Applied and Computational Mechanics 10 (2016) 15–26

Usage of the particle swarm optimization in problems of mechanics

M. Hajžman[a,*], R. Bulín[a], Z. Šika[b], P. Svatoš[b]

[a] European Centre of Excellence, NTIS – New Technologies for Information Society, Faculty of Applied Sciences, University of West Bohemia, Univerzitní 8, 306 14 Plzeň, Czech Republic

[b] Department of Mechanics, Biomechanics and Mechatronics, Faculty of Mechanical Engineering, Czech Technical University in Prague, Technická 4, 166 07 Praha, Czech Republic

Received 17 January 2016; received in revised form 23 June 2016

Abstract

This paper deals with the optimization method called particle swarm optimization and its usage in mechanics. Basic versions of the method is introduced and several improvements and modifications are applied for better convergence of the algorithms. The performance of the optimization algorithm implemented in an original in-house software is investigated by means of three basic and one complex problems of mechanics. The goal of the first problem is to find optimal parameters of a dynamic absorber of vibrations. The second problem is about the tunning of eigenfrequencies of beam bending vibrations. The third problem is to optimize parameters of a clamped beam with various segments. The last complex problem is the optimization of a tilting mechanism with multilevel control. The presented results show that the particle swarm optimization can be efficiently used in mechanical tasks.
© 2016 University of West Bohemia. All rights reserved.

Keywords: particle swarm optimization, dynamic absorber, beam eigenfrequencies, clamped beam, multilevel mechanism

1. Introduction

In modern mechanics an optimization process is very important and common part of designing dynamic systems and solving various problems. Different applications need various optimization algorithms, which should be sufficiently set in order to effectively solve a problem. Optimal parameters can make system less noisy, decrease amplitude of vibrations, make system cheaper etc., so they can be essential in project realization.

Standard approaches to the numerical optimization are described e.g. in [10] while popular genetic algorithms are employed e.g. in [17]. The disadvantages of these methods are related to searching for local optimum only or to requiring differentiable functions. It can be overcome by using methods of zero order, which do not use derivations of an objective function in optimization process. One of these methods is called *particle swarm optimization* (PSO), which has the ability to find global optimum of a given objective function. General review of various PSO algorithms is given in [6] and their usage in global optimization is presented in [11]. Constrained problems and imposing of nonlinear limit functions are studied by authors of [12]. Multiobjective optimization using PSO is shown in [16]. In statics and strength of solids, the PSO approaches are applied for example in shape optimization [7] or optimization of functionally graded structures [9]. Interesting applications in dynamics are shown in [15] for a control of maglev train levitation or in [5] for vibration isolation and control.

*Corresponding author. Tel.: +420 377 632 311, e-mail: mhajzman@kme.zcu.cz.

This paper is divided into 7 sections. After introduction the basic PSO algorithms are described together with their further improvements and with the verification of originally implemented software. The third section deals with the optimization of a vibration absorber. Tunning of beam eigenfrequencies is shown in the fourth section and the constrained optimization of a segmented clamped beam is described in the fifth section. The sixth section is devoted to a complex mechanical example of the optimization of a tilting multilevel mechanism.

2. Particle swarm optimization

The basic PSO algorithm was introduced by James Kennedy and Russel C. Eberhart in the year 1995 [8]. It is based on a behaviour of animals (such as sheep, birds or fish) in flock or swarm. Each individual in a swarm cooperates with others to find areas, where can be enough food or something. All individuals in a swarm create a social behaviour, which leads to finding optimum areas in a search space.

The PSO algorithm uses a *swarm* of size n_s, which is made by individual *particles* represented by vector $\mathbf{x}_i(t) = [x_{i1}, x_{i2}, \ldots, x_{in}]$, where n is the dimension of a search space, $i = 1, 2, \ldots, n_s$ is the index of a particle and t is a discrete time with a constant step equal to 1. The position of particle is changing due to vector of velocity $\mathbf{v}_i(t) = [v_{i1}, v_{i2}, \ldots, v_{in}]$ according to equation [4]

$$\mathbf{x}_i(t+1) = \mathbf{x}_i(t) + \mathbf{v}_i(t+1). \tag{1}$$

The initial position of particle $\mathbf{x}_i(0)$ in the search space is random and is generated by the uniform distribution. The initial velocity $\mathbf{v}_i(0)$ is set to zero vector.

Each particle has a memory, where its best position found during the optimization is stored. This position is called *personal best* or $pbest$, is represented by vector $\mathbf{y}_i(t) = [y_{i1}, y_{i2}, \ldots, y_{in}]$ and is calculated as

$$\mathbf{y}_i(t+1) = \begin{cases} \mathbf{y}_i(t) & \text{for } f(\mathbf{x}_i(t+1)) \geq f(\mathbf{y}_i(t)), \\ \mathbf{x}_i(t+1) & \text{for } f(\mathbf{x}_i(t+1)) < f(\mathbf{y}_i(t)), \end{cases} \tag{2}$$

where $f\colon \mathbb{R}^n \to \mathbb{R}$ is an objective function, which determines the quality of a particle position.

Particles can cooperate with others and they form *neighbourhoods*. Inside a neighbourhood the particles, which belongs to this neighbourhood, share information about their $pbest$ positions. The best position of all $pbest$ positions in neighbourhood is called *local best* or $lbest$. The velocity of a particle is based on the difference between an actual position and $pbest$ and $lbest$ positions. There are two basic versions of the PSO algorithm, which differ in neighbourhood size and structure. The first one is called *Global Best PSO* and the second one is called *Local Best PSO*.

