Swarm optimization of stiffeners locations in 2-D structures

Mirosław Szczepanik[1], **Wacław Kuś**[1] **and Tadeusz Burczyński**[1,2]

[1] *Department for Strength of Materials and Computational Mechanics Silesian University of Technology, Konarskiego 18a, 44-100 Gliwice, Poland*
e-mail: Miroslaw.Szczepanik@polsl.pl, Waclaw.Kus@polsl.pl

[2] *Cracow University of Technology, Institute of Computer Modelling, Artificial Intelligence Department Warszawska 24, 31-155 Cracow, Poland*
e-mail: Tadeusz.Burczynski@polsl.pl

Abstract

The paper is devoted to an application of the swarm methods and the finite element method to optimization of the stiffeners location in the 2-D structures (plane stress, bending plates and shells). The structures are optimized for the stress and displacement criteria. The numerical examples demonstrate that the method based on swarm computation is an effective technique for solving computer aided optimal design.

Keywords: swarm algorithms, optimization, finite element method (FEM), bars, plane stress, bending plates, shells

1. Introduction

The paper deals with an application of particle swarm optimiser (PSO) and the finite element method to the optimization problems of the 2-D structures in respect of stiffeners arrangement. Recently, swarm methods have found various applications in mechanics, and also in structural optimization. The swarm algorithms are based on the models of the animals social behaviours: moving and living in the groups. PSO 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 main advantage of the bio-inspired method is the fact that these approach do not need any information about the gradient of the fitness function and give a strong probability of finding the global optimum. The main drawback of these approaches is the long time of calculations. The fitness function is calculated for each swarm particle in each iteration by solving the boundary-value problem by means of the finite element method (FEM).

2. Formulation of the problem

Consider a 2-D structure (a plate in plane stress, a bending plate or a shell) which is stiffened by several bars. The domain of the 2-D structure and the domains of the bars are filled by a homogeneous and isotropic material of a Young's modulus E and a Poisson ratio ν. The location and shape of the bars can change for each iteration t of the swarm process. The stiffened structures are considered in the framework of the theory of elasticity. The swarm process proceeds in an environment in which the structure fitness is described by the minimization of the stress functional

$$J = \int_{\Omega} \psi(\sigma) d\Omega \tag{1}$$

where ψ is an arbitrary function of stress tensor σ, or maximization of the structure stiffness.

Two different types of optimization tasks are considered:

- optimization of the location of the straight stiffeners,
- optimization of the location and shape of curved stiffeners.

Following the two optimization tasks are described.

2.1. Optimization of the location of the straight stiffeners

The location of the stiffeners in the domain of 2-D structures is controlled by particle parameters h_i, i=1,…,N, which create a particle

$$par = [h_1, h_2, ..., h_i, ..., h_N], \qquad h^{\min} \le h_i \le h^{\max} \tag{2}$$

where

$h^{\min}$ - the minimum value of the particle parameter,

$h^{\max}$ - the maximum value of the particle parameter.

The connection of the stiffeners ends with the 2-D structures boundary has been assumed.

In order to reduce the number of the particle parameters, the particle representation, presented in the Fig. 1, has been introduced. So the number of particle parameters is twice bigger than number of the stiffeners.

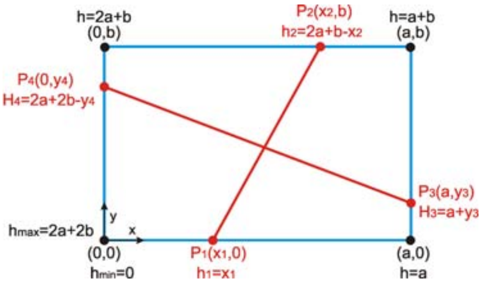


Figure 1: Particle representation of the stiffeners in 2-D structure geometry

2.2. Optimization of the location and shape of curved stiffeners

In order to minimize the number of parameters the curved stiffener is defined by means of NURBS curve (Non Uniform Rational B-Spline) [3]. The shape of this curve is defined by the control points C_k, $k=1,2,\ldots,L$; $C_k \subset \Omega_{2D}$ (L – number of control points). The connection of the stiffeners ends with the 2-D structures boundary has been assumed therefore beginning and end of the stiffener are defined by means of two points P (Fig. 2.).

