COMPDYN 2011
3rd ECCOMAS Thematic Conference on
Computational Methods in Structural Dynamics and Earthquake Engineering
M. Papadrakakis, M. Fragiadakis, V. Plevris (eds.)
Corfu, Greece, 25–28 May 2011

SOFT COMPUTING IN STRUCTURAL DYNAMICS

T. Burczyński[1,2], R. Górski[1], A. Poteralski[1], M. Szczepanik[1]

[1]Department of Strength of Materials & Computational Mechanics, Silesian University of Technology
Konarskiego 18a, 44-100 Gliwice, Poland
e-mail: tb@polsl.pl

[2] Institute of Computer Science, Cracow University of Technology
Warszawska 24, 31-155 Kraków, Poland
tburczyn@pk.edu.pl

Keywords: soft computing, evolutionary algorithm, artificial immune system, particle swarm optimizer, optimization, finite element method, boundary element method, dynamics

Abstract. *The paper is devoted to new computational techniques in structural dynamics where one tries to study, model, analyze and optimize very complex phenomena, for which more precise scientific tools of the past were incapable of giving low cost and complete solution. Soft computing methods differ from conventional (hard) computing in that, unlike hard computing, they are tolerant of imprecision, uncertainty, partial truth, and approximation. The paper deals with an application of the bio-inspired methods, like the evolutionary algorithms (EA), the artificial immune systems (AIS) and the particle swarm optimizers (PSO) to optimization problems. Structures considered in this work are analyzed by the finite element method (FEM) and the boundary element method (BEM). The bio-inspired methods are applied to optimize shape, topology and material properties of 3D structures modeled by the FEM and to optimize location of stiffeners in 2D reinforced plates modeled by the coupled BEM/FEM. The structures are optimized using the criteria depend on frequency, displacements or stresses. Numerical examples demonstrate that the methods based on the soft computation are effective for solving computer aided optimal design problems.*

1 INTRODUCTION

Structures are frequently subjected to dynamic loads and it is very important to analyze their transient dynamic response. Important properties of vibrating structures are eigenfrequencies. The dynamic response or natural frequencies of structures can be established by changing shape, topology and material properties of structures. Another possibility of the response improvement is applying stiffeners. The choice of their number, properties and location in a structure decides about the effectiveness of reinforcement. Reinforced structures are often used in practice because they are resistant, stiff and stable. A typical area of application of such structures is an aircraft industry, where light, stiff and highly resistant structures are required. Many aircraft elements are made as thin panels reinforced by stiffeners.

Dynamic response of structures with an arbitrary geometry, material properties and boundary conditions can be obtained by carrying out laboratory tests but they are usually very expensive and time consuming. In order to reduce costs and time, computer simulations are performed instead of experimental investigations. As a result, dynamic quantities of interest like displacements, velocities, accelerations, forces, stresses, i.e. can be determined. The most versatile methods of analysis of structures subjected to arbitrary static and time dependent boundary conditions are the finite element method (FEM) and the boundary element method (BEM). The coupling of these methods is very desirable in order to exploit their advantages. Optimal properties of dynamically loaded structures can be searched using the computer aided optimization tools.

In the present paper, coupling FEM and BEM with the bio-inspired methods in optimization of dynamically loaded structures is presented. The evolutionary algorithms (EA), the artificial immune systems (AIS) and the particle swarm optimizers (PSO) are used to optimize shape, topology and material properties of 3D freely vibrating structures and 2D dynamically loaded stiffened plates. The former are analyzed by the FEM and the latter by the coupled BEM/FEM.

2 SOFT COMPUTING METHODS

Soft computing techniques resemble human reasoning more closely than traditional techniques, which are largely based on conventional logical systems or rely heavily on the mathematical capabilities of a computer. These computing techniques are often used to complement each other in applications. It should be pointed out that simplicity and complexity of systems are relative, and certainly, most successful mathematical modeling of the past have also been challenging and very significant.

Unlike hard computing schemes, which strive for exactness and for full truth, soft computing techniques exploit the given tolerance of imprecision, partial truth, and uncertainty for a particular problem. Another common contrast comes from the observation that inductive reasoning plays a larger role in soft computing than in hard computing.

Three important areas of soft computing methods, namely:

- Evolutionary Computation (EC),
- Artificial Immune Systems (AIS),
- Particle Swarm Methods (PSM),

are presented in the paper as soft computing methods.

2.1 Evolutionary Computation (EC)

Evolutionary algorithms [1, 10] are algorithms searching the space of solutions and they are based 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. Evolutionary algorithms operate on

populations of individuals, so while the algorithm works, all the time we deal with a set of problem solutions. 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 optimization problems. The adaptation is computed using a fitness function. All genes of an individual decide about the fitness function value. A flowchart of an evolutionary algorithm is presented in Fig. 1.

In the first step, an initial population of individuals is created. Usually, the values of the genes of particular individuals are randomly generated. In the next step, the fitness function value for each individuals is computed. Then, evolutionary operators change genes of the parent population individuals, they are then selected for the offspring population, which becomes a parent population and the algorithm is continuing iteratively till the end of the computation. The termination condition of the computation can be formulated as the maximum number of iterations.

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

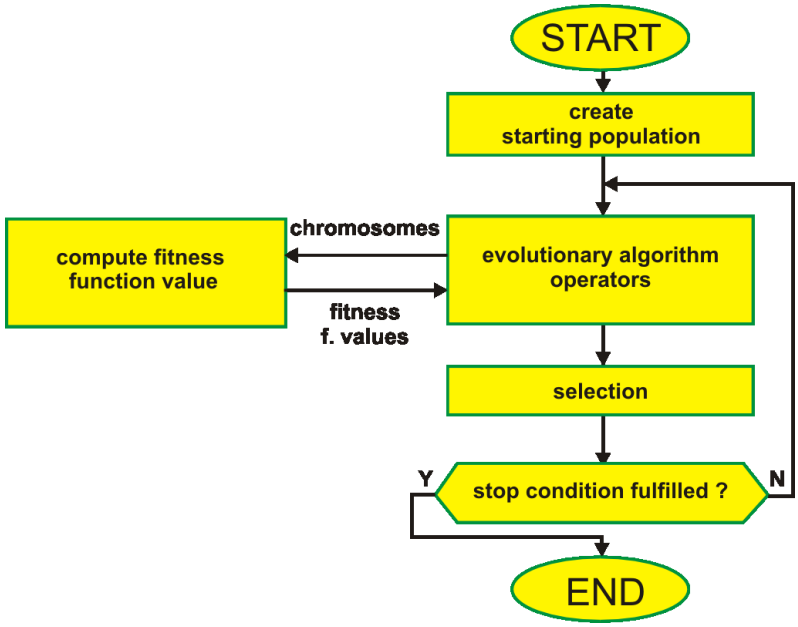

Figure 1: A flowchart of an evolutionary algorithm.

A single-chromosome individual (called a chromosome) ch_i, $i=1,2,...,N$, where N is the population size, may be presented by means of a column or line matrix, whose elements are represented by genes g_{ij}, $j=1,2,...,n$, where n is the number of genes in a chromosome. The sample chromosome is presented in Fig. 2.

$$ch_i = \begin{bmatrix} g_{i1} \\ g_{i2} \\ \vdots \\ g_{in} \end{bmatrix}$$

Figure 2: Structure of an individual.

Evolutionary operators change gene values like the biological mechanisms of mutation and crossover. Different kinds of operators are presented in publications, and the basic ones are:

- uniform mutation,
- mutation with Gaussian distribution,
- boundary mutation,
- simple crossover,
- arithmetical crossover.

A uniform mutation changes the values of randomly chosen genes in randomly selected individual. The new values of the genes are drawn in such a way that they could fulfill constrains imposed on the variation of the gene values. The diagram of how an operator works is presented in Fig. 3.