2.1. Global Best PSO

In this version the neighbourhood is made by the whole swarm. It means, that all particles cooperate with others and there is only one $lbest$ position, which is called *global best* or $gbest$ and is represented by vector $\hat{\mathbf{y}}(t)$. The $gbest$ position is defined as

$$\hat{\mathbf{y}}(t) \in \{\mathbf{y}_1(t), \ldots, \mathbf{y}_{n_s}(t)\} | f(\hat{\mathbf{y}}(t)) = \min\{f(\mathbf{y}_1(t)), \ldots, f(\mathbf{y}_{n_s}(t))\}. \tag{3}$$

In the global best version the velocity in time step $t+1$ is calculated by equation

$$v_{ij}(t+1) = v_{ij}(t) + c_1 r_{1j}(t)[y_{ij}(t) - x_{ij}(t)] + c_2 r_{2j}(t)[\hat{y}_j(t) - x_{ij}(t)], \tag{4}$$

where $v_{ij}(t)$ is velocity of particle i in direction j ($j = 1, 2, \ldots, n$) in time step t, c_1 and c_2 are positive constants which indicate how much is the particle attracted by its $pbest$ position and $gbest$ position, $r_{1j}(t)$ and $r_{2j}(t)$ are random values from $\langle 0, 1\rangle$ generated by the uniform distribution and representing a stochastic element of the algorithm.

2.2. *Local Best PSO*

In this version the neighbourhood is created by a chosen set of particles. The easiest way, how to create the neighbourhood, is by indexes to vector $\mathbf{x}$. The neighbourhood $\mathcal{N}_i$ of particle i is defined as

$$\mathcal{N}_i = \{\mathbf{y}_{i-n_{\mathcal{N}_i}}(t), \mathbf{y}_{i-n_{\mathcal{N}_i}+1}(t), \ldots, \mathbf{y}_{i-1}(t), \mathbf{y}_i(t), \mathbf{y}_{i+1}(t), \ldots, \mathbf{y}_{i+n_{\mathcal{N}_i}}(t)\}, \tag{5}$$

where $n_{\mathcal{N}_i}$ is the size of the neighbourhood. The $lbest$ $\hat{\mathbf{y}}_i(t)$ position of neighbourhood i is defined as

$$\hat{\mathbf{y}}_i(t+1) \in \{\mathcal{N}_i | f(\hat{\mathbf{y}}_i(t+1)) = \min\{f(\mathbf{x})\}, \forall \mathbf{x} \in \mathcal{N}_i\}. \tag{6}$$

The velocity is calculated by

$$v_{ij}(t+1) = v_{ij}(t) + c_1 r_{1j}(t)[y_{ij}(t) - x_{ij}(t)] + c_2 r_{2j}(t)[\hat{y}_{ij}(t) - x_{ij}(t)]. \tag{7}$$

The main advantage of this version is that the particles can explore the search space better and can find better minimum than particles in global best version. The main disadvantage is that it can take longer time to find satisfactory results. For less complex optimization problems, the global best version is faster than local best version because of lower computational complexity.

2.3. *Improvements and modifications of the basic PSO algorithm*

The basic version of PSO usually does not provide satisfactory convergence to optimum. That is why some improvements were applied. The goal of these modifications is to give particles appropriate ration between so called *exploration* and *exploitation* ability. The good exploration ability means, that the particles cover the whole search space during their life. This helps to find optimum areas in the search space, but the particles are unable to focus on a global optimum. On the other hand, the exploitation ability represents the effort of particles to concentrate on a small area from the search space and find there the best solution. It is necessary to find an appropriate balance between these two abilities. In the first half of the optimization process the particles should have the good exploration ability and than the exploitation ability. Improvements will be explained on the Local Best version of PSO.

The first modification is the introduction of an inertia weight. Inertia weight $\omega(t)$ indicates, how much velocity in step $t+1$ depends on velocity in step t. Equation (7) should be changed to

$$v_{ij}(t+1) = \omega(t) v_{ij}(t) + c_1 r_{1j}(t)[y_{ij}(t) - x_{ij}(t)] + c_2 r_{2j}[\hat{y}_j(t) - x_{ij}(t)]. \tag{8}$$

The inertia weight decreases linearly with each time step (for example from $\omega(0) = 0.9$ to $\omega(n_t) = 0.1$, where n_t is a maximum number of iterations). This modification grants better exploitation of the algorithm during time steps and helps the particles to focus on accuracy of global best position.

Next modification is called *velocity clamping*. The velocity can explode into large values in few steps and the particles diverge from the search space. This can be solved by expression

$$v_{ij}(t+1) = \begin{cases} v'_{ij}(t+1), & v'_{ij}(t+1) < V_{\max,j}, \\ V_{\max,j}, & v'_{ij}(t+1) \geq V_{\max,j}, \end{cases} \tag{9}$$

where v'_{ij} is calculated using equations (4) or (7) and $V_{\max,j}$ is a problem dependent parameter. The initial $V_{\max,j}$ is chosen by equation

$$V_{\max,j} = \delta(x_{\max,j} - x_{\min,j}), \tag{10}$$

where $x_{\max,j}$ and $x_{\min,j}$ are maximum and minimum values of j component of variable $\mathbf{x}$, $\delta \in (0, 1\rangle$ is a problem dependent constant. In this work, exponentially decreasing $V_{\max,j}$ is used according to the equation

$$V_{\max,j}(t+1) = (1 - (t/n_t)^{\alpha})V_{\max,j}(t), \tag{11}$$

where α is another problem dependent constant, which can be found by testing the algorithm and n_t is a maximal number of iterations. Additional correction for velocity can be used based on

$$v_{ij}(t+1) = V_{\max,j}\mathrm{tgh}\left(\frac{v'_{ij}(t+1)}{V_{\max,j}}\right). \tag{12}$$

More about this correction can be found in [4].