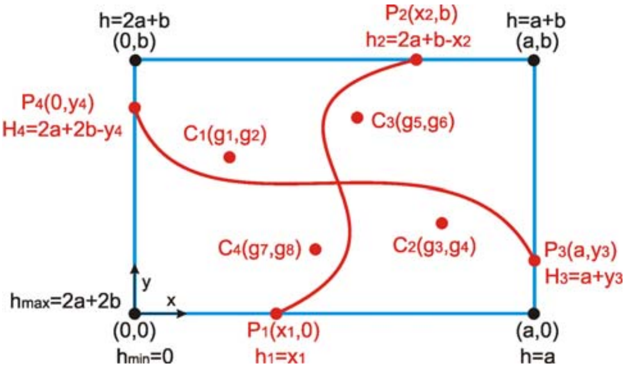


Figure 2: Particle representation of the curved stiffeners in 2-D structure geometry

The location of the stiffeners in the domain of 2-D structures is controlled by particle parameters h_i, i=1,…,N and the shape of them by particle parameters g_j, j=1,…,M. The set of the particle parameters creates a particle

$$par = [h_1, h_2, ..., h_i, ..., h_N, g_1, g_2, ..., g_i, ..., g_M] \tag{3}$$

$$h^{\min} \le h_i \le h^{\max}, \qquad g^{\min} \le g_j \le g^{\max}$$

where

$h^{\min}$ - the minimum value of the particle parameter h,

$h^{\max}$ - the maximum value of the particle parameter h,

$g^{\min}$ - the minimum value of the particle parameter g,

$g^{\max}$ - the maximum value of the particle parameter g.

In order to solve the formulated problems the finite element models of the structures are considered [7]. The 2-D structure domain Ω_{2D} is divided into triangular finite elements $\Omega_s, s = 1,2,...,R$ (for plane stress, bending plate or shell), according to the geometry mapped on the basis of the particle. The edges of the triangular finite elements which belong to the curves mapped on the basis of the particle and playing the role of the stiffeners, creates the bar elements $\Omega_b, b = R+1, R+2,...,C$ (Fig. 3).

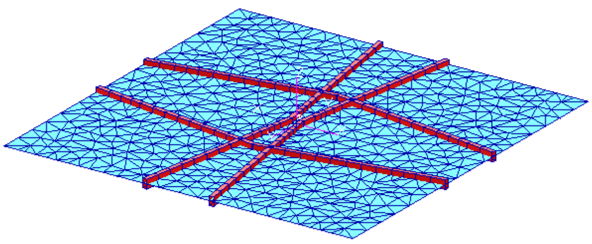


Figure 3: Mesh of 2-D and bar finite elements

After the geometry discretization, the finite element analysis is performed and node displacements are calculated by solving a system of linear algebraic equations

$$\mathbf{KU=F} \tag{4}$$

where **U** is a column matrix of unknown displacements, **F** is a known column matrix of acting forces and **K** is a known global stiffness matrix of the structure which elements are given as follows:

$$\mathbf{k_s} = \int_{\mathbf{V}} \mathbf{B_s^T D_s B_s dV}, \tag{5}$$

for 2-D structure elements, and

$$\mathbf{k_b} = \int_{l} \mathbf{B_b^T D_b B_b dV}, \tag{6}$$

for the bar elements,

where $\mathbf{D_s}$, $\mathbf{B_s}$ and $\mathbf{D_b}$, $\mathbf{B_b}$ are the known elasticity and geometrical matrices for the 2-D structure and bar elements, respectively, l represents the length of the bar element, **V** represents the volume of the finite element.

