Вісник Тернопільського національного технічного університету
Scientific Journal of the Ternopil National Technical University
2016, № 2 (82)
ISSN 1727-7108. Web: visnyk.tntu.edu.ua

INSTRUMENT-MAKING AND INFORMATION-MEASURING SYSTEMS

ПРИЛАДОБУДУВАННЯ ТА ІНФОРМАЦІЙНО-ВИМІРЮВАЛЬНІ СИСТЕМИ

UDC 004.02

THE PROPOSAL OF THE OPTIMISATION TIME REDUCTION ALGORITHM

Krzysztof Augustynek; Kornel Warwas

University of Bielsko-Biała, Bielsko-Biała, Poland

Summary. *In many cases, solving optimisation problems and a dynamic optimisation problem, in particular is time-consuming. This is due to the long time of calculation of the objective function value. For example, during optimisation of the mechanical systems it can be necessary to integrate of the dynamic equations of motion in the whole time interval. For this reason, dedicated methods which allow to calculate approximated value of the objective function have been elaborated. These methods usually are full optimisation algorithms which have embedded methods for calculating of approximated value of the objective function. In this paper new EVCA (Evaluating and Caching) algorithm for reduction optimisation calculations time has been proposed. An important feature of the presented algorithm is that it can be applied to any nonlinear optimisation methods both gradient, non-gradient and stochastic. Presented approach doesn't need to modify of optimisation algorithms and methods which have been used to calculate objective function value. The algorithm uses two mechanisms: estimating of the objective function value and caching its values for all calculated earlier points. Such approach allows to effectively speed up the optimisation process, especially optimisation of the physical systems. The results of the optimisation for the benchmark functions and double pendulum on the cart with using EVCA algorithm have been presented.*

Key words*: optimisation, evaluation of the objective function, genetic algorithm.*

Received 05.04.2016

Introduction. Dynamic optimisation is one of the most common problems which are solved for technical systems. Many of the control tasks are formulated as dynamic optimisation problem. Solution of such tasks with using classical optimisation methods both gradient, non-gradient and stochastic can take long time [1]. It can be caused by the complexity of the physical model or it can result from necessity of integrating the equations describing the dynamic of the system. Due to long time of calculation such approach can't be appropriate for direct implementation in controllers. In order to realize the optimisation calculation in real time it can be necessary to introduce some simplification to the model, which can speed up a calculation of the objective function. Another approach can rely on the use of optimisation results to train an artificial neural network [1, 2, 3]. Such trained neural network is suitable to obtain results in real time, but the training process is time-consuming. This is due to collect a series of optimisation calculations results which form training set of the neural network. From the above it follows that the reduction time of the optimisation process is necessary in order to perform calculations effectively. In many engineering problems time of the optimisation can be significant reduced by using dedicated algorithm, such as subproblem approximation method [4, 5]. In this approach dependent variables are replaced through the least squares fitting

Corresponding author: Krzysztof Augustynek; e-mail: mpkarpinski@gmail.com

approximation process. A constrained problem is formulated into a basic unconstrained problem by using penalty functions. The penalized functions are then minimized until the convergence is reached or the iterations are terminated. Subproblem approximation method possesses the ability to find the global optimum in full design space. Moreover, it offers distinct advantages in many engineering problems, combining simplicity, satisfactory accuracy and efficiency of computation [4]. In the paper [6] has been concluded that the subproblem approximation method is more efficient than the first order optimisation method. The optimisation process can be also improved by limiting of the number of time-consuming calculations. It can be obtained by estimating the value the objective function with using the artificial neural networks or using appropriate regression model [7, 8]. There are also other methods which use for example evolutionary algorithm to estimate of the objective function value [9, 10]. These methods are based on estimation of the objective function value in the whole domain or divide the analysed problem into smaller optimisation tasks which are less complicated to solve.

In this paper new algorithm of reduction optimisation calculations time has been presented. The main feature of this algorithm is that it can be applied to any optimisation method in order to speed up the optimisation process. Proposed algorithm doesn't require to modify methods of solving nonlinear optimisation tasks and it contains two mechanisms: estimating and caching of the objective function values. Such approach allows to effectively speed up the optimisation process, especially for physical systems. In the paper description of the EVCA algorithm and results obtained with using that algorithm for solving various numerical models have been presented.

Application of EVCA algorithm in nonlinear optimisation methods. In this paragraph own algorithm of reduction of the time-consuming calls of the objective function have been presented. It can be acheived by replacing exact determination of the objective function value by appropriate value which are evaluated or taken from the cache.