Last improvement is related to parameters c_1 and c_2, which can change during algorithm according to

$$c_1(t) = (c_{1,\min} - c_{1,\max})\frac{t}{n_t} + c_{1,\max}, \qquad c_2(t) = (c_{2,\max} - c_{2,\min})\frac{t}{n_t} + c_{2,\min}, \tag{13}$$

where e.g. $c_{1,\min} = c_{2,\min} = 0.5$ and $c_{1,\max} = c_{2,\max} = 2.5$. This will cause, that at the beginning of the algorithm particles are more attracted by $pbest$ position and at the end of the algorithm are particles more attracted by $lbest$.

2.4. Benchmark example

The Levy No. 5 function (see Fig. 1) defined on search space $\langle -10, 10\rangle \times \langle -10, 10\rangle$ as

$$f(x,y) = \sum_{i=1}^{5}[i\cos((i-1)x+i)]\sum_{j=1}^{5}[j\cos((j+1)y+j)] + (x+1.425\,13)^2 + (y+0.800\,32)^2 \tag{14}$$

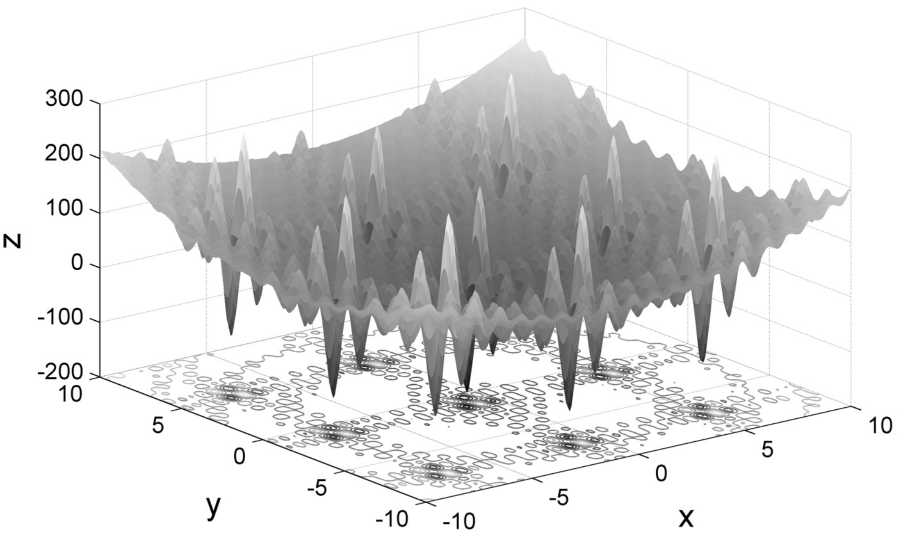

Fig. 1. Levy No. 5. function

was used for testing the algorithm. This function has many local minima but only one global minimum $f(-1.306\,8, -1.424\,8) = -176.137\,5$ and it is designed as the benchmark for testing of various algorithms. The goal was to find proper algorithm constants, which guarantee, that the PSO method find global minimum with probability at least 99 percents. The algorithm was launched many times and proper constants were found. The local best algorithm was very effective with maximum number of iterations set to $n_t = 100$, swarm size $n_s = 20$, neighbourhood radius $n = 2$, constant $\delta = 0.9$ and constant $\alpha = 5$. Figs. 2 and 3 show, that all particles converge to one solution.

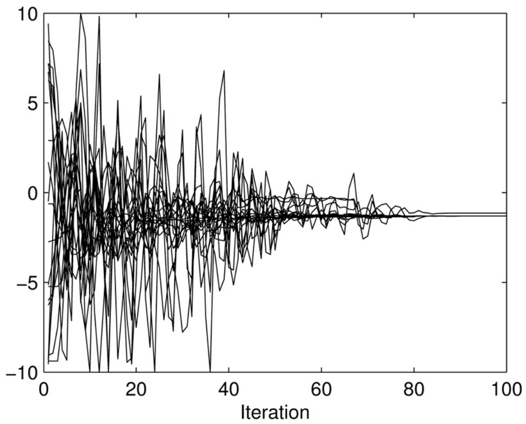

Fig. 2. x values of particles in iterations

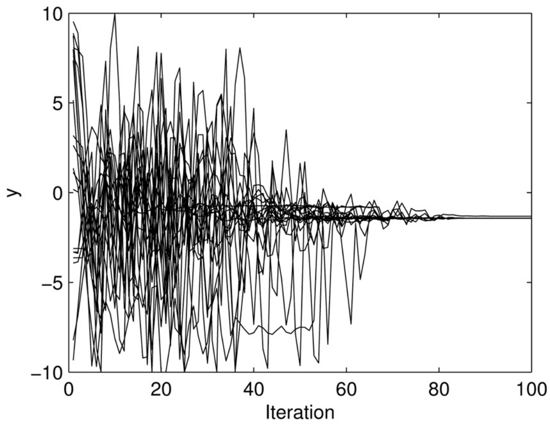

Fig. 3. y values of particles in iterations

3. Tuning of the dynamic vibration absorber model

The dynamic absorber of vibrations (Fig. 4) consists of matter m_3 attached to mass m_2 using spring of stiffness k_3 and damper of coefficient b_3. Optimization parameters are $\mathbf{x} = [m_3, b_3, k_3]$ with lower barrier $\mathbf{x}_d = [0.1, 10, 10^3]$ and upper barrier $\mathbf{x}_h = [2, 500, 10^5]$. The mechanical system is excited by force

$$f_2 = F_2 \sin \omega t. \tag{15}$$

The mechanical system can be described by equation

$$\mathbf{M}\ddot{\mathbf{q}}(t) + \mathbf{B}\dot{\mathbf{q}}(t) + \mathbf{K}\mathbf{q}(t) = \mathbf{f}(t), \tag{16}$$

where $\mathbf{M}$ is mass matrix, $\mathbf{B}$ is damping matrix and $\mathbf{K}$ is stiffness matrix and they can be formed as

$$\mathbf{M} = \begin{bmatrix} m_1 & 0 & 0 \\ 0 & m_2 & 0 \\ 0 & 0 & m_3 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} b_1 & 0 & 0 \\ 0 & b_3 & -b_3 \\ 0 & -b_3 & b_3 \end{bmatrix}, \quad \mathbf{K} = \begin{bmatrix} k_1 + k_2 & -k_2 & 0 \\ -k_2 & k_2 + k_3 & -k_3 \\ 0 & -k_3 & k_3 \end{bmatrix}. \tag{17}$$

The chosen parameters of mechanical model are summarized in Table 1.

