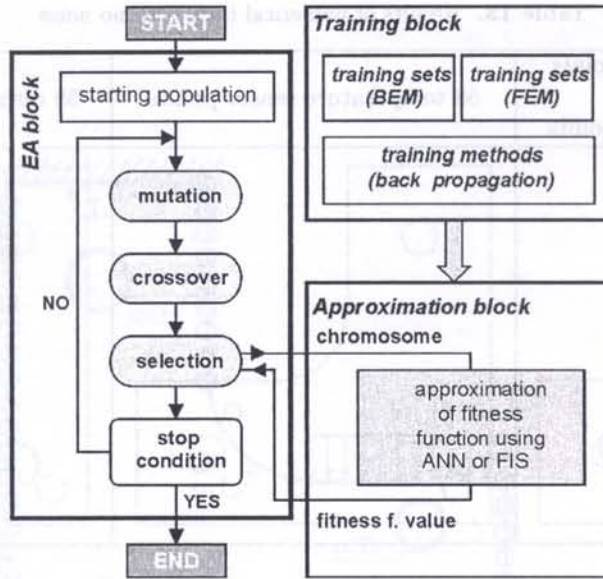


Fig. 31. Computational intelligence system for defect identification

find the number, shapes and positions of internal defects basing on the results obtained by means of the approximators.

This paper presents the application of the artificial neural networks (ANNs) and the adaptive fuzzy inference systems (FISs) in the approximation of the boundary displacements and the natural frequencies of an elastic body with defects. The neural network with Gaussian radial basis functions (RBFNN) and the fuzzy inference system with Gaussian (FIS) and pseudo-Gaussian (PGFIS) membership functions are taken into account. An example of the application of such approximators connected with the evolutionary algorithm to identify the number of internal defects in the body, their position and size is shown.

The approximators (the ANN and the FIS) have been trained with the help of gradient methods, the evolutionary method and the evolutionary method connected with the gradient method.

## 10.2. The adaptive fuzzy inference system

Two kinds of the adaptive fuzzy inference system were taken into consideration. One is the fuzzy inference system with the Gaussian type of membership function, another one is the fuzzy inference system with pseudo-Gaussian membership functions [17].

The formula (28) shows the Gaussian function while the formula (29) presents the pseudo-Gaussian function,

$$G(x) = \exp\left(-\frac{1}{2} \frac{(x-t)^2}{\sigma^2}\right), \quad (28)$$

$$G(x) = \exp\left(-\frac{1}{2} \frac{(x-t)^2}{\sigma_L^2}\right) U(x; -\infty, t) + \exp\left(-\frac{1}{2} \frac{(x-t)^2}{\sigma_P^2}\right) U(x; t, \infty), \quad (29)$$

where

$$U(x; a, b) = \begin{cases} 1 & \text{if } a \leq x < b, \\ 0 & \text{otherwise,} \end{cases}$$

$t$  is the mean,  $\sigma$  is the standard deviation, and  $\sigma_L$  and  $\sigma_P$  are left-hand and right-hand standard deviations, respectively. Figure 32 shows the Gaussian and pseudo-Gaussian functions.



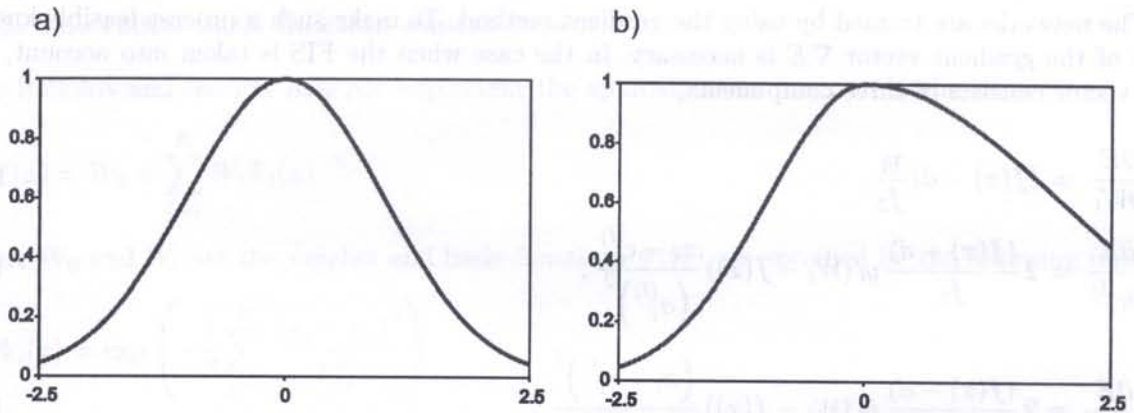


Fig. 32. Functions: a) Gaussian  $t = 0.0, s = 1.0$ ; b) pseudo-Gaussian  $t = 0.0, sL = 1.0, sP = 2.0$ ;

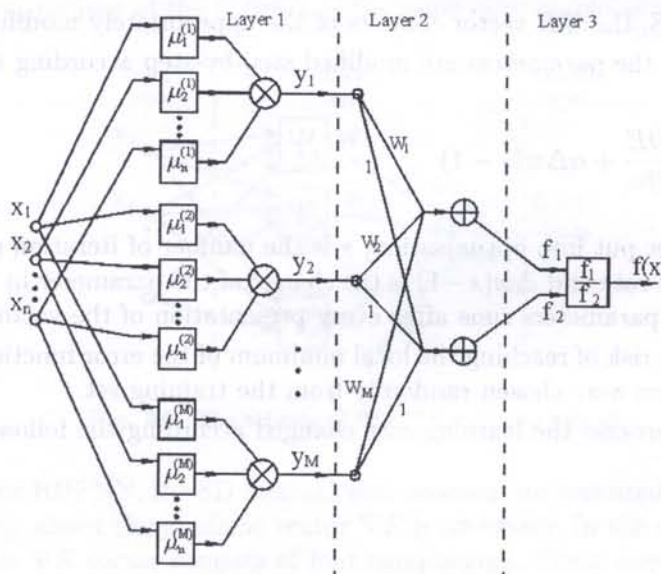


Fig. 33. The scheme of the FIS and PGFIS with one output

The approximation function realized by FIS and PGFIS is as follows,

$$f(x) = \frac{\sum_{l=1}^M W_l \prod_{i=1}^N G(x_i)}{\sum_{l=1}^M \prod_{i=1}^N G(x_i)} \tag{30}$$

where  $G(x)$  is given by Eq. (28) in the case, when the FIS is considered, and by Eq. (29) when the PGFIS is taken into account,  $\mathbf{x}$  is the input vector,  $N$  is the number of inputs,  $M$  is the number of fuzzy rules.