Let us assume $\mathbf{x} = \left[x_1 \quad \ldots \quad x_i \quad \ldots \quad x_{n_d}\right]^T$ is a point in n_d dimensional feasible set calculated in current optimisation iteration. Further, let $\mathbf{x}^{(j)} = \left[x_1^{(j)} \quad \ldots \quad x_i^{(j)} \quad \ldots \quad x_{n_d}^{(j)}\right]^T$ be one from n_O points in neighborhood of point $\mathbf{x}$ for which objective function value $y^{(j)} = \Omega\left(\mathbf{x}^{(j)}\right)$ has been calculated in previous iterations. All points from that neighborhood have to fulfil the following condition:

$$\left\|\mathbf{x}^{(j)} - \mathbf{x}\right\| \le \varepsilon_1 \tag{1}$$

where $\left\|\mathbf{x}^{(j)} - \mathbf{x}\right\|$ is Euclidean norm,

ε_1 – defines the acceptable distance between point $\mathbf{x}$ and points taken from its neighborhood.

The main aim of further considerations is to present the methodology for determination of exact or estimated value of the objective function in point $\mathbf{x}$.

General concept of the EVCA algorithm has been presented in Figure 1. It has been considered a few cases that can occur during optimisation calculations. In the first case (figure 1a) objective function value is neither evaluated nor taken from cache, because the number of points n_O in neighborhood of point $\mathbf{x}$ is too small. In this case exact value of the objective function has been calculated. In the next considered case (figure 1b) the number of points in neighborhood of point $\mathbf{x}$ is acceptable. Additionally, if all coordinates of the point $\mathbf{x}$ satisfy condition:

$$\min_{j \in <1,n_O>} x_i^{(j)} \le x_i \le \max_{j \in <1,n_O>} x_i^{(j)} \tag{2}$$

then objective function value can be evaluated using appropriate estimator. Otherwise, exact value of the objective function $y = \Omega(\mathbf{x})$ will be calculated (figure 1c). In the last considered case it has been assumed, that there exists point $\mathbf{x}^{(j)}$ and it satisfies condition:

$$\left\| \mathbf{x}^{(j)} - \mathbf{x} \right\| \le \varepsilon_2 \tag{3}$$

where ε_2 defines acceptable distance between two points which are very close to each other. It can be assumed that objective function value of the point $\mathbf{x}$ is calculated according formulae $y = \Omega(\mathbf{x}) = \Omega\left(\mathbf{x}^{(j)}\right)$. It means that objective function value can be taken from the cache which contains values of objective function for points $\mathbf{x}^{(1)},\ldots,\mathbf{x}^{(n)}$, where n is the number of all points for which exact value of the objective function have been calculated earlier.