After the finite element analysis, the value of the fitness function given for example by:

$$J = \int_{\Omega_{2D}} \sigma_{eq} d\Omega_{2D} \tag{7}$$

is evaluated and the swarm algorithm is applied.

The formulation of the optimization task which assumes the possibility of the stiffeners intersection, causes some problems connected with impossibility of the proper discretization of the structures geometry mapped on the particles basis. The problems appear when the distance between the ends of two stiffeners or between the end of the stiffener and a corner of 2-D structure is too small. Then, the angles between the stiffeners and the boundary appear very small and the automatic mesh generator [5] has difficulties with creating the proper mesh and generates errors which cause breaks in the optimization program. Introduction of the additional constraints imposed on the particle parameters values is necessary. It was assumed that the distance between the ends of two stiffeners or between the end of the stiffener and the corner of the 2-D structure could not be less than the declared value. This constraint was applied in the case of second and third example presented in paragraph 5. Another possibility for solving the problem is improving of the geometry, mapped on the basis of the particle, by connecting the stiffeners ends when the distance between them is less than declared value. This constraint was applied in the case of the first example (paragraph 5). The problem can be easily solved by the introduction of the proper constraints, but it will be more complex in the case of the optimization task of many stiffeners locations. Then, many intersection points and many small angles between the stiffeners appear. The implementation of the very resistant mesh generator would be the best solution of the problem.

3. Particle Swarm Optimizer

The particle swarm algorithms, 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. 4).

a)  b) 

Figure 4: 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 by Craig and Raynolds [4]. 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 [1]. The results of this biological examination where used by Kennedy and Eberhart [2], 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. 5).

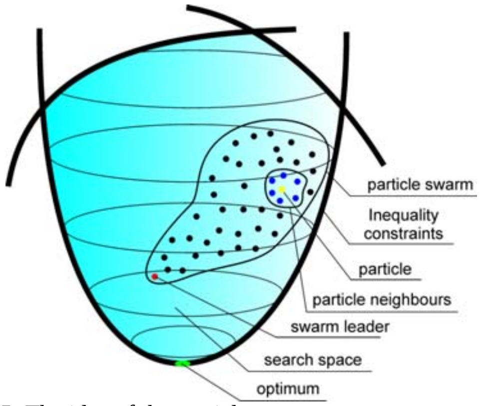


Figure 5: The idea of the particle swarm.

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:

$$v_{ij}(k+1) = wv_{ij}(k) + \phi_{1j}(k)\left[q_{ij}(k) - h_{ij}(k)\right] + \\ +\phi_{2j}(k)\left[\hat{q}_{ij}(k) - h_{ij}(k)\right] \tag{8}$$

$$h_{ij}(k+1) = h_{ij}(k) + v_{ij}(k+1), \quad i = 1,2,...,m\ ;\ j = 1,2,...,n \tag{9}$$

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],
$h_i(k)$ – 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 Eqn. 8. 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 $\phi_1(k)\left[q_i(k) - h_i(k)\right]$ 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 $\phi_2(k)\left[\hat{q}_i(k) - h_i(k)\right]$ 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. 6. 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 Eqn. 8 and particles positions are modified according to the Eqn. 9. The process is iteratively repeated until the stop condition is fulfilled. The stop condition is typically expressed as the maximum number of iterations.

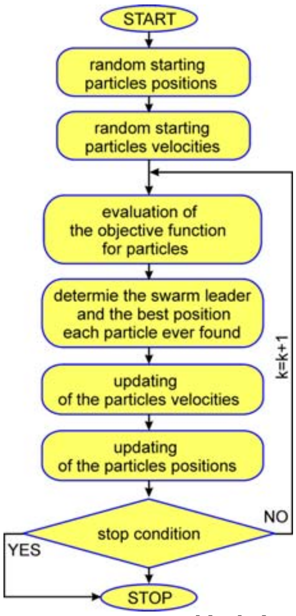


Figure 6: Particle swarm optimiser – block diagram.

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.