The scheme of the FIS and the PGFIS is presented in Fig. 33.

Training the network consists of changes made in the parameters  $W, t, \sigma, \sigma_L, \sigma_P$  step-by-step in order to minimize an error given by

$$E = \frac{1}{2} (y^p - d^p)^2 \tag{31}$$

where  $p = 1 \dots P$  is the number of training patterns,  $y^p$  is the response of the ANN on the  $p$  input vector,  $d^p$  is the desirable response of the ANN.

The networks are trained by using the gradient method. To make such a process feasible knowledge of the gradient vector  $\nabla E$  is necessary. In the case when the FIS is taken into account, the  $\nabla E$  vector consists of three components,

$$\frac{\partial E}{\partial W_l} = (f(x) - d) \frac{y_l}{f_2}, \quad (32)$$

$$\frac{\partial E}{\partial c_i^{(l)}} = 2 \frac{(f(x) - d)}{f_2} y_l (W_l - f(x)) \frac{x_i - c_i^{(l)}}{(\sigma_i^{(l)})^2}, \quad (33)$$

$$\frac{\partial E}{\partial \sigma_i^{(l)}} = 2 \frac{(f(x) - d)}{f_2} y_l (W_l - f(x)) \frac{(x_i - c_i^{(l)})^2}{(\sigma_i^{(l)})^3}. \quad (34)$$

In the case of the PGFIS, the  $\nabla E$  vector consists of the appropriately modified four components.

As it was mentioned, the parameters are modified step-by-step according to the formula

$$w(s+1) = w(s) - \eta \frac{\partial E}{\partial w} + \alpha \Delta w(s-1) \quad (35)$$

where  $w$  is the parameter put into optimisation,  $s$  is the number of iteration step,  $\eta$  is the learning rate,  $\alpha$  is the momentum rate and  $\Delta w(s-1)$  is the change of the parameter in the previous iteration step. The change of the parameters runs after every presentation of the vector  $\mathbf{x}^p$  to the ANN.

In order to reduce the risk of reaching the local minimum of the error function during the training process, the training pairs were chosen randomly from the training set.

During the training process the learning rate changed according the following formula,

$$\eta(s) = \frac{\eta_0}{1 + \frac{s}{T}}, \quad (36)$$

where  $s$  is the iteration step,  $\eta_0$  is the initial value of the learning rate,  $T$  is a constant.

Initial values of parameters  $W$ ,  $c$  and  $\sigma(\sigma_L, \sigma_P)$  were selected arbitrarily.

After choosing starting parameters, the gradient learning methods, notably the steepest descent method (SD) and the conjugate gradient method (CG), were applied to find the optimal set of parameters of the networks [16]. More details about the training process of the FIS are presented in [11]. The training procedure of the PGFIS proceeds similarly to that of the FIS.

During the training process the EA was applied, but it did not yield satisfactory results – the training and testing errors were too significant. Better results were obtained when the evolutionary method connected with the gradient methods were applied. In the first step of the training process the EA finds the parameter vectors of the neural network. In the second step these vectors are used as starting parameters in the gradient method (SD or CG). Such a connection gives better results than the exclusive use of the EA method; however, the final conclusion is that the training and testing errors are not smaller than those produced when only the gradient methods are used. The second disadvantage is that the training time needed to reach a certain error value is not shorter than in the case when only the gradient method is used. Due to a large number of variables needed to be optimised during the learning process, the chromosome is very large. Due to the great number of genes in the chromosome the number of chromosomes in population is also big. This leads to a vast increase of the computation time. It is worth noting that the results obtained in any test with the EA or the EA connected with the gradient learning method were not better (smaller training and testing errors) than when solely the gradient training method was used.

### 10.3. The radial basis function neural network

The RBFNN and the PG-RBFNN implement the approximation function in the form

$$f(x) = W_0 + \sum_{i=1}^K W_i \Psi_i(x) \tag{37}$$

where  $W_0$  and  $W_i$  are the weights and basis functions  $\Psi_i(x)$  are specified by the following formula

$$\Psi_i(x) = \exp \left( -\frac{1}{2} \sum_{j=1}^N \frac{(x_j - t_{ij})^2}{\sigma_{ij}^2} \right). \tag{38}$$

$N$  means the number of inputs of the neural network.

Figure 34 presents the structure of the RBFNN with  $N$  input nodes, one output and  $K$  basis functions. The training process of the network is the same in principle as that of the FIS, Eq. (31).

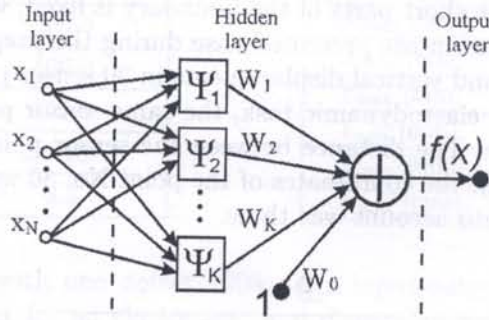


Fig. 34. The scheme of RBFNN with one output

In order to train the RBFNN, the SD method with momentum was used. To make such a process feasible, the knowledge about the gradient vector  $\nabla E$  is necessary. In the case when the RBFNN is taken into account the  $\nabla E$  vector consists of four components. These components have the form

$$\frac{\partial E}{\partial W_0} = (y^p - d^p), \tag{39}$$

$$\frac{\partial E}{\partial W_j} = (y^p - d^p) \exp \left[ -\frac{1}{2} \sum_{z=1}^N \left( \frac{x_z - t_z^{(i)}}{\sigma_z^{(i)}} \right)^2 \right], \tag{40}$$

$$\frac{\partial E}{\partial t_k^{(i)}} = (y^p - d^p) W_i \exp \left[ -\frac{1}{2} \sum_{z=1}^N \left( \frac{x_z - t_z^{(i)}}{\sigma_z^{(i)}} \right)^2 \right] \frac{x_k - t_k^{(i)}}{(\sigma_k^{(i)})^2}, \tag{41}$$

$$\frac{\partial E}{\partial \sigma_k^{(i)}} = (y^p - d^p) W_i \exp \left[ -\frac{1}{2} \sum_{z=1}^N \left( \frac{x_z - t_z^{(i)}}{\sigma_z^{(i)}} \right)^2 \right] \frac{(x_k - t_k^{(i)})}{(\sigma_k^{(i)})^3}, \tag{42}$$

where  $N$  is the number of inputs and  $K$  is the number of basis functions.