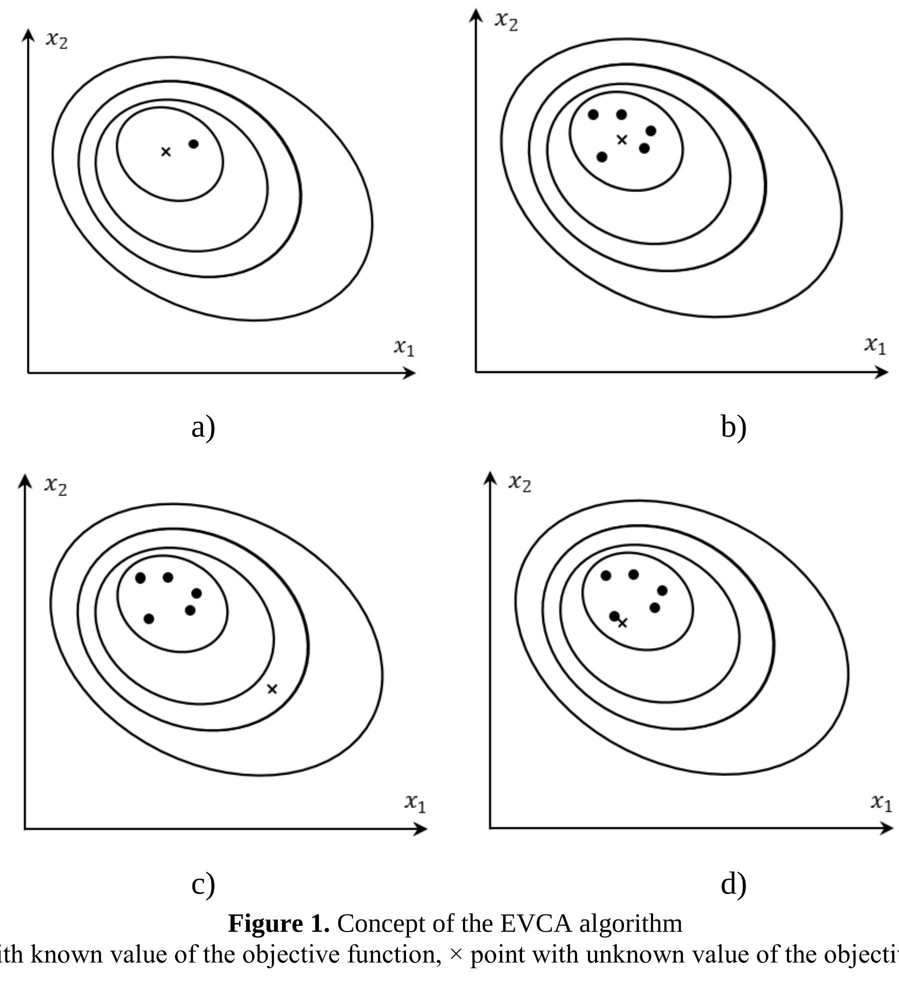

Figure 1. Concept of the EVCA algorithm
● point with known value of the objective function, × point with unknown value of the objective function

General diagram of the EVCA algorithm written in flowchart form has been presented in figure 2. In this diagram it can be seen the main loop of the algorithm with the key modules (check, get, add).

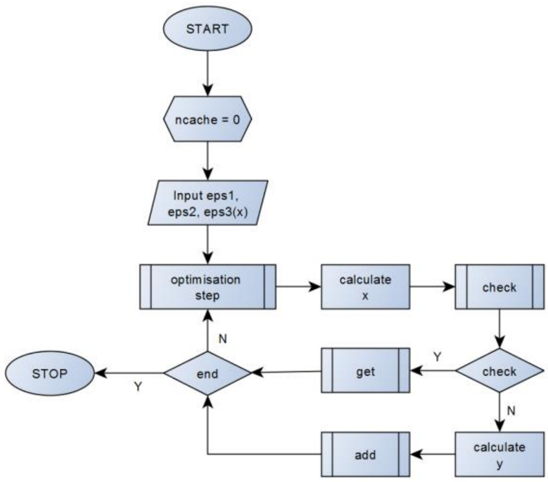

Figure 2. Flow diagram of the EVCA algorithm

Figure 3 shows basic operations and data flow in the module which is responsible for source selection of objective function value. The sources can be as follow:

- cache – value of the objective function will be taken from the cache,
- estimator – value of the objective function will be evaluated with appropriate estimator,
- exact value – value of the objective function will be calculated by calling method which contains definition of the objective function.

In the first step the algorithm looks for any point in the cache which is close to considered point $\mathbf{x}$. If no point has been found in the cache than it searches appropriate neighborhood of point $\mathbf{x}$ which is necessary to evaluate objective function value for this point. The algorithm introduces an additional probabilistic factor, which can force exact calculation of the objective function value, despite the fulfilment of the conditions required for the evaluation.