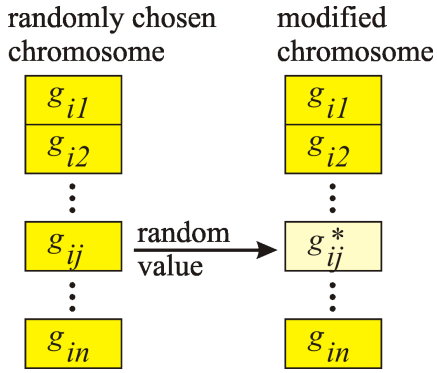

Figure 3: A diagram of an uniform mutation.

A mutation with Gaussian distribution is an operator changing the values of an individual's genes randomly, similarly to uniform mutation. New values of the genes are created by means of random numbers with Gaussian distribution. The operator searches the individual's surrounding.

A boundary mutation (Fig. 4) operates similarly to a uniform mutation, however, new values of the genes are equal to the left or right value from the gene variation range (left or right constraint on gene values).

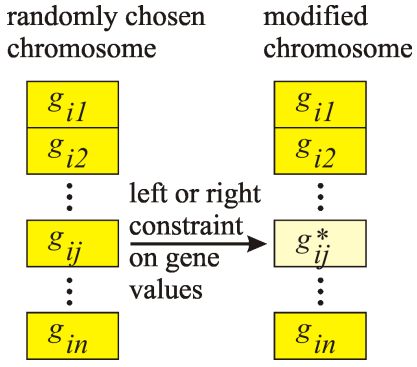

Figure 4: A diagram of boundary mutation

A simple crossover is an operator creating an offspring on the basis of two parent individuals. A cutting position is drawn (Fig. 5), and a new individual consists of the genes coming partly form the first and partly form the second individual.

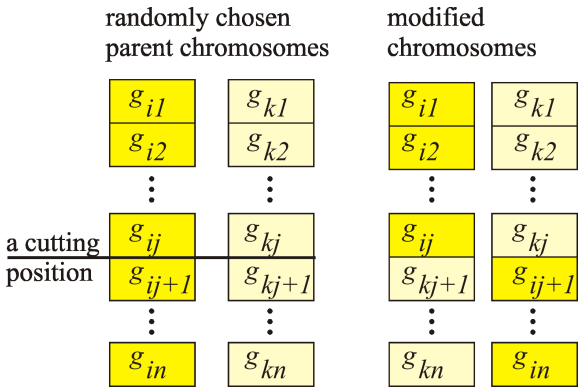

Figure 5: A diagram of a simple crossover

An arithmetical crossover has no biological counterpart. A new individual is formed similarly to a simple crossover, on the basis of two parent individuals, however, the values of the individual's genes are defined as the average value of the parent individuals' genes (Fig. 6).

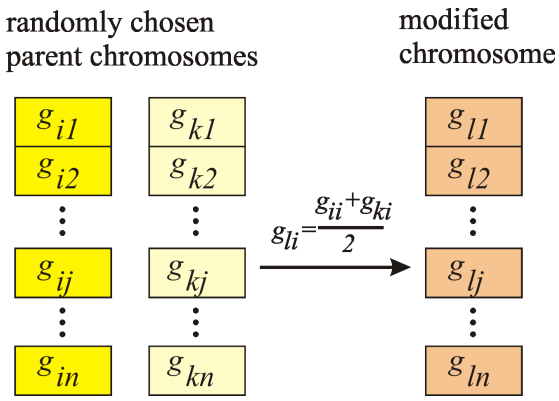

Figure 6: A diagram of an arithmetical crossover

An important element of an evolutionary algorithm is the mechanism of selection. The probability of the individual's survival depends on the value of the fitness function. 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.

2.2 Artificial Immune Systems (AIS)

The artificial immune systems (AIS) are developed on the basis of a mechanism discovered in biological immune systems [11]. An immune system is a complex system which contains distributed groups of specialized cells and organs. The main purpose of the immune system is to recognize and destroy pathogens - funguses, viruses, bacteria and improper functioning cells. The lymphocytes cells play a very important role in the immune system. The lymphocytes are divided into several groups of cells. There are two main groups B and T cells, both contains some subgroups (like B-T dependent or B-T independent). The B cells contain antibodies, which could neutralize pathogens and are also used to recognize pathogens. There is a big diversity between antibodies of the B cells, allowing recognition and neu-

tralization of many different pathogens. The B cells are produced in the bone marrow in long bones. A B cell undergoes a mutation process to achieve big diversity of antibodies. The T cells mature in thymus, only T cells recognizing non self cells are released to the lymphatic and the blood systems. There are also other cells like macrophages with presenting properties, the pathogens are processed by a cell and presented by using MHC (Major Histocompatibility Complex) proteins. The recognition of a pathogen is performed in a few steps (Fig. 7). First, the B cells or macrophages present the pathogen to a T cell using MHC (Fig. 7b), the T cell decides if the presented antigen is a pathogen. The T cell gives a chemical signal to B cells to release antibodies. A part of stimulated B cells goes to a lymph node and proliferate (clone) (Fig. 7c). A part of the B cells changes into memory cells, the rest of them secrete antibodies into blood. The secondary response of the immunology system in the presence of known pathogens is faster because of memory cells. The memory cells created during primary response, proliferate and the antibodies are secreted to blood (Fig. 7d). The antibodies bind to pathogens and neutralize them. Other cells like macrophages destroy pathogens (Fig. 7e). The number of lymphocytes in the organism changes, while the presence of pathogens increases, but after attacks a part of the lymphocytes is removed from the organism.

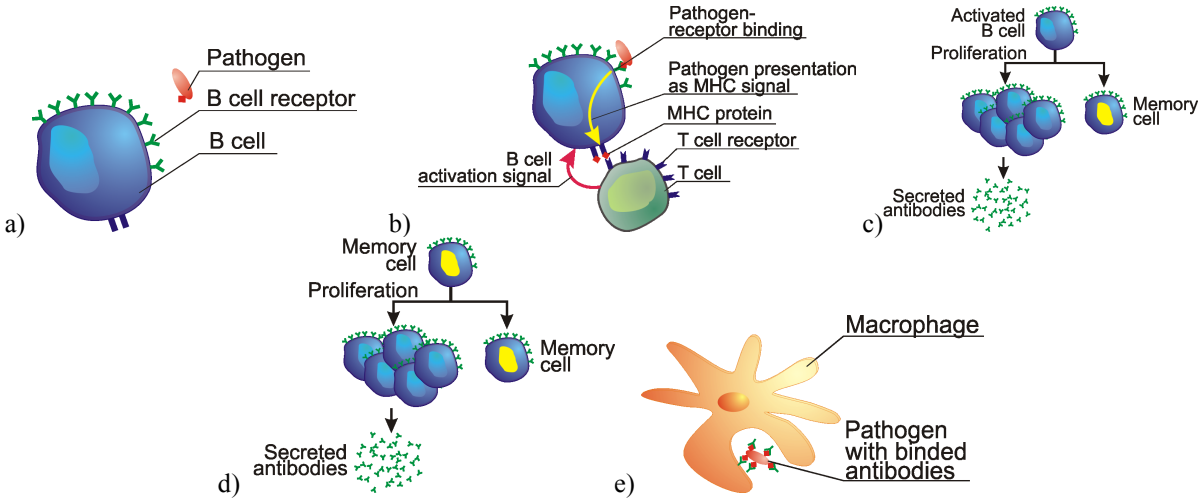

Figure 7: An immune system, a) a B cell and pathogen,
b) the recognition of pathogen using B and T cells, c) the proliferation of activated B cells, d) the proliferation of a memory cell – secondary response, e) pathogen absorption by a macrophage.