In order to reduce the risk of reaching the local minimum of the error function, during the training process the simulated annealing and the “jog of weight” technique [10] were used. The “jog of weight” technique consists of adding a random value to the parameter in order to restart the process, in the time when there is a suspicion that the minimization process stopped in the local minimum.

During the training process the learning rate changed according to the formula (35).

The initial values of parameters  $W$  and  $\sigma$  were selected arbitrarily. Initial values of parameters  $\mathbf{t}$  were chosen according to the algorithm [13]:

- divide the training set into  $K$  subsets ( $K$  is the number of basis functions)
- find input vector  $\mathbf{x}_{\max}^p$  for which the output value is maximum in every subset
- take  $\mathbf{t}^{(i)} = \mathbf{x}_{\max}^p$

After choosing starting parameters the gradient learning method was applied to find the optimal set of parameters of the networks.

#### 10.4. The formulation of an approximation problem

An elastic body with a number of internal defects is considered (Fig. 35). In the presented case a plate contains one or two defects in the form of a circular hole. The upper part of the boundary is loaded by traction  $p$ , one of the short parts of the boundary is fixed, while the last two parts of the boundary contain sensor points. In the presented case during the preparation of the training data, information about horizontal and vertical displacements in 30 sensor points for the elastostatic task was taken. When running the elastodynamic task, the same sensor points are used, however, over a timeframe of 101 time steps. The distance between the sensor points equals 10 mm. The point No. 1 has coordinates  $(0.0, 0.0)$ , the coordinates of the point No. 30 were  $(200.0, 90.0)$ . The number of natural frequencies taken into account was three.

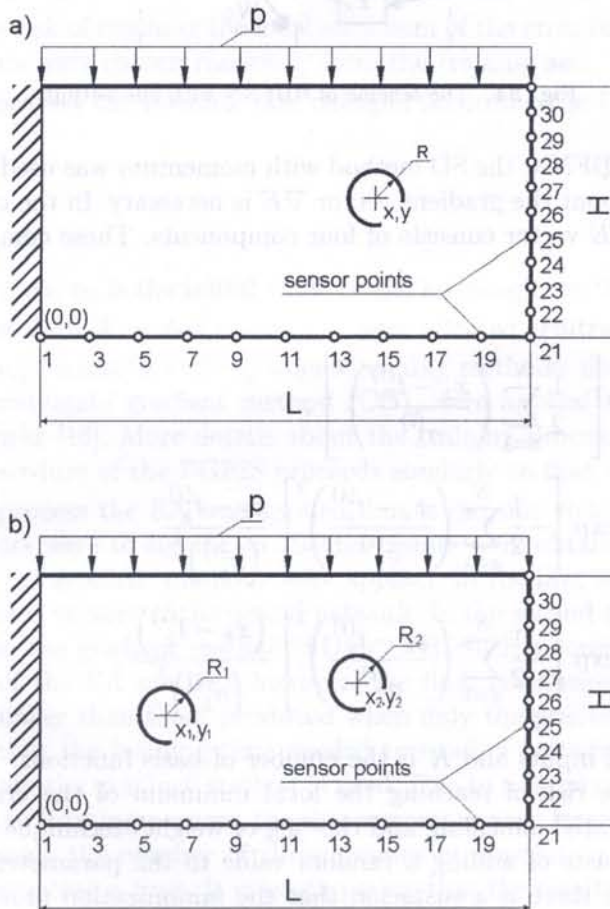


Fig. 35. An elastic plate with a) one; b) two internal defects in the form of a circular hole

The ANN or the FIS should approximate the boundary displacements in some sensor points on the boundary of the structure. Another ANN (FIS) should approximate the natural frequencies of the plate with the defect.

The positions and radii of defects needs to be inputted to the approximator. On the outputs the results are horizontal or vertical displacement in some sensor points on the boundary of the body, or the natural frequency of the body with the defect.

The training and testing data sets used during the training and testing process were obtained by making use of the boundary element method (BEM) [4].

Table 15 shows the material and geometrical parameters of the plate taken into consideration. The maximum and minimum values of the centres coordinates of the defects, and the range of the variability of radius - the maximum and minimum radius of one defect are shown in Table 16.

**Table 15.** Geometrical and material parameters of a plate

Parameter	Value
$L$ [mm]	200.0
$H$ [mm]	100.0
$p$ [MPa]	100.0
$E$ [MPa]	$2.0 \cdot 10^5$
$\nu$	0.3

**Table 16.** The range of coordinates and radius of the defect

	defect 1	defect 2
$X_{\max}$ [mm]	170.0	170.0
$X_{\min}$ [mm]	30.0	30.0
$Y_{\max}$ [mm]	75.0	75.0
$Y_{\min}$ [mm]	25.0	25.0
$R_{\max}$ [mm]	14.0	14.0
$R_{\min}$ [mm]	5.0	5.0

In the case of the plate with one defect 1008 pairs input-output created a training set and 300 pairs formed a testing set for an elastostatic and eigenvalue problem, and 3135 and 1890 for elastodynamic. In order to train the ANN (or the FIS) to approximate the boundary displacements and natural frequencies for the plate with two defects, the training set consists of 12800 pairs and the testing sets consists of 1600 pairs for the elastostatic and eigenvalue problem. When the elastodynamic problem is taken into consideration the training and testing sets consists of 3160 and 3060 pairs, respectively.

The multi input-single output (MISO) neural networks and fuzzy inference systems were chosen [13, 14]. One should prepare and train three approximators to approximate each of three natural frequencies of the plate, and several ANNs (or fuzzy inference systems) to approximate displacements in only one direction (vertical or horizontal) in one sensor point on the boundary of the structure.