4. The coupling of the swarm algorithm and MSC NASTRAN

In order to calculate the fitness function value for a single particle, the boundary value problem for 2-D structure stiffened by the set of the bars has to be solved. To solve the boundary value problem the professional program of the finite element method MSC NASTRAN is applied. The coupling of the swarm optimization program and the finite element method is based on data transfer between both programs (Fig. 7). First the file containing input data to the optimization program (the structure of mesh of 2-D elements, the structure of mesh of the bar elements, the orientation of the bar elements, the thickness of 2-D structure, the moments of inertia for the stiffeners, boundary conditions, material data) is built. This file has a special structure, which can be read by MSC NASTRAN and is the basis on which the boundary value problem is solved. After the computations the MSC NASTRAN returns the result file from which the result data, necessary for calculation of the fitness function value (stresses, strains, displacements) are taken.

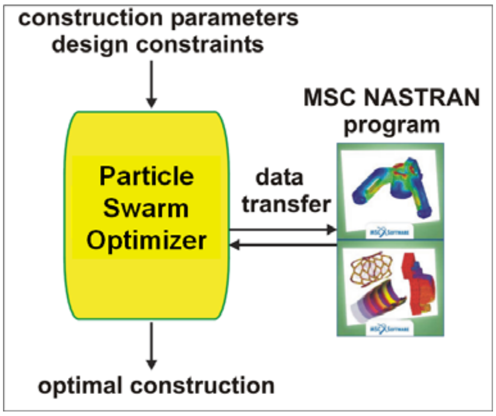


Figure 7: Coupling of the swarm algorithm and MSC NASTRAN

5. Examples of swarm optimization of structures

Four numerical examples of the optimization of the stiffeners location in geometry of 2-D structures are considered. Example 1 – the optimization of a plate in plane stress stiffened with 3 ribs, Example 2 – optimization of a bending plate stiffened with 4 ribs. Example 3 – the optimization of a shell structure stiffened with 5 ribs. Example 4 – the optimization of a plate in plane stress stiffened with 2 curved ribs. The domain of 2-D structures and domains of the bars in each example are filled by a homogeneous and isotropic material of a Young's modulus $E_0=2*10^5$ MPa and a Poisson ratio $\nu=0.3$. The value of the maximal stress $\sigma^{\max}=100MPa$. The stiffened structures are considered in the framework of the theory of elasticity. The results of the examples are obtained by use of optimization method based on swarm algorithm with parameters included in Table 1. The stiffeners in each of the numerical examples have rectangular cross-section of dimensions $w \times h$.

Table 1: Parameters of Particle Swarm Optimizer

Number of particles	20
Inertia weight w	0.73
acceleration coefficient c_1	1.47
acceleration coefficient c_2	1.47

5.1. *Example 1*

The optimization task of three stiffeners location by the minimization of the stress functional in a plate in plane stress with boundary conditions shown in the Fig. 8 is considered. Input data to the optimization program and the parameters of the swarm algorithm are included in Tab. 2 and 1, respectively. The results of the optimization process are presented in the Fig. 9.

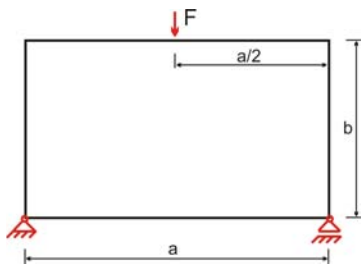


Figure 8: Geometry and boundary conditions for the plate in plane stress (example 1)

Table 2: Input data to the optimization program for example 1

a × b [mm]	F [N]	Number of stiffeners	Number of particle parameters	Rectangular cross-section of dimensions $w \times h$ [mm]	Thickness of the plate [mm]
400 × 600	300	3	6	10 × 20	8

a)

b)