The artificial immune systems [2], [3], [4] take only a few elements from the biological immune systems. The most frequently used are the mutation of the B cells, proliferation, memory cells, and recognition by using the B and T cells. The artificial immune systems have been used to optimization problems in [5], classification and also computer viruses recognition in [2]. The cloning algorithm presented by von Zuben and de Castro [4], [5] uses some mechanisms similar to biological immune systems to global optimization problems. The unknown global optimum is the searched pathogen. The memory cells contain design variables and proliferate during the optimization process. The B cells created from memory cells undergo mutation. The B cells evaluate and better ones exchange memory cells. In Wierzchoń [13] version of Clonalg the crowding mechanism is used - the diverse between memory cells is forced. A new memory cell is randomly created and substitutes the old one, if two memory cells have similar design variables. The crowding mechanism allows finding not only the global optimum but also other local ones. The presented approach is based on the Wierzchoń [13] algorithm, but the mutation operator is changed. The Gaussian mutation is used instead of the nonuniform mutation in the presented approach.

A flowchart of an artificial immune system is presented in Fig. 8. The memory cells are created randomly. They proliferate and mutate creating B cells. The number of nc clones created by each memory cell is determined by the memory cells objective function value. The objective functions for B cells are evaluated. The selection process exchanges some memory cells for better B cells. The selection is performed on the basis of the geometrical distance between each memory cell and B cells (measured by using design variables). The crowding mechanism removes similar memory cells. The similarity is also determined as the geometrical distance between memory cells. The process is iteratively repeated until the stop condition is fulfilled. The stop condition can be expressed as the maximum number of iterations.

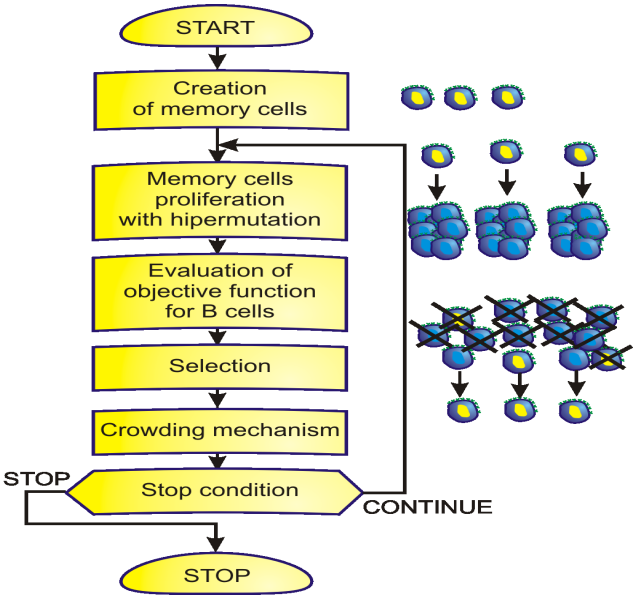

Figure 8: An artificial immune system.

2.3 Particle Swarm Methods (PSM)

The particle swarm algorithms [12], similarly to the evolutionary and immune algorithms, are developed on the basis of the mechanisms discovered in the nature. The swarm algorithms are based on the models of the animals social behaviours: moving and living in the groups. The animals relocate in the three-dimensional space in order to change their stay place, the feeding ground, to find the good place for reproduction or to evading predators. We can distinguish many species of the insects living in swarms, fishes swimming in the shoals, birds flying in flocks or animals living in herds (Fig. 9).

a)  b) 

Figure 9: Particles swarms: a) fish shoal (http://www.sxc.hu/photo/1187373), b) bird flock (http://www.sxc.hu/photo/1095384).

A simulation of the bird flocking was published in [12]. They assumed that this kind of the coordinated motion is possible only when three basic rules are fulfilled: collision avoidance, velocity matching of the neighbours and flock centring. The computer implementation of these three rules showed very realistic flocking behaviour flaying in the three dimensional space, splitting before obstacle and rejoining again after missing it. The similar observations concerned the fish shoals. Further observations and simulations of the birds and fishes behaviour gave in effect more accurate and more precise formulated conclusions [9]. The results of this biological examination where used by Kennedy and Eberhart [7], who proposed Particle Swarm Optimiser – PSO. This algorithm realizes directed motion of the particles in n-dimensional space to search for solution for n-variable optimisation problem. PSO works in an iterative way. The location of one individual (particle) is determined on the basis of its earlier experience and experience of whole group (swarm). Moreover, the ability to memorize and, in consequence, returning to the areas with convenient properties, known earlier, enables adaptation of the particles to the life environment. The optimisation process using PSO is based on finding the better and better locations in the search-space (in the natural environment that are for example hatching or feeding grounds).

The algorithm with continuous representation of design variables and constant constriction coefficient (constricted continuous PSO) has been used in presented research. In this approach each particle oscillates in the search space between its previous best position and the best position of its neighbours, with expectation to find new best locations on its trajectory. When the swarm is rather small (swarm consists of several or tens particles) it can be assumed that all the particles stay in neighbourhood with currently considered one. In this case we can assume the global neighbourhood version and the best location found by swarm so far is taken into account – current position of the swarm leader (Fig. 10).

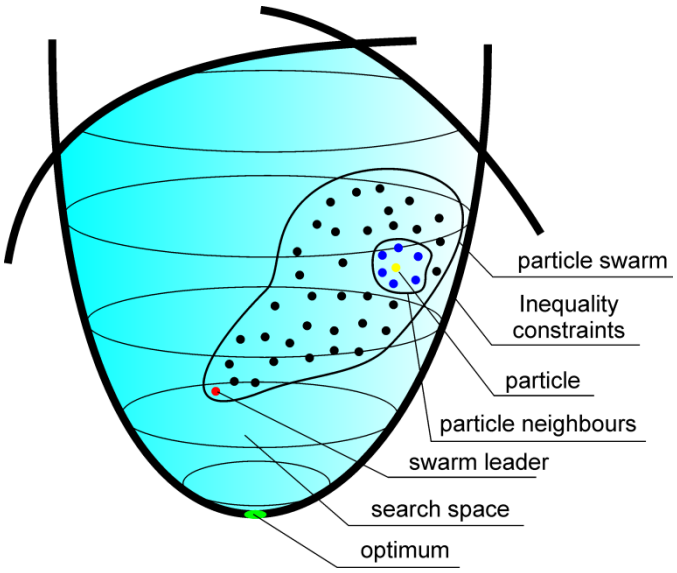

Figure 10: The idea of the particle swarm.

The position of the i-th particle is changed by stochastic velocity vi, which is dependent on the particle distance from its earlier best position and position of the swarm leader. This approach is given by the following equations:

$$v_{ij}(k+1) = wv_{ij}(k) + \phi_{1j}(k)\left[q_{ij}(k) - d_{ij}(k)\right] + \phi_{2j}(k)\left[\hat{q}_{ij}(k) - d_{ij}(k)\right] \tag{1}$$

$$d_{ij}(k+1) = d_{ij}(k) + v_{ij}(k+1), \quad i = 1,2,...,m \; ; \; j = 1,2,...,n \tag{2}$$

where:

$$\phi_{1j}(k) = c_1 r_{1j}(k); \; \phi_{2j}(k) = c_2 r_{2j}(k),$$

m – number of the particles,
n – number of design variables (problem dimension),
w – inertia weight,
c_1, c_2 – acceleration coefficients,
r_1, r_2 – random numbers with uniform distribution [0,1],
$d_i(k)$ – position of the i-th particle in k-th iteration step,
$v_i(k)$ – velocity of the i-th particle in k-th iteration step,
$q_i(k)$ – the best found position of the i-th particle found so far,
$\hat{q}_i(k)$ – the best position found so far by swarm – the position of the swarm leader,
k – iteration step.