The assessment of the approximation abilities of the ANN for the training set was done by the learning error according to the formula

$$MSE_L = \frac{1}{2P} \sum_{p=1}^P (y^p - d^p)^2 \quad (43)$$

where  $p = 1, \dots, P$  is the number of input-output pairs of the training set,  $y^p$  is the value of the output of the ANN,  $d^p$  is the desirable output of the network.

In order to estimate the approximation abilities for the testing set, the testing error was calculated according to the formula

$$MSE_T = \frac{1}{2T} \sum_{p=1}^T (y^p - d^p)^2 \quad (44)$$

where  $p = 1, \dots, T$  is the number of input-output pairs of the testing set,  $y^p$  is the answer of the ANN for the  $p$  input vector,  $d^p$  is the desirable output.

Below there are presented the results of training and testing the FIS, the PGFIS and the RBFNN when the second natural frequency and vertical displacements in elastostatic task in the point No. 25, which coordinates are (200.0, 40.0), were approximated. Table 17 shows the training and testing error given by Eqs. (43) and (44) for the plate with one and two defects.

Figure 36 illustrates a sample diagram of the change of learning and testing error during the RBFNN and the FIS training process. The approximators were trained to approximate the vertical displacements in the sensor point No. 27, which coordinates are (200.0, 60.0). The learning rate  $\eta$  changed as it is shown in formula (36), the starting value of the learning rate was  $\eta_0 = 0.01$  and  $\eta_0 = 0.6$ , the time rate  $T = 1500$  and  $T = 15000$ , the momentum rate equals  $\alpha = 0.008$  and

Table 17. The results of approximation for different approximators

Second eigenfrequency						
	FIS		PGFIS		RBFNN	
No. of defects	1	2	1	2	1	2
$MSE_L$	2.6e-6	2.2e-4	1.9e-4	4.41e-4	5.0e-6	7.31e-4
$MSE_T$	5.4e-6	5.2e-4	2.0e-4	9.08e-4	1.3e-5	1.03e-3
No. of epochs	100 000	500	100 000	500	100 000	500
Displacement						
	FIS		PGFIS		RBFNN	
No. of defects	1	2	1	2	1	2
$MSE_L$	1.56e-6	4.41e-6	6.77e-4	1.32e-3	3.92e-6	1.04e-5
$MSE_T$	3.14e-6	9.08e-6	8.13e-4	2.01e-3	7.02e-6	3.80e-5
No. of epochs	100 000	500	100 000	500	100 000	3000

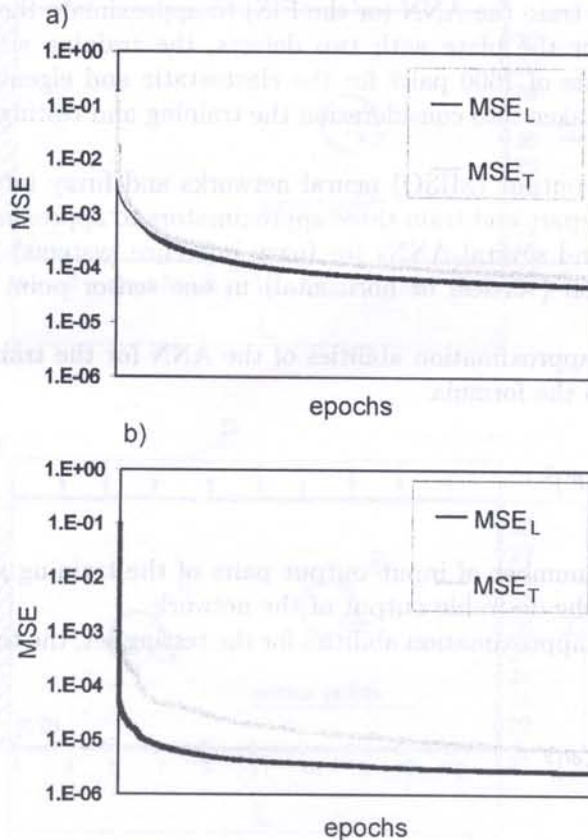


Fig. 36. Training and testing error of a) RBFNN and b) FIS for displacement approximation

$\alpha = 0.06$  for the RBFNN and the FIS, respectively. The number of iterations is 30 000 for the RBFNN and 20 000 in the case when the FIS is taken into consideration. The number of basis functions is  $K = 85$ , the number of fuzzy rules used in this ANN is  $M = 15$ .

As it is shown in Fig. 36 the FIS reaches a much smaller level of training and testing error than the RBFNN in a smaller number of iteration steps.

Initial tests show that the FIS has the best abilities of the approximation of the natural frequencies and boundary displacements for elastostatic in some sensor points among the three tested approximators. Due to these abilities it was decided the FIS will be trained to approximate the solution of the boundary-value problem in the elastostatic and eigenvalue tasks. In the case of the PGFIS it was possible to reach a very small error of the training set but simultaneously the testing error was bigger than that for FIS.

Table 18 presents the final FIS errors of training and testing for approximation of several tasks, the architecture of the network (the number of fuzzy rules  $M$ ) and the number of iteration steps in which the error has been reached.

In the case when displacements in the elastodynamic problem are taken into account the training and testing errors of the FIS and the PGFIS are shown in Table 19.

**Table 18.** Training and testing error of the FIS for several approximation tasks

	The plate with one defect				The plate with two defects			
	$MSE_L$	$MSE_T$	$M$	epochs	$MSE_L$	$MSE_T$	$M$	epochs
first eigenfrequency	6.2e-7	8.3e-7	45	2.0e5	8.2e-5	1.7e-4	9	0.6e3
Second eigenfrequency	2.2e-7	1.1e-6	85	6.0e5	2.2e-4	5.2e-4	6	4.0e3
third eigenfrequency	8.9e-7	2.6e-6	85	1.0e5	4.7e-5	2.4e-4	9	3.0e3
vert. displ. at point (200,20)	9.6e-7	2.0e-6	15	2.0e5	3.7e-6	4.9e-6	40	5.5e3
vert. displ. at point (200,40)	1.7e-6	2.7e-6	15	2.0e5	7.6e-6	7.8e-6	24	3.3e3
vert. displ. at point (200,60)	1.6e-6	3.0e-6	15	2.4e5	2.8e-6	4.1e-6	40	2.0e3