The amplitude $\mathbf{q}$ of steady dynamics response is defined as

$$\tilde{\mathbf{f}} = \begin{bmatrix} 0 \\ -\mathrm{i}F_2 \\ 0 \end{bmatrix}, \qquad \mathbf{Z} = -\omega_b^2\mathbf{M} + \mathrm{i}\omega_b\mathbf{B} + \mathbf{K}, \qquad \tilde{\mathbf{q}} = \mathbf{Z}^{-1}\tilde{\mathbf{f}}, \qquad \mathbf{q} = |\tilde{\mathbf{q}}|. \tag{18}$$

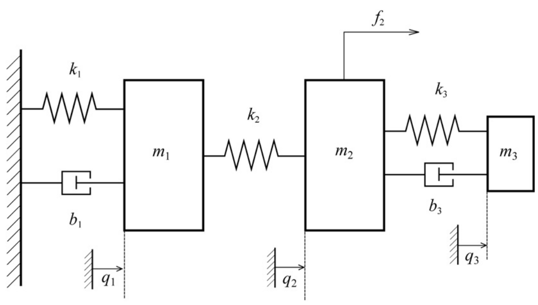

Fig. 4. The mechanical system with dynamic absorber of vibrations

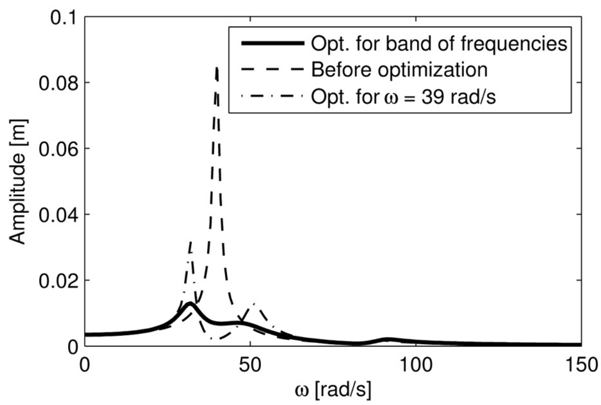

Fig. 5. Steady response q_2 before and after optimization with PSO algorithm

Table 1. Known parameters of mechanical model

$m_1 = 10$ kg	$k_1 = 5 \cdot 10^4$ Nm^{-1}	$b_1 = 100$ Nsm^{-1}
$m_2 = 7$ kg	$k_2 = 2 \cdot 10^4$ Nm^{-1}	$F_2 = 50$ N
$m_3 = 1$ kg	$k_3 = 10^4$ Nm^{-1}	$b_3 = 100$ Nsm^{-1}

The first task is to minimize amplitude of steady response q_2 of mass m_2 for constant excitation frequency $\omega = 39\,\text{rad} \cdot \text{s}^{-1}$, which is related to one of original eigenfrequencies. The objective function is $f(\mathbf{x}) = q_2$ The optimization process based on the PSO algorithm found optimum parameters as $\mathbf{x}^* = [m_3,\ b_3, k_3] = [2, 10, 3.057 \cdot 10^3]$. The algorithm with a swarm that contains $n_s = 60$ particles and neighbourhood radius $n = 10$ shows good convergence in about 150 time steps.

The second task is to minimize average amplitude of steady dynamics response q_2 for band of frequencies $\langle 0, 150 \rangle$. The objective function is $f(\mathbf{x}) = \frac{1}{150} \sum_{\omega_b=0}^{150} q_{2,\omega_b}$ for frequencies $\omega_b = 0, 1, 2, \ldots, 150$ rad $\cdot$ s^{-1}. PSO algorithm found parameters $\mathbf{x}^* = [2, 34.764, 2.767 \cdot 10^3]$. Very good convergence was reached with about $n_s = 30$ particles, neighbourhood radius $n = 10$ and in about 100 time steps. The resulting steady response with comparison to the original one is shown in Fig. 5.

4. Tuning of beam eigenfrequencies

Another studied mechanical system was a supported beam of length $l = 1$ m (Fig. 6). The parameters of the beam are modulus of elasticity $E = 2.1 \cdot 10^5$ MPa, mass density $\rho = 7\,800$ kg $\cdot$ m^{-3}, height a [m], width b [m], cross-section area $A = ab$ and moment of