The velocity of i-th particle is determine by three components of the sum in Equation (1). The first component $wv_i(k)$ plays the role of the constraint to avoid excessive oscillation in the search space. The inertia weight w controls the influence of particle velocity from the previous step on the current one. In this way this factor controls the exploration and exploitation. Higher value of inertia weight facilitates the global searching, and lower – the local searching. The inertia weight plays the role of the constraint applied for the velocities to avoid particles dispersion and guaranteeing convergence of the optimisation process. The second component $\phi_1(k)\left[q_i(k) - d_i(k)\right]$ realizes the cognitive aspect. This component represents the particle distance from its best position found earlier. It is related to the natural inclination of the individuals (particles) to the environments where they had the best experiences (the best value of the fitness function). The third component $\phi_2(k)\left[\hat{q}_i(k) - d_i(k)\right]$ represents the particle distance from the position of the swarm leader. It refers to the natural inclination of the individuals to follow the other which achieved a success.

The flowchart of the particle swarm optimiser is presented in Fig. 11. At the beginning of the algorithm the particle swarm of assumed size is created randomly. Starting positions and velocities of the particles are created randomly. The objective function values are evaluated for each particle. In the next step the best positions of the particles are updated and the swarm leader is chosen. Then the particles velocities are modified by means of the Equation (1) and particles positions are modified according to the Equation (2). The process is iteratively repeated until the stop condition is fulfilled. The stop condition is typically expressed as the maximum number of iterations.

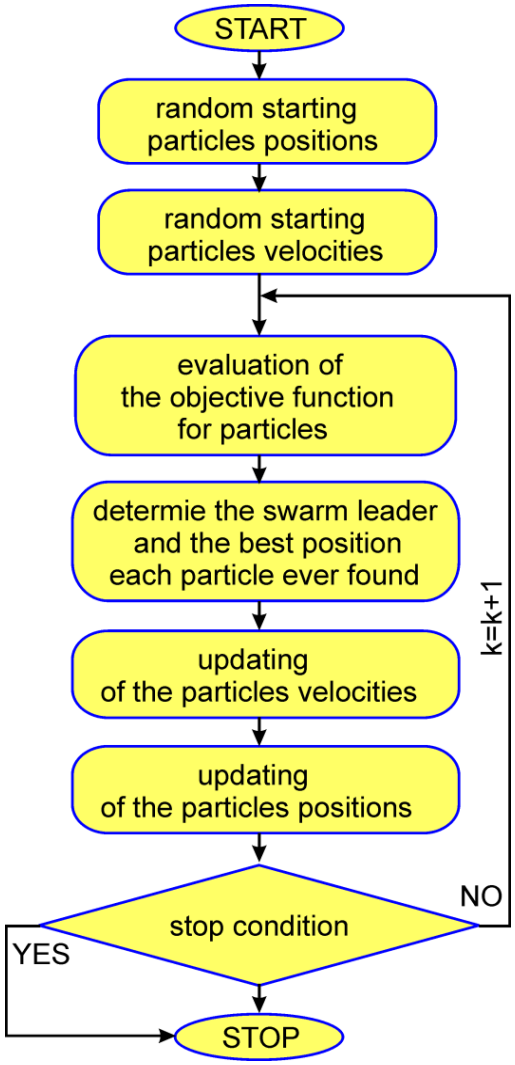

Figure 11: Particle swarm optimiser – block diagram.

The general effect is that each particle oscillates in the search space between its previous best position (position with the best fitness function value) and the best position of its best neighbour (relatively swarm leader), hopefully finding new best positions (solutions) on its trajectory, what in whole swarm sense leads to the optimal solution.

3 SOFT COMPUTING IN OPTIMIZATION OF DYNAMICALLY LOADED STRUCTURES

3.1 Evolutionary generalized optimization of structures modeled by the FEM

Consider a structure which, at the beginning of an evolutionary process, occupies a domain $\Omega_0 \left(in\ E^3 \right)$, bounded by a boundary Γ_0. The domain Ω_0 is filled by a elastic homogeneous and isotropic material of a Young's modulus E_0 and a Poisson ratio ν. The 3-D structures are considered in the framework of the linear theory of elasticity. During the evolutionary process the domain Ω_t, its boundary Γ_t and the field of Young's modulus $E\left(x,y,z\right) = E_t, \left(x,y,z\right) \in \Omega_t$ can change for each generation t (for t=0, E_0=const). The evolutionary process proceeds in an environment in which the structure fitness is describing by

maximization of the objective functions:

a) maximization of the first eigenfrequency

$$\max(\omega_1) \tag{3}$$

with a constraint imposed on the volume of the structure

$$\begin{aligned} V &\equiv |\Omega| \\ V &\le V^{\max} \end{aligned} \tag{4}$$

b) maximization of the difference between first, second and third eigenfrequency

$$\max\left[(\omega_2 - \omega_1) + (\omega_3 - \omega_2)\right] \tag{5}$$

with a constraint imposed on the volume of the structure (4)

c) maximization of the difference between first, second, third eigenfrequency and forced vibration frequency ω_{forced}

$$\max\left[\left|\omega_1 - \omega_{forced}\right| + \left|\omega_2 - \omega_{forced}\right| + \left|\omega_3 - \omega_{forced}\right|\right] \tag{6}$$

with a constraint imposed on the volume of the structure (4).

The distribution of Young's modulus $E(x,y,z)$, $(x,y,z) \in \Omega_t$ in the structure is described by a hyper surface $W(x,y,z), (x,y,z) \in H^3$. The hyper surface $W(x,y,z)$ is stretched under $H^3 \subset E^3$ and the domain Ω_t is included in H^3, i.e. $(\Omega_t \subseteq H^3)$.

The shape of the hyper surface $W(x,y,z)$ is controlled by genes d_j, j=1,2,…,N, which create a chromosome

$$ch = \langle d_1, d_2, ..., d_j, ..., d_N \rangle \tag{7}$$

Gene values are described by the function $W(x,y,z)$ in interpolation nodes (control points) $(x,y,z)_j$, i.e. $d_j = W\left[(x,y,z)_j\right]$, j=1,2,…,N.

The following constraints are imposed on genes

$$d_j^{\min} \le d_j \le d_j^{\max} \tag{8}$$

where $d_j^{\min}$ - the minimum value of the gene and $d_j^{\max}$ - the maximum value of the gene.

The assignation of Young's moduli to each finite element $\Omega_e, e = 1, 2, ..., R$ is performed by the mapping:

$$E_e = W\left[(x,y,z)_e\right], (x,y,z)_e \in \Omega_e, e = 1, 2, ..., R \tag{9}$$

It means that each finite element can have different material.

When the value of Young's modulus for the e-th finite element is included in the interval $0 \le E_e < E_{\min}$, the finite element is eliminated and the void is created, the interval $E_{\min} \le E_e < E_{\max}$, the finite element remains having the value of the Young's modulus from this material. As a result, shape, topology and material properties of the structure are changing simultaneously and this procedure is called *evolutionary generalized optimization*.

Example 1 – maximization of the first eigenfrequency of a 3D bracket

A structure like a 3D bracket (Fig. 12a) is optimized. The criterion of optimization is the maximization of the first eigenfrequency. The best solution obtained after 88 generations is presented in Fig. 12b. The Table 1 contains input data.

Minimal Young's module	Maximal volume
0.4 x $2*10^5$ MPa	4000 mm^3
Numbers of chromosomes	
100	

Table 1: Input data.