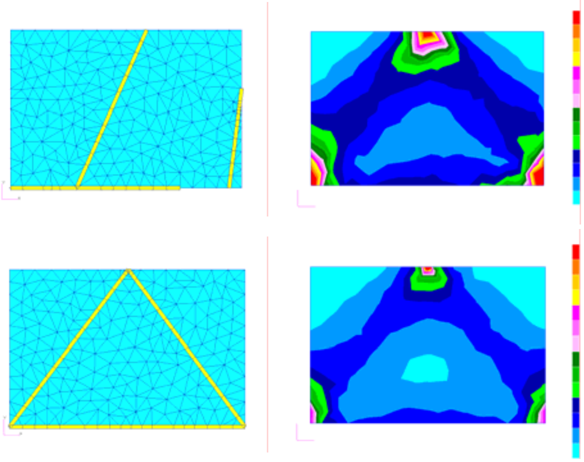


Figure 9: Location of three stiffeners in the plate in plane stress and the map of stresses; a) 1st iteration , b) 54th iteration

5.2. Example 2

The optimization task of four stiffeners location by the minimization of the stress functional in a bending plate loaded with the pressure *p* and fixed on the boundary (Fig. 10) is considered. Input data to the optimization program and the parameters of the swarm algorithm are included in Tab. 3 and 1, respectively. The results of the optimization process are presented in the Fig. 11.

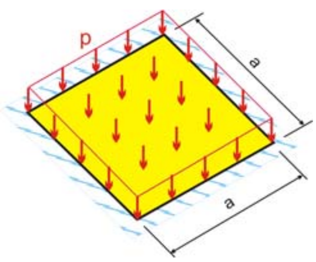


Figure 10: Geometry and boundary conditions for the bending plate (example 2)

Table 3: Input data to the optimization program for example 2

a × a [mm]	*p* [MPa]	Number of stiffeners	Number of particle parameters	Rectangular cross-section of dimensions $w \times h$ [mm]	Thickness of the plate [mm]
400 × 400	0.1	4	8	25 × 35	10

a)

b)

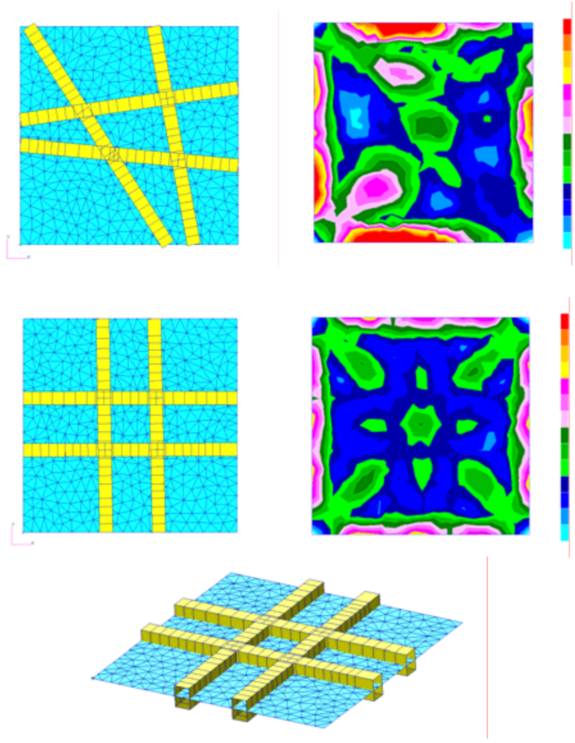


Figure 11: Location of four stiffeners in the bending plate and the map of stresses; a) 1st iteration, b) 527th iteration

5.3. Example 3

The optimization task of five stiffeners location by the minimization of the stress functional in a cylindrical shell is considered. The structure is stretched with continuous load *q* and is fixed as presented in the Fig. 12. Input data to the optimization program and the parameters of the swarm algorithm are included in Tab. 4 and 1, respectively. The results of the optimization process are presented in the Fig. 13.