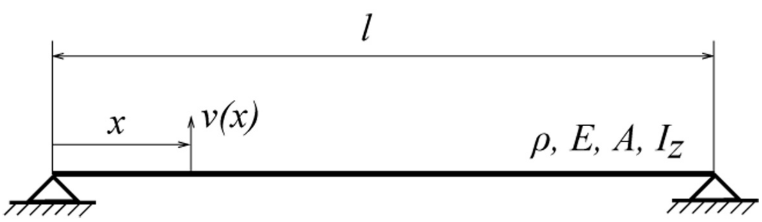

Fig. 6. Supported beam and its parameters

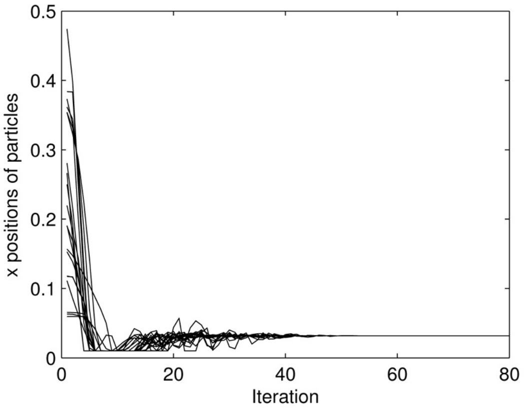

Fig. 7. Values of height a of particles in iterations

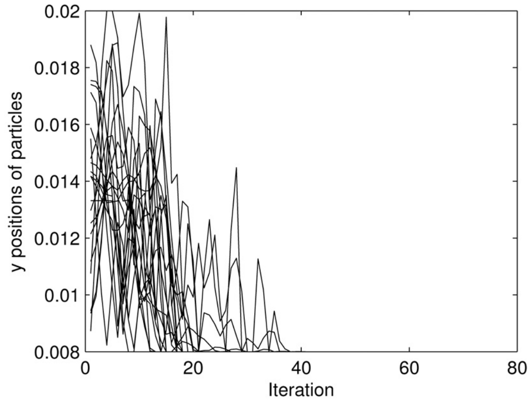

Fig. 8. Values of width b of particles in iterations

inertia $I_z = ba^3/12$. The equation for eigenfrequency Ω_j in [Hz] is [3]

$$\Omega_j = \sqrt{\frac{j^4\pi^4 E I_z}{2\pi l^4 \rho A}}, \quad j = 1, 2, \ldots, N. \tag{19}$$

The task is to tune the second eigenfrequency to $\bar{\Omega}_2 = 300$ Hz with the constraint that the weight of te beam cannot exceed $m_0 = 2$ kg. The objective function is formed as

$$f(\mathbf{x}) = \left(1 - \frac{\Omega_2(\mathbf{x})}{\bar{\Omega}_2}\right)^2 + p(m), \qquad p(m) = \begin{cases} 0 & \text{for } m \le m_0, \\ 10^6 & \text{for } m > m_0. \end{cases} \tag{20}$$

Optimization parameters are $\mathbf{x} = [x,\ y] = [a, b]$, lower limit $\mathbf{x}_d = [0.01, 0.008]$, upper limit $\mathbf{x}_u = [0.05, 0.02]$. The PSO algorithm has easily optimized this discontinuous objective function using only $n_s = 20$ particles with neighbourhood radius $n = 4$ in 80 time steps. The optimized parameters are $\mathbf{x}^* = [0.031\,88, 0.008]$. Fig. 7 and Fig. 8 show the motion of particles in each direction of the search space.

5. Optimization of the beam with variable cross-section

The clamped beam is divided into five segments (see Fig. 9). The length of the beam is $l = 5$ m and each segment is $l_s = 1$ m long. The beam end is loaded by force $F = 50$ kN. The

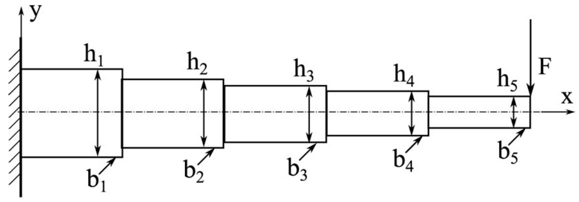

Fig. 9. Clamped beam with segment dimensions

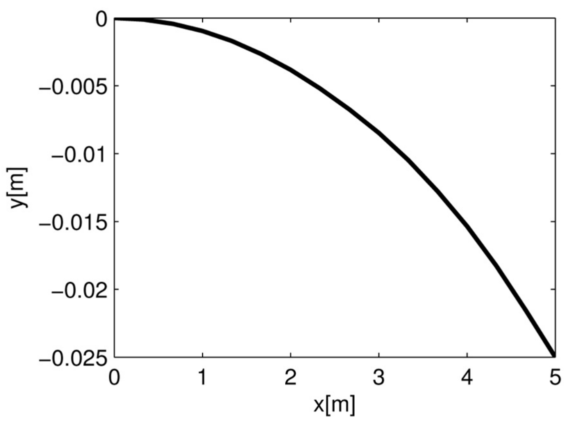

Fig. 10. Displacement of the beam

modulus of elasticity is $E = 2.1 \cdot 10^5$ MPa, mass density $\rho = 7\,800$ kg $\cdot$ m^{-3} and Poisson constant $\nu = 0.3$. The goal of this optimization is to minimize weight considering not exceeding maximum displacement $y_{\max} = -0.05$ m at the beam end. Due to known parameters, this problem can be solved by finding optimal height h_i and width b_i (where $i = 1, \ldots, 5$). The second restriction is on the ratio between h_i and b_i of each segment. Equation $h_i/b_i \leq 20$ must be fulfilled. Vector of parameters is $\mathbf{p} = [h_1, b_1, h_2, b_2, \ldots, h_5, b_5]$, the lower barrier for heights is set to $h_{i,\min} = 0.05$ m, lower barrier for widths is set to $b_{i,\min} = 0.01$ m, upper barrier for both heights and widths is set to $h_{i,\max} = b_{i,\max} = 1$ m.