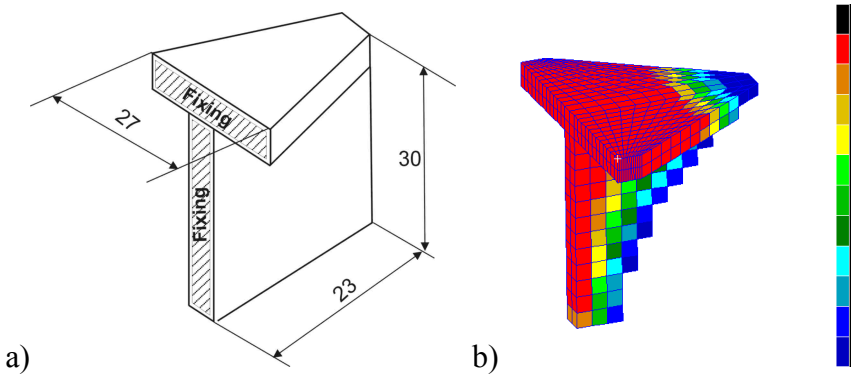

Figure 12: A 3-D bracket: a) geometrical dimensions b) distribution of Young's moduli

Example 2 – maximization of the difference between first, second and third eigenfrequency of a rectangular prism

A 3D structure in the form of a rectangular prism (Fig. 13a) is optimized. The criterion of optimization is the maximization of the difference between first, second and third eigenfrequencies. The best solution in the form of a distribution of Young's moduli obtained after 169 generations is performed in Fig. 13b. Input data are included in Table 2.

Minimal Young's module	Maximal volume
0.4 x $2*10^5$ MPa	4.8e4 mm^3
Numbers of chromosomes	Dimensions of cubicoid
100	200 x 80 x 12 mm

Table 2: Input data.

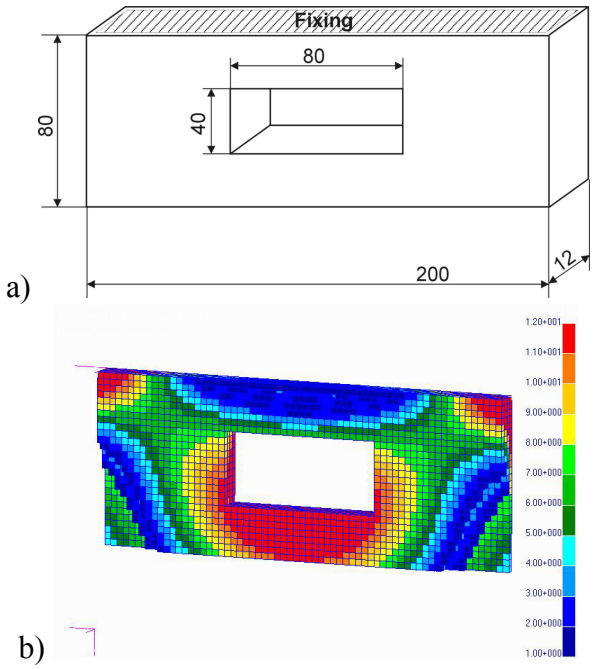

Figure 13: A rectangular prism: a) dimensions, b) distribution of Young's moduli.

Example 3 - maximization of the difference between first, second and third eigenfrequency and forced vibration frequency of a rectangular prism

The last example concerns optimization of a 3D structure from the previous example (Fig. 13a). The criterion of optimization is the maximization of the difference between first, second third eigenfrequencies and forced vibration frequency. The best solution is obtained after 134 generations is presented in Fig. 14. Input data are included in Table 3.

Minimal Young's module	Maximal volume
0.4 x $2*10^5$ MPa	80000 mm^3
Numbers of chromo-somes	Dimensions of cubicoid
100	200 x 80 x 12 mm

Table 3. Input data

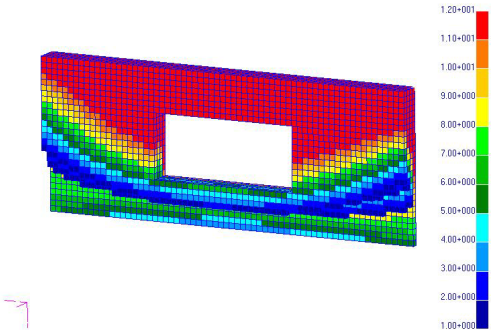

Figure 14: Distribution of Young's moduli for a rectangular prism obtained for the resonance criterion.

3.2 Bio-inspired optimization of structures modeled by the coupled BEM/FEM

A two-dimensional, homogenous, isotropic and linear elastic deformable body with boundary Γ_1 and occupying domain Ω_1, is considered. The body is modeled as a plate in plane stress or strain and it is reinforced by the stiffener occupying domain Ω_2. The body is supported (displacements $u(x,\tau)$ are known at a part of the outer boundary) and subjected to dynamic tractions $t(x,\tau)$ (where τ is time), applied at the outer boundary, as shown in Fig.15.

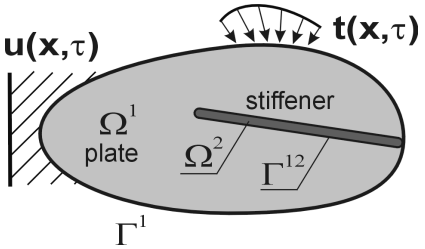

Figure 15: Reinforced plate subjected to dynamic loads.

The plate is modeled by the boundary element method (BEM) [6] and the stiffener by the finite element method (FEM) using beam finite elements, attached along the Γ_{12} boundary (the interface). A perfect bonding between the plate and the stiffener is assumed. The whole structure is analyzed by the coupled BEM/FEM and the subregion method [8]. The method allows modeling of bodies with many plate subdomains and stiffeners of different properties. The numerical equations, which are written for each plate and beam subdomain separately, are coupled using displacement compatibility conditions and traction equilibrium conditions at all nodes along the common boundaries.

A set of algebraic equations for the plate in Fig.15 has the following form:

$$\begin{bmatrix} \boldsymbol{M}^{1} & \boldsymbol{M}^{12} \end{bmatrix} \begin{Bmatrix} \ddot{\boldsymbol{u}}^{1} \\ \ddot{\boldsymbol{u}}^{12} \end{Bmatrix} + \begin{bmatrix} \boldsymbol{H}^{1} & \boldsymbol{H}^{12} \end{bmatrix} \begin{Bmatrix} \boldsymbol{u}^{1} \\ \boldsymbol{u}^{12} \end{Bmatrix} = \begin{bmatrix} \boldsymbol{G}^{1} & \boldsymbol{G}^{12} \end{bmatrix} \begin{Bmatrix} \boldsymbol{t}^{1} \\ \boldsymbol{t}^{12} \end{Bmatrix} \tag{10}$$

where: $\mathbf{M}$ is the mass matrix, $\mathbf{H}$ and $\mathbf{G}$ are the BEM coefficient matrices, $\mathbf{u}$ and $\ddot{\mathbf{u}}$ are displacement and acceleration vectors, respectively, $\mathbf{t}$ is a vector of tractions applied at the outer boundary or the interface. The superscripts denote the matrices, which correspond to the outer boundary or the interface.

The equation of motion for the stiffener in Fig.15 in a matrix form is:

$$\boldsymbol{M}^{21}\,\ddot{\boldsymbol{u}}^{21} + \boldsymbol{K}^{21}\,\boldsymbol{u}^{21} = \boldsymbol{T}^{21}\,\boldsymbol{t}^{21} \tag{11}$$

where: $\mathbf{K}$ is the FEM stiffness matrix, $\mathbf{T}$ is the matrix, which expresses the relationship between the FE nodal forces and the BE tractions. The latter matrix allows treatment the finite element region as an equivalent boundary element region.