**Table 19.** Training and testing errors of the FIS and PGFIS for vertical displacement approximation in the sensor point No. 10 in the time step 75

	The plate with one defect				The plate with two defects			
	$MSE_L$	$MSE_T$	$M$	epochs	$MSE_L$	$MSE_T$	$M$	epochs
FIS	1.3e-4	1.4e-4	198	190	4.7e-5	1.4e-3	56	338
PGFIS	5.3e-5	6.9e-5	198	473	1.3e-4	4.9e-4	56	228

**Table 20.** Training and testing error of the PGFIS

No. of point	time step	The plate with one defect				The plate with two defects			
		$MSE_L$	$MSE_T$	$M$	epochs	$MSE_L$	$MSE_T$	$M$	epochs
7(v)	62	3.8e-5	4.6e-5	64	734	4.8e-5	6.6e-4	64	344
7(v)	75	8.1e-5	8.2e-5	64	562	2.7e-5	8.9e-4	96	496
10(v)	75	5.3e-5	6.9e-5	198	473	1.3e-4	4.9e-4	56	228
10(v)	98	8.5e-5	1.3e-4	216	887	1.9e-4	6.7e-4	196	230
25(h)	41	1.2e-4	1.3e-4	32	366	2.3e-5	4.8e-4	64	371
25(h)	74	4.1e-5	5.2e-5	64	303	2.5e-4	6.3e-4	64	207
25(v)	74	3.4e-5	6.3e-5	68	236	2.9e-5	6.2e-4	56	270

Tests show that in the case of the elastodynamic problem the PGFIS has better approximation abilities than the FIS. Due to this, it was decided that the PGFIS will be trained to be the approximator of the solution of the boundary value problem in elastodynamic tasks.

Table 20 presents the final PGFIS errors of training and testing for the approximation of vertical ( $v$ ) and horizontal ( $h$ ) displacements in an elastodynamic problem, the architecture of the network (the number of fuzzy rules  $M$ ) and the number of iteration step in which the error has been reached.

The possibility of using only the evolutionary algorithm and the EA coupled with the gradient learning method during the training process was also studied. Figure 37a shows the change of training error of the FIS during the first stage of training with the use of EA. Figure 37b illustrates the diagram of the change of the training error in the second stage, when the FIS was trained by using the SD method for starting parameters from the EA stage.

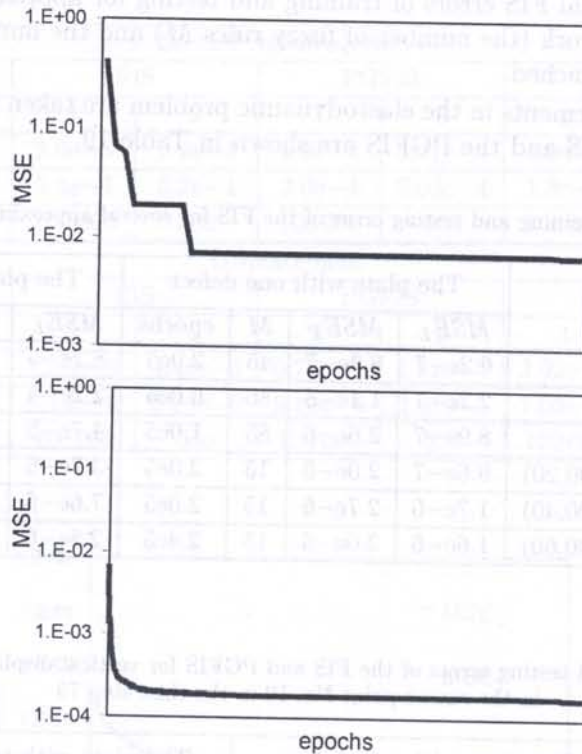


Fig. 37. The diagram of the FIS training error during a) the EA learning process; b) SD learning process (starting parameters from EA)

The FIS with  $M = 85$  fuzzy rules is trained to approximate the third natural frequency for the plate with two defects. The parameters of the EA and the SD are presented below:

- the number of genes: 256,
- the number of chromosomes: 50,
- the number of epochs in the EA: 50,
- the probability of the arithmetic crossover: 0.6,
- the kind of mutation: Gaussian mutation,
- the number of iterations in SD: 1000,
- learning rate  $\eta_0 = 0.2$ ,
- momentum rate  $\alpha = 0.4$ .



It can be noticed that in a few iteration steps the EA finds the set of parameters which allows reaching a small value of errors given by formulas (43) and (44). Unfortunately, the gradient process after the EA stage proceeds very slowly. Such a behaviour was typical for this kind of training method.

As it was mentioned above, the input-output pairs were obtained by using the BEM and the FEM. The approximator for the natural frequency or boundary displacements approximation for the body with one defect has three input nodes and one output. The first input is the  $x$ -coordinate of the centre of the defect, the second input is the  $y$ -coordinate of this centre and the third input node needs the value of the radius of the defect. When the plate with two internal defects is taken into consideration, the approximator has six inputs – three for the first defect and another three for second one.

The input values were from the range [30.0,170.0] for the  $x$ -coordinate, [25.0,75.0] for the  $y$ -coordinate and [5.0,14.0] for the radius. The output values of the output training vectors were put into the range [0.1,0.9].

### 10.5. The identification problem

A 2-D elastic body with  $D \leq D_{max}$  internal defects in the form of circular holes is considered.  $D_{max}$  means the maximum number of defects which can be expected in the body. The number of defects, their position and size are not known.

The EA should identify the actual number of defects  $D$  and their parameters on the basis of the knowledge about  $F$  natural frequencies of the body with defect and displacements in  $S$  sensor points on the boundary of the body. The unknown parameters of the defect are coordinates of the hole's centre ( $X_z, Y_z$ ) and its size  $R_z$  ( $z = 1, 2, \dots, D$ ). Defects are specified by a chromosome

$$ch = [X_1, Y_1, R_1, \dots, X_z, Y_z, R_z, \dots, X_{D_{max}}, Y_{D_{max}}, R_{D_{max}}] \tag{45}$$

where:  $X_z, Y_z$  are the coordinates of the centre of defect and  $R_z$  is the radius of the defect.  $X_z, Y_z$  and  $R_z$  play the role of genes. The EA sends the chromosome with the suggested values of the position and radii of each defect to the approximation block.