The objective function is

$$f(\mathbf{p}) = \mathbf{h}^T\mathbf{b} + p_1 + p_2, \tag{21}$$

where $\mathbf{h} = [h_1, \ldots, h_5]^T$ is vector of heights, $\mathbf{b} = [b_1, \ldots, b_5]^T$ is vector of widths, p_1 and p_2 are penalty functions. The displacement y of the beam is calculated using the finite element method. The beam is divided into fifteen finite elements with three finite elements per each segment. Penalty function p_1 is formulated as

$$p_1 = \begin{cases} 0 & \text{for } y \geq y_{\max}, \\ 10^3 & \text{for } y < y_{\max} \end{cases} \tag{22}$$

and represents the condition of maximum displacement y. When the maximum displacement is exceeded, the functional value of the penalty function raises by 10^3. The second penalty function p_2 is

$$p_2 = \begin{cases} 0 & \text{for } h_i/b_i \leq 20, \\ 10^3 & \text{for } h_i/b_i > 20 \end{cases} \tag{23}$$

and it represents the condition to the ration between h_i and b_i of each segment. When $h_i/b_i > 20$ the functional value of the penalty function raises by 10^3. The objective function is therefore discontinuous.

The PSO algorithm found optimized parameters as

$$\mathbf{p}^* = [0.617\,4, 0.030\,9, 0.574\,6, 0.028\,7, 0.547\,4, 0.028\,0, 0.430\,2, 0.021\,6, 0.333\,4, 0.016\,7], \tag{24}$$

the value of the objective function with optimized parameters is $f(\mathbf{p}^*) = 0.06577$ and the maximum displacement $y = -0.025$ m. These results are comparable with results published in [1]. The PSO algorithm used $n_s = 100$ particles with neighborhood radius $n = 10$ in 200 time steps.

6. Optimization of the controllability of a tilting lightweight multi-level mechanism

The idea behind this optimization example is the hierarchical motion control of lightweight multi-level mechanisms consisting of the large motion level realized by cable driven parallel mechanisms [2] and the small motion levels realized using active structures [13]. The final goal of the optimization and control of multi-level structures is the widening of frequency bandwidth of their feedback motion control. The superimposed active structures can improve the positioning accuracy and operational speed of the end-effectors of recently developed cable driven parallel mechanisms of different types.

Particularly the spherical redundantly actuated tilting mechanism is considered. It is primarily actuated by four fibres and performs a spherical motion with three degrees of freedom [14]. The four fibres lead over pulleys to sliders with servo-drives (see Fig. 11). Active structure with six piezoactuators [13] is integrated between the end-effector platform and auxiliary platform

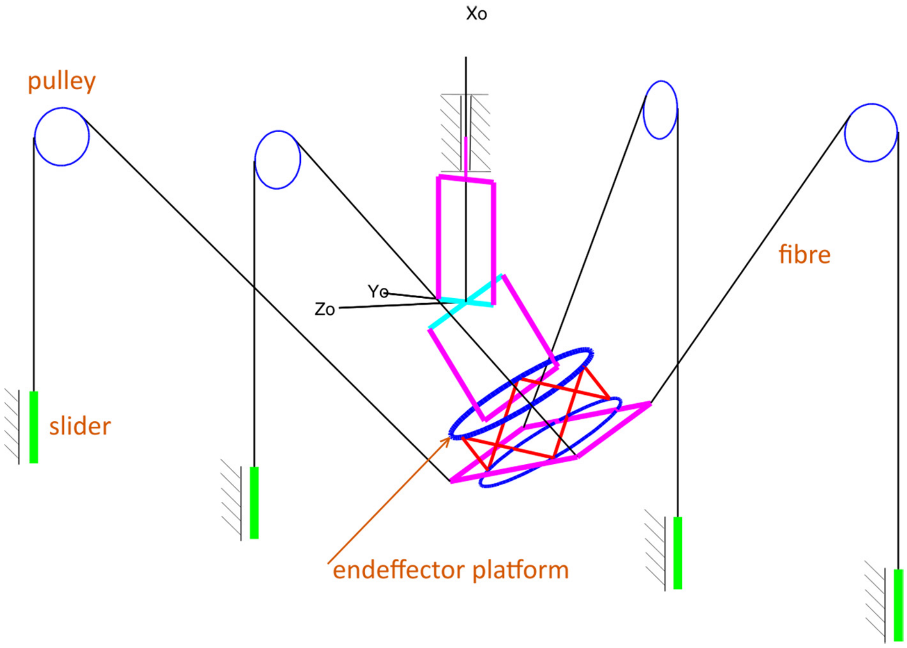

Fig. 11. Scheme of the 3 DOF tilting fibre mechanism with the 6 DOF active structure with piezoactuators