If the structure is subjected to time dependent boundary conditions, the dynamic interaction forces between the plate and the stiffener act along the interface. These tractions are treated as body forces distributed along the attachment line and they are unknowns of the problem. The displacement compatibility conditions and the traction equilibrium conditions at the nodes along the interface are:

$$\boldsymbol{u}^{12} = \boldsymbol{u}^{21} \quad ; \quad \boldsymbol{t}^{12} = -\boldsymbol{t}^{21} \tag{12}$$

If the above conditions are taken into account in equations for the plate (10) and stiffener (11), the following system of equations for the whole structure is obtained:

$$\begin{bmatrix} \boldsymbol{M}^1 & \boldsymbol{M}^{12} \\ 0 & \boldsymbol{M}^{21} \end{bmatrix} \begin{Bmatrix} \ddot{\boldsymbol{u}}^1 \\ \ddot{\boldsymbol{u}}^{12} \end{Bmatrix} + \begin{bmatrix} \boldsymbol{H}^1 & \boldsymbol{H}^{12} & -\boldsymbol{G}^{12} \\ 0 & \boldsymbol{K}^{21} & \boldsymbol{T}^{21} \end{bmatrix} \begin{Bmatrix} \boldsymbol{u}^1 \\ \boldsymbol{u}^{12} \\ \boldsymbol{t}^{12} \end{Bmatrix} = \boldsymbol{G}^1 \boldsymbol{t}^1 \tag{13}$$

The unknowns are displacements and tractions on the external boundary and at the interface in each time step.

Example 4 – Reinforced rectangular plate

Optimization of a reinforced rectangular plate (Fig.16) is performed by means of AIS, PSO and EA. The plate is dynamically loaded and it is reinforced by the frame-like structure composed of straight beams. The plate and the stiffeners are modeled by the boundary elements and frame finite elements, respectively. Different kinds of load and support are considered. The structure before optimization (the reference plate) is shown in Fig. 16.

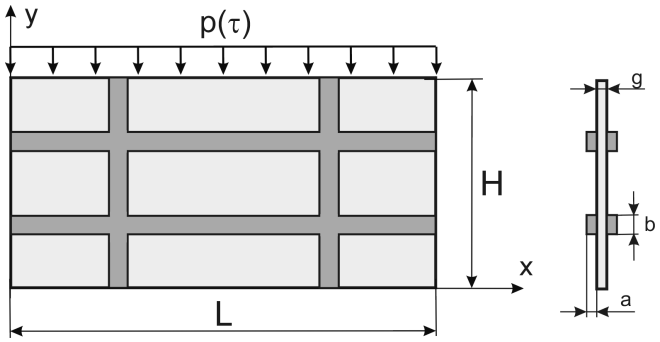

Figure 16: Reinforced rectangular plate.

The length and the height of the plate is L=10 cm and H=5 cm, respectively. The thickness of the plate is g=0.25 cm, the dimensions of beams cross-section are 2a=0.5 cm and b=0.5 cm.

The material of the plate and frame is aluminum and the mechanical properties are: modulus of elasticity E=70 GPa, Poisson's ratio ν=0.34 and density ρ=2700 kg/m^3. The material is homogeneous, isotropic and linear elastic and the plane stress is assumed.

The uniformly distributed load is applied at the upper edge of the plate. Two kinds of time dependent loads are considered (see Fig.17): a) the sinusoidal load $p(\tau)=p_o\sin(2\pi\tau/T)$ with the period of time $T=20\pi$ μs, and b) the Heaviside load $p(\tau)=p_oH(\tau)$. The value of the load in both cases is p_o=10 MPa. The time of analysis is 600 μs and the time step Δt=2 μs.

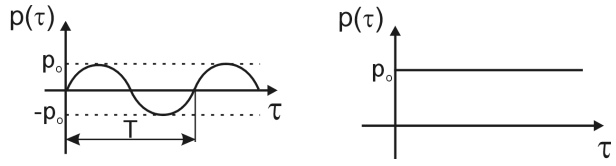

Figure 17: Dynamic loadings: a) sinusoidal, b) Heaviside.

Three different supports are considered (see Fig.18):

a) support A – the plate is fixed on the left and right edge,
b) support B – the plate is supported at two segments, each of 0.5 cm long,
c) support C – the plate is fixed at the bottom edge.

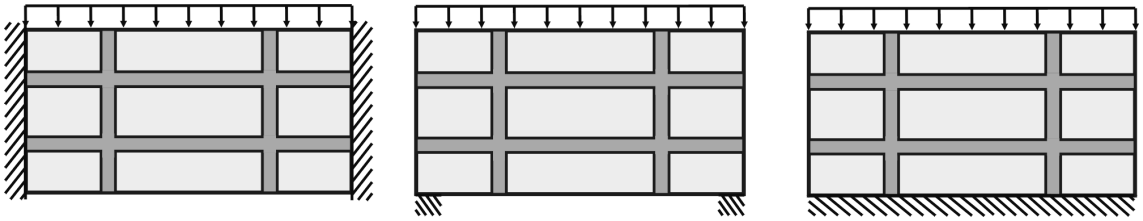

Figure 18: Types of supports: a) support A, b) support B, c) support C.

The optimal positions of stiffeners are searched in order to maximize stiffness of the plate. The maximal dynamic vertical displacement on the loaded edge is minimized. Because of symmetry of the structure and boundary conditions, only a half of the structure is considered. The number of design variables defining the position of the frame is 4: X1, X2, Y1 and Y2 (see Fig.19). The longer beams are parallel to the x axis. The end points of beams can move along the edges of the plate within the constraints, as shown in Fig.19. The constraints on design variables are imposed: X1 and X2 variables are within the range from 0.5 to 4.75 cm, Y1 from 0.5 to 2.25 cm and Y2 from 2.75 to 4.5 cm. The parameters of AIS are: the number of memory cells and the clones is 6, the crowding factor and the Gaussian mutation is 0.5. The parameters of EA are: the number of chromosomes is 20, the probability of the Gaussian mutation is 0.5, the probability of a simple and arithmetic crossover is 0.05. The parameters of PSO are: numbers of particles is 20, inertia weight is 0.73 and two acceleration coefficients are 1.47.

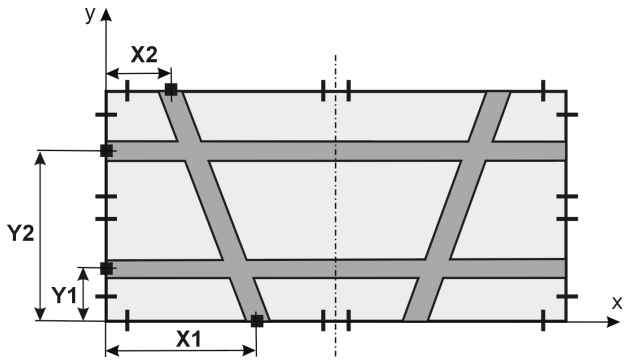

Figure 19: Design variables and constraints.

The total number of boundary and finite elements in the BEM/FEM analysis is 120 and 120, respectively (each horizontal and vertical beam is discretized into 40 and 20 finite elements, respectively). The number of boundary and finite elements during the optimization is constant.

The values of design variables obtained by AIS, PSO and EA for the plate subjected to the sinusoidal load, the Heaviside load and for three kinds of supports, are presented in Table 3. The results obtained by three different methods are almost the same. The values of J_o and J (where: J_o and J is the objective function for the reference and the optimal plate, respectively) and the reduction $R=(J_o-J)/J_o \cdot 100\%$, are also presented.

A significant reduction R, resulting in the improvement of dynamic response of the optimal plates in comparison with the initial designs, can be observed. The optimal structures for different kinds of supports and for the sinusoidal and the Heaviside loads are shown in Fig.20a and Fig.20b, respectively. It can be seen that in the present example most of constraints are active.

Load	Support	Design variables [cm]				J_o [10^{-4}cm]	J [10^{-4}cm]	R [%]
		X1	X2	Y1	Y2			
AIS, PSO and EA								
Sinusoidal	A	4.75	2.86	0.88	2.75	89	76	15
	B	4.75	1.81	0.57	2.75	92	73	21
	C	1.20	1.82	0.50	2.75	82	62	24
Heaviside	A	0.50	4.75	0.50	4.50	112	91	19
	B	4.75	1.41	0.50	4.50	211	149	29
	C	0.50	2.20	1.70	2.80	49	42	14

Table 4: Values of design variables, J and R.