In the case when  $R_i < R_{min}$  the program assumes that genes  $X_i, Y_i, R_i$  are inactive genes

$$R_i = 0 \quad \forall (R_i < R_{min}). \tag{46}$$

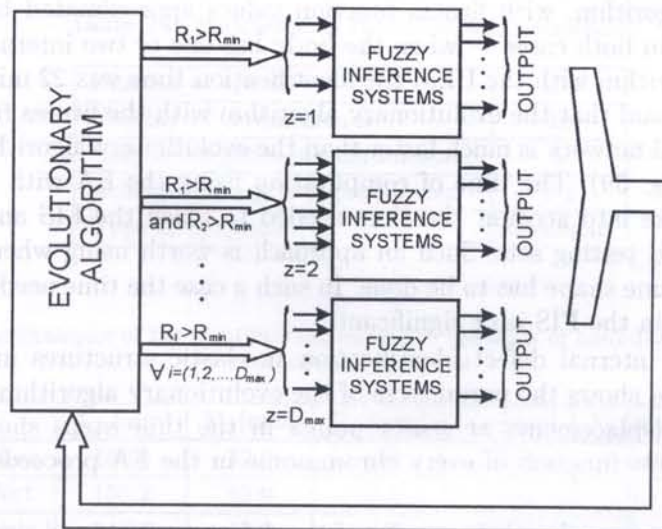


Fig. 38. The approximation of displacements and natural frequencies for one, two or more internal defects

Genes with information about the position and shape of defects are sent to inputs of the approximators. Approximated displacements in several sensor points on the boundary of the model and natural frequencies are obtained on the outputs of FIS (or PGFIS). They are sent back to the EA where the fitness function of each chromosome is computed as it is shown in Fig. 24.

The condition (46) controls the number of defects. The number of input values that are sent from the EA to the fuzzy inference systems depends on the number of active genes. Thus, in the approximation block there are several fuzzy inference systems with a different number of input neurons (Fig. 38). Every FIS is responsible for the approximation of natural frequencies or displacements on the boundary of the model with different number of internal defects.

The EA enables finding multiple defects. It minimizes the fitness function which is formulated as a weighted sum of differences between measured boundary displacements and natural frequencies of the examined body and computed displacements and natural frequencies for the numerical model of the body with an assumed number and shapes of defects as it is shown in formulas  $J_u$  (7),  $J_{ut}$  (9) and  $J_\omega$  (11).

## 10.6. Numerical examples

A two-dimensional elastic structure is considered (Fig. 35) The geometrical and material parameters of the body are shown in Table 15. One assumes that the maximum number of defects  $D_{\max} = 2$ , however, the actual number of defects is one in the first example and two in the second. One should find the number of defects, their position and size.

To solve the problem the EA coupled with the FIS and PGFIS is applied. It should be noted that as the FANNs with one output were used in order to approximate displacements in  $S$  sensor points,  $S$  neural networks should be used.

Table 21 shows the parameters of the evolutionary algorithm and the number of sensor points and natural frequencies that take part in the identification process. The computation of the fitness function of every chromosome in the EA proceeds with the help of the FIS.

Table 22 presents the examples of the results of the defect position and size identification for a rectangular plate with one defect ( $D = 1$ ). The identification proceeds on the basis of the knowledge about 3 natural frequencies and vertical displacements in 3 sensor points. The coordinates of these sensor points are: (200.0, 20.0), (200.0, 40.0) and (200.0, 60.0). Table 23 presents the identification in the case when the number of defects  $D = 2$ .

The evolutionary algorithm, with fitness function values approximated by the FIS, found the best solution in 18 sec. in both cases — when the body has one or two internal defects. In the case of the evolutionary algorithm with the BEM the identification time was 22 min. 25 sec. and 11 min. 40 sec. [4]. So it can be said that the evolutionary algorithm with the fitness function approximated by using the fuzzy neural network is much faster than the evolutionary algorithm with the boundary element method (see Fig. 39). The time of computation using the EA with the FIS, presented in this paper, does not take into account the time needed to teach the FIS and the time needed to prepare the learning and testing sets. Such an approach is worth using when the identification of many bodies with the same shape has to be done. In such a case the time needed to prepare training and testing sets and train the FIS isn't significant.

The problem of the internal defect identification in elastic structures under dynamical loads was also tested. Table 24 shows the parameters of the evolutionary algorithm. This process utilizes the knowledge about displacements at sensor points in the time steps shown in Table 20. The computation of the fitness function of every chromosome in the EA proceeds with the help of the PGFIS.

Table 25 presents the examples of the results of the defect position and size identification for the rectangle plate with one and two defects. The identification proceeds basing on the knowledge of boundary displacements under dynamical load.

**Table 21.** Parameters of the evolutionary algorithm

Number of epochs	100
Number of chromosomes	50
Number of genes per chromosome	6
Arithmetic crossover probability	0.8
Type of mutation	Gaussian
Number of sensor points $S$	3
Number of natural frequencies $F$	3

**Table 22.** The examples of the identification results when  $D = 1$  and  $D_{\max} = 2$ 

	$X_1$ [mm]	$Y_1$ [mm]	$R_1$ [mm]	$X_2$ [mm]	$Y_2$ [mm]	$R_2$ [mm]
actual defect	130.0	55.0	9.0	0.0	0.0	0.0
found defect	126.8	47.0	9.1	0.0	0.0	0.0
actual defect	130.0	35.0	12.0	0.0	0.0	0.0
found defect	130.5	34.6	10.4	0.0	0.0	0.0
actual defect	50.0	40.0	7.0	0.0	0.0	0.0
found defect	49.0	36.0	6.7	0.0	0.0	0.0

**Table 23.** The examples of the identification results when  $D = 2$  and  $D_{\max} = 2$ 

	$X_1$ [mm]	$Y_1$ [mm]	$R_1$ [mm]	$X_2$ [mm]	$Y_2$ [mm]	$R_2$ [mm]
actual defect	65.0	35.0	8.0	135.0	45.0	11.0
found defect	57.6	43.7	7.5	135.6	40.8	11.4
actual defect	83.0	40.0	11.0	103.0	65.0	10.0
found defect	73.5	37.0	10.5	105.2	64.2	10.4
actual defect	65.0	35.0	8.0	100.0	45.0	8.0
found defect	75.8	35.9	7.6	102.7	40.9	7.5