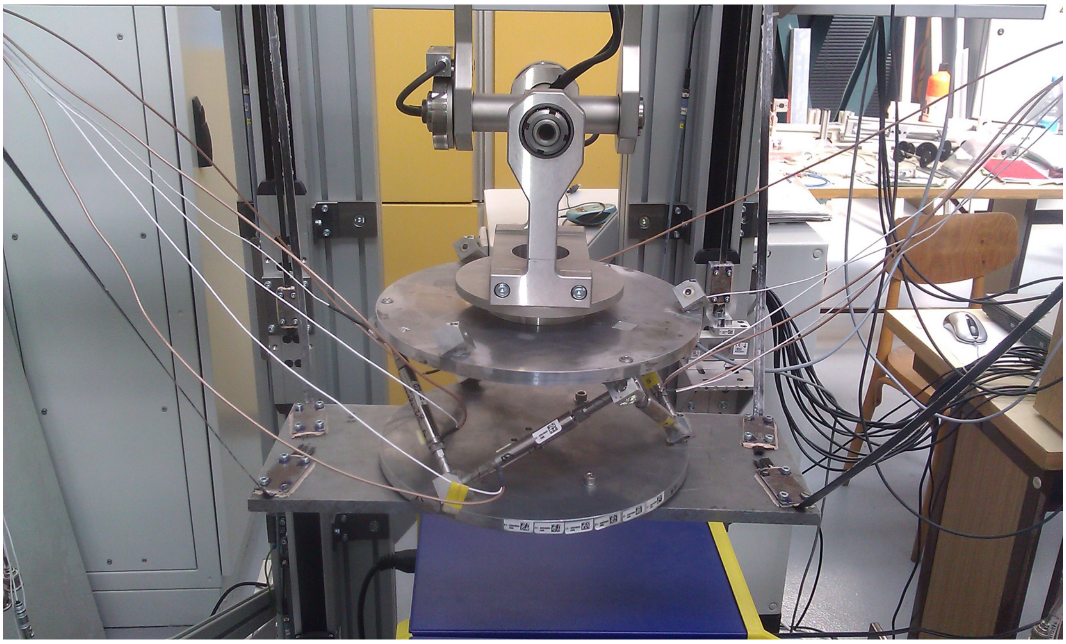

Fig. 12. Functional model of the 3 DOF tilting fibre mechanism with the 6 DOF active structure with piezoactuators

suspended and moved by four fibres. The piezoactuators are mounted in the cubic configuration (Fig. 12). The central spherical joint (Cardan universal joint) equipped with rotational incremental sensors simultaneously ensures the measurement of the end-effector platform position. The target of the optimization is the maximization of the end-effector platform controllability using all available actuators. The large lower frequency components of motion are controlled by the servo-drives through the fibres, whereas the small high frequency components of motion are controlled by the piezoactuators.

The optimization process should help to tune chosen mechanical properties of the system. Namely the stiffness of piezoactuators and dimensions of the additional body are considered. The cylindrical body is situated in the middle of the platform and described by its diameter and height. Parameters of the optimization are stiffness of the piezoactuators denoted as p_1 in range $\langle 4 \cdot 10^6, 1.8 \cdot 10^9 \rangle$ [N/m], diameter of the additional weight p_2 in range $\langle 0.06, 0.20 \rangle$ [m] and height of the additional weight p_3 $\langle 0.01, 0.1 \rangle$ [m].

The cost function $C(\boldsymbol{p})$ is defined as the condition number (conditionality) of controllability gramian W_c

$$C(\boldsymbol{p}) = \mathrm{cond}\, W_c = \mathrm{cond} \left(\int_0^\infty \mathrm{e}^{\mathbf{A}\tau} \mathbf{B} \mathbf{B}^T \mathrm{e}^{\mathbf{A}^T \tau} \, \mathrm{d}\tau \right), \tag{25}$$

where $\mathbf{B} = \mathbf{B}(\boldsymbol{p})$ and $\mathbf{A} = \mathbf{A}(\boldsymbol{p})$ are matrices of the state space model of the linearized system and depend on the optimization parameters. The global best optimization algorithm was employed for the minimization of the objective function formed as

$$f(\boldsymbol{p}) = (1 - C(\boldsymbol{p}))^2 \tag{26}$$

and thus for the maximization of the controllability.

During the optimization, it was observed that parameter p_1 (stiffness of piezoactuators) always reaches its lower boundary, while other two parameters converged to different local extremes (Fig. 13). The contours of the criterion ($C(\boldsymbol{p})$ function) in dependence on the end-effector diameters for boundary value of p_1 are shown in Fig. 14, where the big black dots are representing the positions of particles in the last iteration. It can be seen that the particles reached several different points, which are very close to local extremes. Fig. 14 also shows that the objective function is very complex and therefore the algorithm for actual setting were not able to find global optimum.

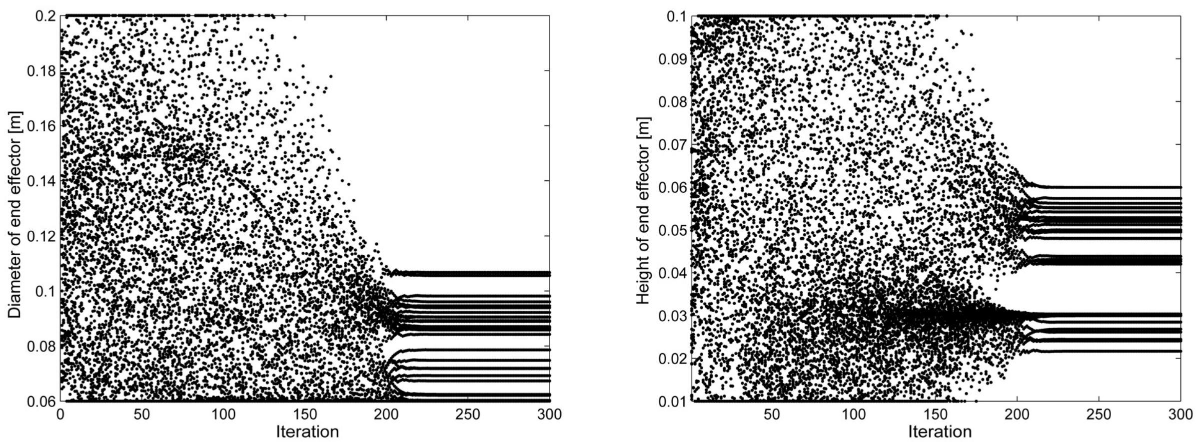

Fig. 13. Iteration history of the height and the diameter of the end-effector

7. Conclusions

The particle swarm optimization algoritm was presented in this paper. Several modifications and improvements of the basic algorithm were described and it was the basis for the creating of the in-house implementation in the MATLAB system. This implementation was further successfully verified using typical example of the very complex Levy No. 5 function.