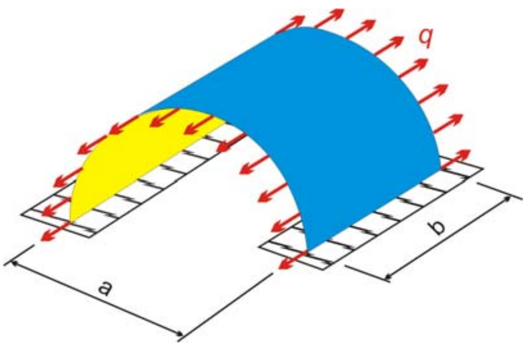


Figure 12: Geometry and boundary conditions for the cylindrical shell (example 3)

Table 4: Input data to the optimization program for example 3

a × b [mm]	*q* [N/mm]	Number of stiffeners	Number of particle parameters	Rectangular cross-section of dimensions *w* × *h* [mm]	Thickness of the plate [mm]
300 × 200	450	5	10	10 × 20	10

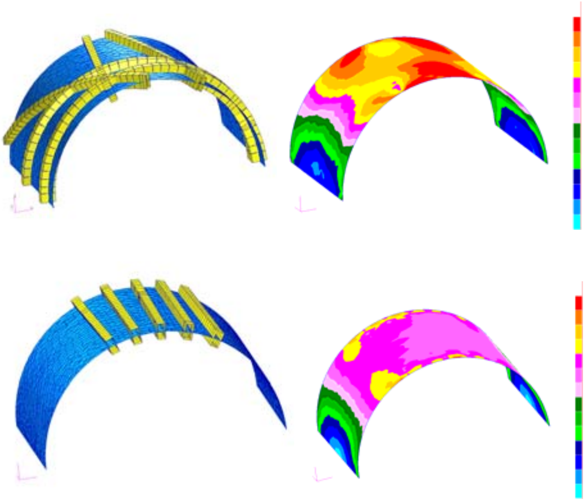


Fig. 13. Location of five stiffeners in the plate in plane stress and the map of stresses; a) 1st iteration , b) 186th iteration

5.4. *Example 4*

The optimization task of two stiffeners location and shape in a plate in plane stress with boundary conditions shown in the Fig. 14 is considered. The optimal positions of stiffeners are searched in order to maximize stiffness of the plate. The maximal nodal displacement in the structure is minimized. The stiffeners are modeled using 3-point NURBS curves. The value of weight of each control point is 1 (no influence on distance between the control point and the NURBS curve). Input data to the optimization program and the parameters of the swarm algorithm are included in Tab. 5 and 1, respectively. The results of the optimization process are presented in the Fig. 15.

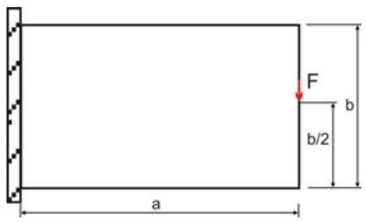


Figure 14: Geometry and boundary conditions for the plate in plane stress (example 4)

Table 5: Input data to the optimization program for example 4

a × b [mm]	F [N]	Number of stiffeners	Number of particle parameters	Rectangular cross-section of dimensions *w* × *h* [mm]	Thickness of the plate [mm]
400 × 600	1000	2	8	10 × 20	8

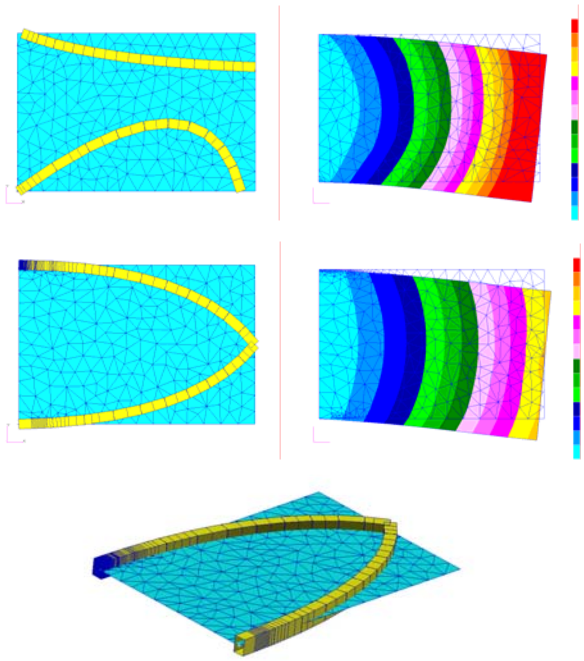


Figure 15: Location of two stiffeners in the plate in plane stress and the map of stresses; a) 1st iteration , b) 339th iteration