**Table 24.** Parameters of the evolutionary algorithm

Number of epochs	150
Number of chromosomes	18
Number of genes per chromosome	6
Arithmetic crossover probability	0.2
Type of mutation	Gaussian

**Table 25.** The examples of the identification results on the basis of boundary displacements in elastodynamic problem

	$X_1$ [mm]	$Y_1$ [mm]	$R_1$ [mm]	$X_2$ [mm]	$Y_2$ [mm]	$R_2$ [mm]
actual defect	152.0	42.0	7.8	0.0	0.0	0.0
found defect	156.2	43.0	7.7	0.0	0.0	0.0
actual defect	148.0	42.0	9.5	62.0	58.0	8.4
found defect	155.4	38.2	12.4	59.1	62.7	7.9

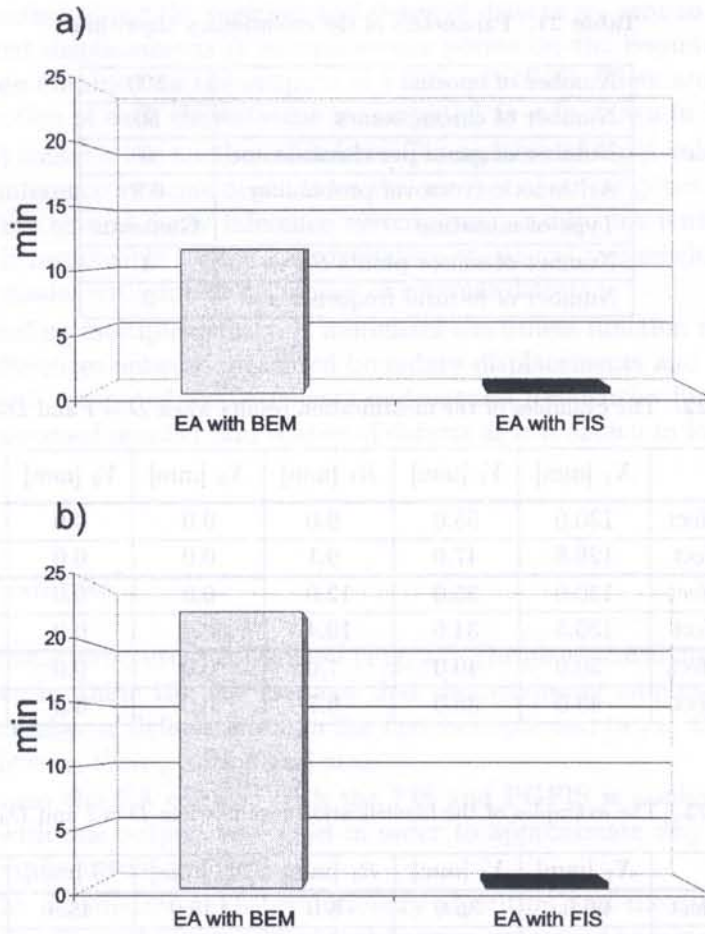


Fig. 39. Identification time using EA with BEM and EA with FIS for the body with a) one defect b) two defects

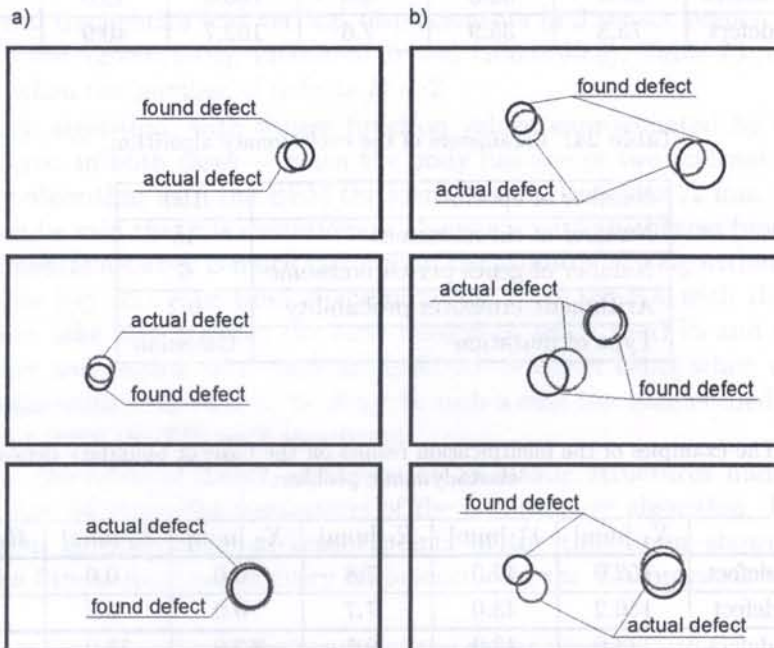


Fig. 40. The identification of a) one and b) two internal defects

Figure 40 shows the actual and found defects for six different examples. Figure 40a presents the results of the identification in the case when the actual number of defects in the body was one, Fig. 40b illustrates the situation when the actual number of defects was two. It can be said, that the CIS identifies not only the position and size, but also the number of internal defects.

## 11. FINAL CONCLUSIONS

Evolutionary computing is a very effective technique in inverse problems. This approach enables finding not only the positions and shape of defects but also the type of defects and the number of defects.

The influence of some parameters on the evolutionary identification has been examined. A lot of interesting conclusions can be formulated.

The right location of the sensor points makes the identification process easier.

The combining the measured information can shorten the time of identification. In the identification of voids, which are close to each other the combining of the measured information is necessary.

The correct selection of the forced load functions makes the identification process more effective.

Unfortunately, this approach is very time consuming due to the necessity of solving many direct problems in the form boundary value or boundary initial value problems. To speed up the evolutionary identification one can use parallel or distributed evolutionary computing techniques.

The alternative approach of speeding up the solutions of the identification problem is to use computational intelligence system (CIS) which consists of the evolutionary algorithm and the artificial neural network or the adaptive fuzzy inference system.

Approximation abilities of the radial basis function neural network and two fuzzy inference systems were compared. In further studies, the fuzzy inference system with Gaussian and pseudo-Gaussian membership functions were chosen. Such approximators were trained using the evolutionary algorithm, the steepest descent method with momentum and the conjugate gradient method. The best results were reached in the case when the FIS was trained using the gradient methods – the SD method at first and the CG method next. Both learning rate and momentum rate should change during the training process, the training input-output pairs should be presented randomly.