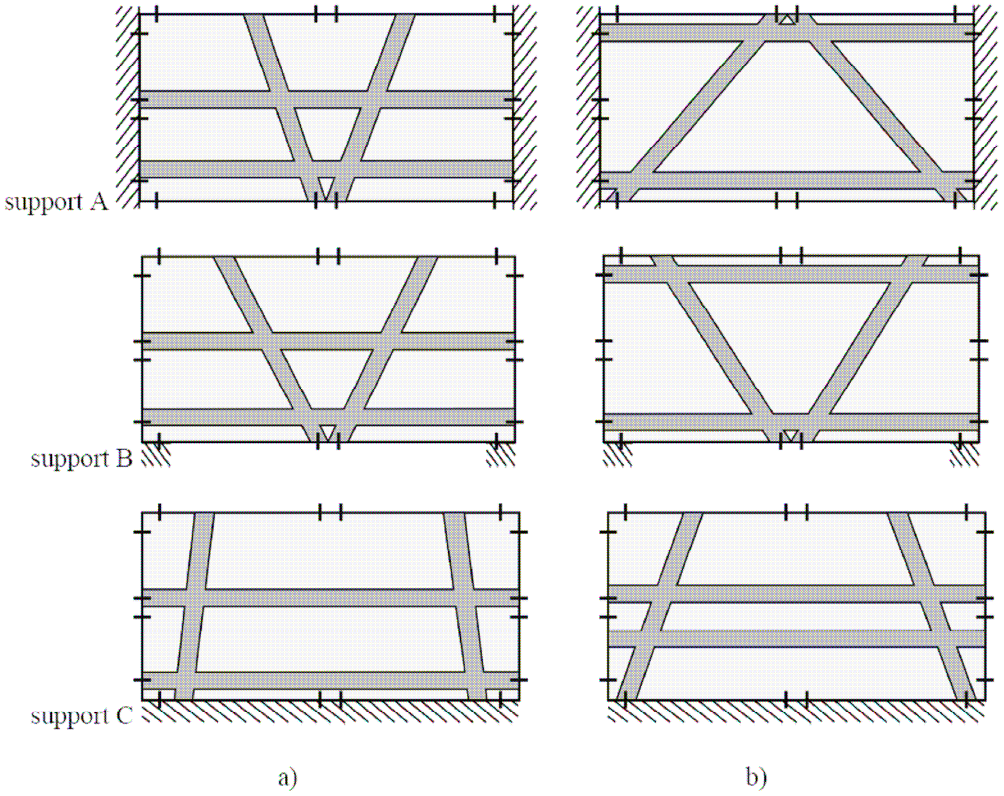

Figure 20: Optimal plates subjected to dynamic loads: a) sinusoidal, b) Heaviside.

The number of fitness function evaluations by three different bio-inspired algorithms used in this example is presented in Table 5. It can be observed that the number of fitness function evaluations for obtaining the final design variables and the corresponding fitness functions presented in Table 5 is different and depends on the applied load and support. Generally, the efficiency of the AIS and the PSO is similar and much better for this particular example, than the efficiency of the EA.

Load	Support	EA	AIS	PSO
		fitness function evaluations		
Sinusoidal	A	2515	336	360
	B	3705	408	440
	C	1952	432	520
Heaviside	A	303	276	60
	B	1526	252	120
	C	2797	528	580

Table 5: Efficiency of bio-inspired methods.

Example 5 – Reinforced plate with a hole

Optimization of a rectangular reinforced plate with a hole (Fig. 21) is performed by means of PSO with the same parameters like in the example 4. The plate is dynamically loaded and it is reinforced by 8 symmetrically distributed rods of circular cross-section. The plate and the reinforcing rods are modeled by the boundary elements and beam finite elements, respectively. The structure before optimization (the reference plate) is shown in Fig. 21.

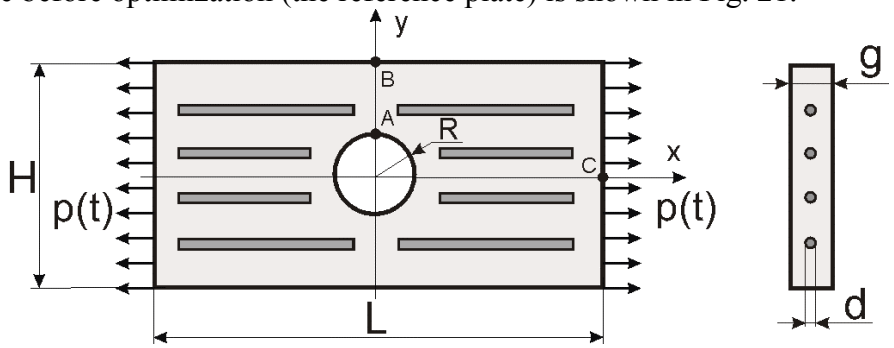

Figure 21. Reinforced plate with a hole.

The plate is stretched by a uniformly distributed load applied at its left and right edge. The dynamical load is defined by the Heaviside impulse $p(t)=p_oH(t)$, the value of the load is p_o=10 *MPa*. The time of analysis is T=300 μs and the time step Δt=3 μs.

The length and the height of the plate and the hole radius is L=10 *cm*, H=5 *cm* and R=1 *cm*, respectively. The thickness of the plate is g=1 *cm* and the diameter of each rod is d=0.3 *cm*. Distance between the rod axes for the reference plate is 1 *cm*, the length of the shorter and longer rods is 3 *cm* and 4 *cm*, respectively. Distance between the end points of the rods to the left or right edge of the plate is 0.5 *cm*.

The plane stress is assumed. The materials of the plate (p) and stiffeners (s) are epoxy and steel, respectively. They are homogeneous, isotropic and linear elastic. The values of mechanical properties are: modulus of elasticity E_p=4.5 *GPa* and E_s=210 *GPa*, Poisson's ratio ν_p=0.37 and ν_s=0.3, density ρ_p=1160 kg/m^3 and ρ_s=7860 kg/m^3.

The optimal location of reinforcement in the interior of the plate is searched and the following objective function J is minimized:

$$J = \int_0^T \frac{\left|\sigma_x^A(t)\right|^2}{\sigma_o} dt \tag{14}$$

where $\sigma_x^A(t)$ is the x-component of stress at the point A (see Fig. 21), σ_o is a nominal stress at the weakened cross-section, defined as the ratio of the applied load to the area of this cross-section, T is a time of analysis.

The objective function (14) is minimized with respect to design variables (*Xij*, *Yij*, *i,j*=1,2), defining the coordinates of the *j*-th end point of the *i*-th rod. It is assumed that during optimization the reinforcement is symmetrical with respect to two symmetry axes. Thus only a quarter of the plate with two rods is modeled (the appropriate boundary conditions at the symmetry axes are assumed) and the number of design variables is 8.

The constraints on design variables are imposed. The distance between the rods and the outer boundary (of the quarter of the plate) cannot be lower than 0.5 *cm*. The intersection of rods is not allowed.

The total number of boundary and finite elements in the BEM/FEM analysis is 92 and 64, respectively (each rod is discretized into 32 finite elements).

For this example five tests were performed and similar results were obtained. The values of design variables for the optimal solutions, rounded off to two decimal places, are: *X11*=0.97 *cm*, *Y11*=1.03 *cm*, *X12*=4.50 *cm*, *Y12*=1.50 *cm*, *X21*=1.57 *cm*, *Y21*=2.00 *cm*, *X22*=4.50 *cm* and *Y22*=2.00 *cm*. The optimal structure is shown in Fig. 22.