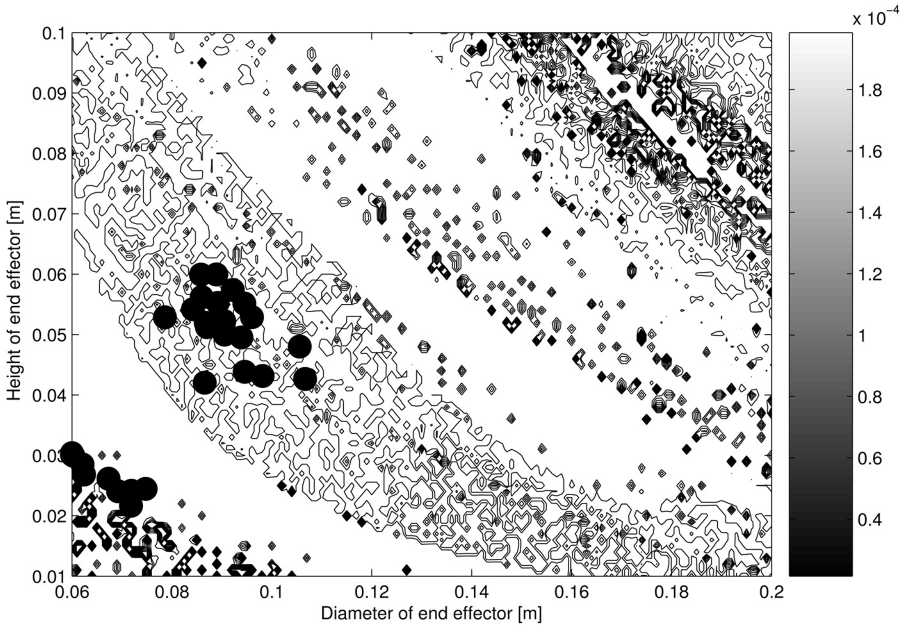

Fig. 14. Countours of the criterion (controllability) with depicted particles

The main part of the paper was aimed at the algorithm testing on various examples from mechanics. Three basic problems and one very complex problem of multilevel mechanism were introduced. The PSO method has the ability to find global optimum of objective functions and it can also optimize nondifferentiable functions. However, the PSO method requires more objective function evaluations than gradient based methods, but in some problems it can show better results in comparable times. The method is efficient in many optimization problems in dynamics, where the global optimum is sought.

Acknowledgements

The paper has originated in the framework of solving No. 15-20134S project of the Czech Science Foundation entitled "Multi-level Light Mechanisms with Active Structures".

References

[1] Abolbashari, M. H., Majdi, M., Mahpeykar, M. R., Structural optimization using optimizer program, Proceedings of the 3rd European Conference on Computational Mechanics, Lisbon, Instituto Superior Técnico, 2006, pp. 709.

[2] Duan, X., Qiu, Y., Duan, Q., Du, J., Calibration and motion control of a cable-driven parallel manipulator based triple-level spatial positioner, Advances in Mechanical Engineering, 2014, ID 368018.

[3] Dupal, J., Computational methods of mechanics, Lecture notes, University of West Bohemia, Pilsen, 1999. (in Czech)

[4] Engelbrecht, A. P., Fundamentals of computational swarm intelligence, John Wiley & Sons, Chichester, 2005.

[5] Farshidianfar, A., Saghfi, A., Kalami, S. M., Saghafi, I., Active vibration isolation of machinery and sensitive equipment using H_∞ control criterion and particle swarm optimization method, Meccanica 47 (2012) 437–453.

[6] Jordehi, A. Z., Particle swarm optimisation for dynamic optimisation problems: A review, Neural Computing and Applications 25 (7) (2014) 1507–1516.

[7] Kaveh, A., Javadi, S. M., Shape and size optimization of trusses with multiple frequency constraints using harmony search and ray optimizer for enhancing the particle swarm optimization algorithm, Acta Mechanica 225 (2014) 1595–1605.

[8] Kennedy, J., Eberhart, R., Particle swarm optimization, Proceedings of the IEEE International Conference on Neural Networks, Perth, IEEE, 1995, vol. 4, pp. 1942–1948.

[9] Loja, M. A. R., On the use of particle swarm optimization to maximize bending stiffness of functionally graded structures, Journal of Symbolic Computation 61–62 (2014) 12–30.

[10] Nocedal, J., Wright, S. J., Numerical optimization, Springer, New York, 2006.

[11] Parsopoulos, K. E., Vrahatis, M. N., Recent approaches to global optimization problems through Particle Swarm Optimization, Natural Computing 1 (2002) 235–306.

[12] Sedlaczek, K., Eberhard, P., Using augmented Lagrangian particle swarm optimization for constrained problems in engineering, Structural and Multidisciplinary Optimization 32 (2006) 277–286.

[13] Svatoš, P., Šika, Z., Steinbauer, P., Zavřel, J., Vampola, T., Valášek, M., Synthesis and implementation of vibration suppression by 6 DOF active platform, Bulletin of Applied Mechanics 7 (28) (2011) 85–90.

[14] Svatoš, P., Šika, Z., Valášek, M., Bauma, V., Polach, P., Optimization of anti-backlash fibre driven parallel kinematical structures, Bulletion of Applied Mechanics 8 (31) (2012) 40–44.

[15] Wang, H., Shen, G., Zhou, J., Control strategy of maglev vehicles based on particle swarm algorithm, Journal of Modern Transportation 22 (1) (2014) 30–36.

[16] Yildiz, A. R., Solanki, K. N., Multi-objective optimization of vehicle crashworthiness using a new particle swarm based approach, International Journal of Advanced Manufacturing Technology 59 (2012) 367–376.

[17] Zajíček, M., Dupal, J., Possibility of identification of elastic properties in laminate beams with cross-ply laminae stacking sequences, Applied and Computational Mechanics 5 (2) (2011) 239–252.