The fuzzy inference system can be used as an approximator of the solutions of the boundary value problem in identification tasks. It can assist in the fitness value computation.

The evolutionary algorithm combined with the FIS produce the computational intelligence system that is able to identify the number, shape and position of defects.

## ACKNOWLEDGEMENT

The support from the Foundation for Polish Science is gratefully acknowledged.

## REFERENCES

- [1] J.T. Aleander. *An Indexed Bibliography of Distributed Genetic Algorithms*. University of Vaasa, Report 94-1-PARA, Vaasa, Finland, 2000.
- [2] J. Arabas. *Lectures in Evolutionary Algorithms* (in Polish). WNT, Warszawa, 2001.
- [3] M. Bonnet, T. Burczyński, M. Nowakowski. 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, 2002.
- [4] T. Burczyński. *The Boundary Element Method in Mechanics*. WNT, Warszawa, 1995.
- [5] T. Burczyński (ed). *Computational Sensitivity Analysis and Evolutionary Optimization of Systems with Geometrical Singularities*. ZN KWMiMKM, Gliwice, 2002.
- [6] T. Burczyński, W. Beluch. The identification of cracks using boundary elements and evolutionary algorithms. *Engineering Analysis with Boundary Elements*, **25**: 313–322, 2001.

- [7] T. Burczyński, W. Beluch, A. Długosz, P. Orantek, M. Nowakowski. Evolutionary methods in inverse problems of engineering mechanics. In: M. Tanaka, G.S. Dulikravich, eds., *Inverse Problems in Engineering Mechanics II*, 553–562. Elsevier, 2000.
- [8] T. Burczyński, W. Beluch, A. Długosz, P. Orantek, M. Nowakowski. Evolutionary computation in optimization and identification. *Computer Assisted Mechanics and Engineering Science*, **9**: 3–20, 2002.
- [9] T. Burczyński, M. Bonnet, P. Fedeliński, M. Nowakowski. Sensitivity analysis and identification of material defects in dynamical systems. *Journal of Mathematical Modelling and Simulation in System Analysis SAMS*, **42C4**: 559–674, 2002.
- [10] T. Burczyński, E. Majchrzak, W. Kuś, P. Orantek, M. Dzięwoński. Evolutionary computation in inverse problems. In: T. Burczyński, A. Osyczka, eds., *Evolutionary Methods in Mechanics*. Kluwer, Dordrecht, 2004, 33–46.
- [11] T. Burczyński, P. Orantek, A. Skrobol. Fuzzy-neural and evolutionary computation in identification of defect. *Journal of Theoretical and Applied Mechanics*, **42(3)**: 445–460, 2004.
- [12] T. Burczyński, A. Osyczka. *Evolutionary Methods in Mechanics*. Kluwer, Dordrecht 2004.
- [13] T. Burczyński, A. Skrobol. Approximation of a boundary-value problem using artificial neural networks. In: T. Burczyński, W. Cholewa, W. Moczulski, eds., *Recent Developments in Artificial Intelligence Methods*, AI-METH Series, 79–84, Gliwice, 2004.
- [14] T. Burczyński, A. Skrobol. Using of radial basis function and fuzzy-artificial neural networks in approximation of boundary problem (in Polish). *IV Sympozjum Modelowanie i Symulacja Komputerowa w Technice*, 31–36, Łódź, 2005.
- [15] E. Cantu-Paz. A Survey of parallel genetic algorithms. *Calculateurs Paralleles, Reseaux et Systems Repartis*, **10(2)**: 141–171, Paris, 1998.
- [16] J.R. Jang, Ch. Sun, E. Mizutani. *Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence*. Prentice-Hall, Upper Saddle River, 1997.
- [17] W.-H. Ho, C.-J. Lin. A pseudo-Gaussian-based neural fuzzy system and its applications. *The Seventh Conference on Artificial Intelligence and Applications*, 12–16, Taiwan, 2002.
- [18] Z. Michalewicz. *Genetic Algorithms + Data Structures = Evolution Programs*. Springer-Verlag, Berlin 1996.
- [19] M. Kleiber (ed.). *Handbook of Computational Solid Mechanics*. Springer-Verlag, Berlin 1998.
- [20] M. Nowakowski. *Sensitivity analysis and shape identification of internal boundaries of vibrating mechanical systems using boundary element method*. Doctor's thesis (in Polish). Politechnika Śląska, Gliwice, 2000.
- [21] S. Osowski. *Sieci Neuronowe w Ujęciu Algorytmicznym*. WNT, Warszawa, 1996.
- [22] G. Piątkowski, L. Ziemiański. Neural network identification of a circular hole in the rectangular plate. In: L. Rutkowski, J. Kacprzyk, eds., *Neural Networks and Soft Computing*, 778–783, Heidelberg, Physica-Verlag Springer, 2003.
- [23] A. Portela, M.H. Aliabadi and D.P. Rooke. The dual boundary element method: effective implementation for crack problems. *International Journal of Numerical Methods in Engineering*, **33**: 1269–1287, 1992.
- [24] D. Rutkowska. *Computational Intelligent Systems* (in Polish). Akademicka Oficyna Wydawnicza PLJ, Warszawa, 1997.
- [25] R. Tanese. Distributed genetic algorithms. In: J.D. Schaffer ed., *Proc. 3rd ICGA*, 434–439. San Mateo, USA, 1989.
- [26] Z. Waszczyszyn, L. Ziemiański. Neural networks in mechanics of structures and materials – new results and prospects of applications. *Computer and Structures*, **79**: 2261–2276, 2001.
- [27] Z. Waszczyszyn, L. Ziemiański. Neural networks in the identification analysis of structural mechanics problems. *CISM Advanced School on Parameter Identification of Materials and Structures*, Udine, 2003.
- [28] L. Ziemiański, G. Piątkowski. Use of neural networks for damage detection in structural elements using wave propagation. In: *Computational Engineering using Metaphors from Nature*, 25–30, Civil-Comp Press, Edinburgh, 2000.
- [29] O.C. Zienkiewicz, R. Taylor. *The Finite Element Method*, Butterworth, Boston 2000.