5.5. *Comparison of the effectiveness between PSO and DEA*

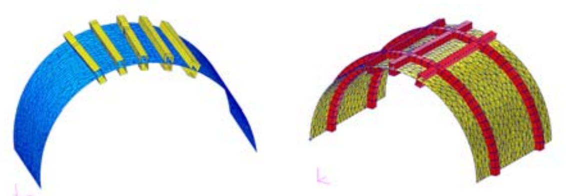
The main drawback of the bio-inspired approaches is the long time of calculations. So the choose of the effective method seems to be quite important. The comparison of the particle swarm optimiser (PSO) and distributed evolutionary algorithm (DEA) with parameters included in Tab. 6 has been made. The results of the comparison obtained for all the presented above numerical examples are included in the Tab. 7. The stiffeners arrangements obtained for examples 1,2 and 4 are consistent for both applied algorithms and different for example 3 (Fig. 16). Fitness function value for the result obtained using PSO is better.

Table 6. Parameters of distributed evolutionary algorithm

Number of subpopulations	2
Number of chromosomes in each subpopulation	10
Probability of Gaussian mutation	100%
Probability of simple crossover	100%
Selection method	rang selection

Table 7. The results of the comparison between PSO and DEA

	Example 1		Example 2		Example 3		Example 4	
	DEA	PSO	DEA	PSO	DEA	PSO	DEA	PSO
Fitness function value	25543	25480	406776	405119	1675702	1589670	0.001642	0.001640
Number of iterations	86	54	2746	527	463	186	1563	339
Number of individuals in each iteration	20	30	20	40	20	30	20	40
Number of fitness function evaluations	1720	1620	54920	21080	9260	5580	31260	13560



a) b)

Fig. 16. Location of five stiffeners in the plate in plane stress (Example 3) obtained using: a) PSO , b) DEA

6. Conclusions

An effective tool of swarm optimization of 2-D structures stiffened with several ribs is presented. Using this approach the optimal arrangement of the stiffeners in geometry of 2-D structures can be found. Implementing of the swarm algorithms to this approach gives a strong probability of finding the global optimal solutions. Described approach is free from limitations connected with classic gradient optimization methods referring to the continuity of the objective function, the gradient or hessian of the objective function and the substantial probability of getting a local optimum. Besides in the case of using gradient methods finding the global solution depends on the starting point. The swarm algorithm performs multidirectional optimum searching by exchanging information between particles and finding better and better particles positions. Comparison between PSO and DEA proves good effectiveness of particle swarm optimization method. Creating the finite element mesh for some locations of the stiffeners in the geometry of 2-D structures may be disadvantage. The problems grow when the task of the optimization of many stiffeners location is considered. So the very resistant mesh generator is necessary for more complex optimization problems.

References

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

[2] Kennedy, J. and Eberhart, R.C., *Swarm Intelligence*, Morgamn Kauffman, 2001.

[3] Piegel L, Tiller W, *The NURBS Book*, second ed., Springer-Verlag, Berlin, 1995.

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

[5] Shewchuk, J. R., Triangle: Engineering a 2D Quality Mesh Generator and Delaunay Triangulator, *First Workshop on Applied Computational Geometry, Association for Computing Machinery*, pp. 124-133, Philadelphia, Pennsylvania, USA, (1996).

[6] Tanese R, Distributed Genetic Algorithms. Proc. *3rd ICGA*, Ed. Schaffer JD, San Mateo, USA, pp. 434-439, 1989.

[7] Zienkiewicz, O.C. and Taylor, R.L., *The Finite Element Method*, Butterworth Heinemann, Oxford 2000.