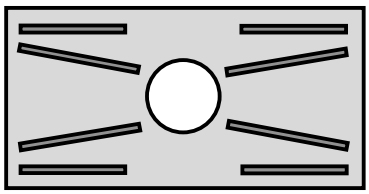

Figure 22. Optimal location of rods in the plate.

Example 6 – Reinforced cantilever plate

Optimization of a reinforced cantilever plate (Fig. 23) is performed by means of PSO with the same parameters like in the example 4. The dynamically loaded plate is reinforced at the whole non-fixed outer boundary and between two holes (at the interface between two BE regions). The reinforcement has rectangular cross-section. The plate and the reinforcement are modeled by the boundary elements and frame finite elements, respectively. The structure before optimization (the reference plate) is shown in Fig. 23.

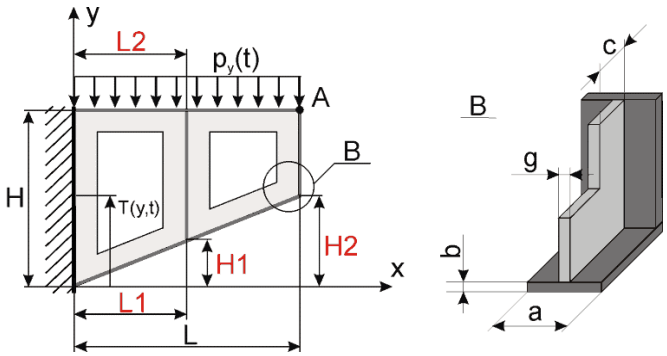

Figure 23. Reinforced cantilever plate.

The uniformly distributed load is applied at the upper edge. The plate is subjected to the sinusoidal load $p(t)=p_o sin(2\pi t/T)$. The amplitude of the load is p_o=1 *MPa* and the period of time is T=5 *ms*. The time of analysis is 12 *ms* and the time step Δt=0.02 *ms*.

The length and the height of the plate is L=50 *cm* and H=40 *cm*, respectively. The other dimensions are: a=5 *cm*, b=1 *cm*, c=5 *cm* and g=1 *cm*. The *L1*, *L2* and *H1*, *H2* defining the shape of the cantilever, are design variables of the problem and they are within the range from 15 to 35 *cm* and 0 to 25 *cm*, respectively.

The plane stress is assumed. The cantilever is linear elastic, isotropic and homogeneous and its material is steel. The values of mechanical properties are: modulus of elasticity E=210 *GPa*, Poisson's ratio ν=0.3 and density ρ=7860 kg/m^3.

The optimal shape of the cantilever is searched and the following objective function J is minimized:

$$J = \int_0^T \frac{\left|u_y^A(t)\right|^2}{u_o} dt \tag{15}$$

where $u_y^A(t)$ is a vertical displacement at the point A (see Fig. 23), u_o is an admissible displacement, T is a time of analysis.

The objective function (15) is minimized with respect to design variables (Li, Hi, i,j=1,2), defining dimensions of the structure.

The total number of boundary and finite elements in the BEM/FEM analysis is 84 and 72, respectively The quadratic elements (with 2 degrees of freedom per node) are employed for the BEM mesh. The frame elements (with 3 degrees of freedom per node) are used for the FEM mesh. During optimization the number of boundary and finite elements is constant.

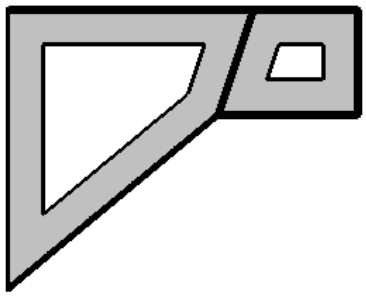

Figure 24. Optimal shape of the cantilever.

For this example five tests were performed and similar results were obtained. The values of design variables for the optimal solutions are (rounded off to two decimal places): *L1*=30.62 *cm*, *L2*=35.00 *cm*, *H1*=25.00 *cm* and *H2*=25.00 *cm*. The optimal structure is shown in Fig. 24.

4 CONCLUSIONS

In the paper, the formulation and application of the finite element method, the boundary element method and the bio-inspired methods to optimization of shape, topology and material properties of dynamically loaded structures for different criteria of optimization is presented. The bio-inspired methods can be simply implemented because they need only the values of objective functions. An important feature of these approaches is a strong probability of finding the global optimal solutions. The described approaches are free from limitations of classic gradient optimization methods.

In the presented approach, shape, topology and material optimization is performed simultaneously for 3D freely vibrating structures analyzed by the finite element method. The optimal location of reinforcement is searched in order to increase a stiffness or strength of dynamically loaded reinforced plates analyzed by the coupled boundary/finite element method. As a result of optimization, a significant improvement of dynamic response is obtained, in comparison with initial designs.

Coupling of finite or boundary element method and the bio-inspired algorithms give an effective and efficient alternative optimization tool, which enables solving a large class of the optimization problems of mechanical structures. Numerical examples confirm the efficiency of the proposed optimization method and demonstrate that the methods based on soft computing are effective techniques for solving computer aided optimal design problems. Generally, for the considered numerical examples, the efficiency of the artificial immune systems and the particle swarm optimizers is better than the evolutionary algorithms.

Acknowledgments

This work was supported by the Polish Ministry of Science and Higher Education under the research grant.

REFERENCES

[1] J. Arabas, *Lectures on evolutionary algorithms*, WNT, Warszawa, 2001 (in Polish).

[2] J. Balthrop, F. Esponda, S. Forrest, M. Glickman, Coverage and Generalization in an Artificial Immune System. In Proceedings of the *Genetic and Evolutionary Computation Conference GECCO 2002*, 3-10, Morgan Kaufmann, New York, 2002.

[3] L. N de Castro, J. Timmis, Artificial Immune Systems as a Novel Soft Computing Paradigm, *Soft Computing*, **7**, 8, 526-544, 2003.

[4] L. N. de Castro, F. J. Von Zuben, Immune and Neural Network Models: Theoretical and Empirical Comparisons, *International Journal of Computational Intelligence and Applications (IJCIA)*, **1**, 3, 239-257, 2001.

[5] L. N. de Castro, F. J. Von Zuben, Learning and Optimization Using the Clonal Selection Principle, *IEEE Transactions on Evolutionary Computation, Special Issue on Artificial Immune Systems*, **6**, 3, 239-251, 2002.

[6] J. Dominguez, *Boundary elements in dynamics*, Computational Mechanics Publications, Elsevier Applied Science, Southampton-Boston, London-New York, 1993.

[7] J. Kennedy, RC. Eberhart, Particle Swarm Optimisation. *Proceedings of IEEE Int. Conf. on Neural Networks*, Piscataway, NJ, 1942-1948, 1995.

[8] P. Fedelinski, R. Gorski, Analysis and optimization of dynamically loaded reinforced plates by the coupled boundary and finite element method, *Computer Modeling in Engineering & Sciences*, **15**, 1, 31-40, 2006.

[9] F. Heppner, U. Grenander, *A stochastic nonlinear model for coordinated bird flocks*. In: Krasner S, editor. The Ubiquity of Chaos. Washington, DC: AAAS Publications, 1990.

[10] Z. Michalewicz, *Genetic algorithms + data structures = evolutionary algorithms*, Springer-Verlag, Berlin, 1996.

[11] M. Ptak, W. Ptak, *Basics of Immunology*, Jagiellonian University Press, Cracow, 2000 (in Polish).

[12] C. W. Reynolds, Flocks, herds, and schools, A distributed behavioral model. *Computer Graphics*, **21**, 25–34, 1987.

[13] S. T. Wierzchoń, *Artificial Immune Systems, theory and applications*, EXIT, 2001.