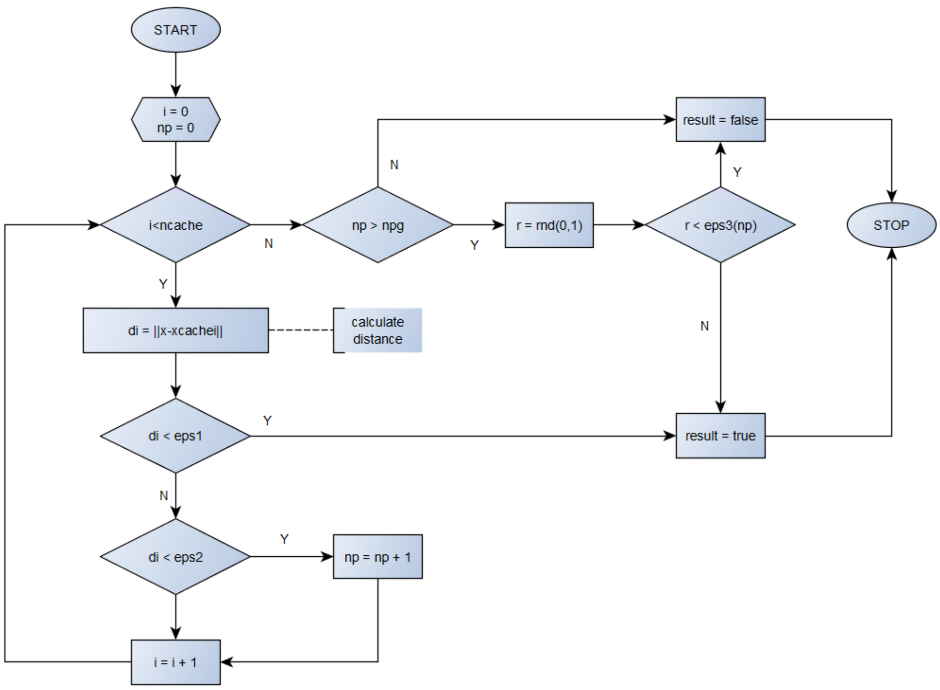

Figure 3. Flow diagram of check submodule in EVCA algorithm

The next diagram (Figure 4) shows necessary steps during which objective function value is taken from the cache or evaluated with using appropriate estimator.

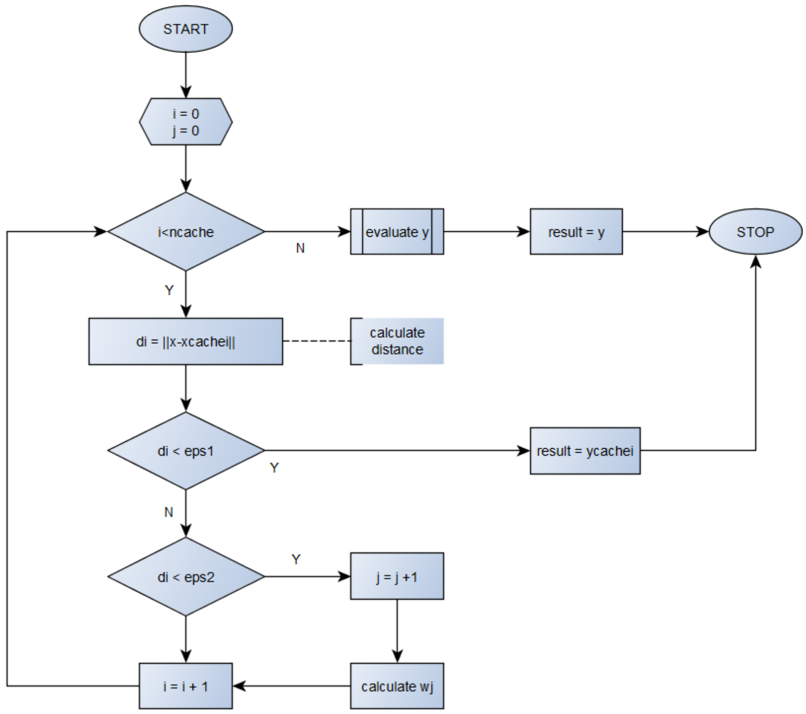

Figure 4. Flow diagram of cache and estimation of objective function in EVCA algorithm

After each exact calculation of the objective function value in a given point, this value is added to the cache according to Figure 5.

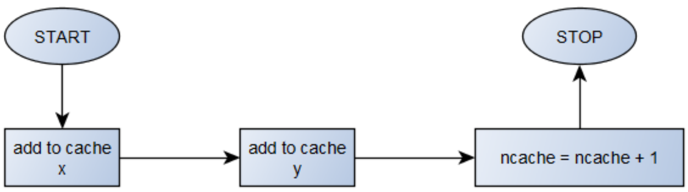

Figure 5. Flow diagram of adding objective function to the cache in EVCA algorithm

Sample possible estimators for evaluations of objective function value have been presented in Figure 6. Among the estimators there are weighted average (Figure 6a), artificial neural network (Figure 6b), regression functions (Fig. 6 c) and evolutionary estimator (Figure 6 d) [11]. In this work objective function value has been estimated with using a weighted average method. Estimated objective function value can be calculated according formula:

$$\Omega^*(\mathbf{x}) = \sqrt{\frac{\sum_{i=1}^{n_O} w_i \Omega\left(\mathbf{x}^{(i)}\right)}{\sum_{i=1}^{n_O} w_i}} \tag{4}$$

where w_i are weights which values depend on distance between $\mathbf{x}^{(i)}$ and $\mathbf{x}$.

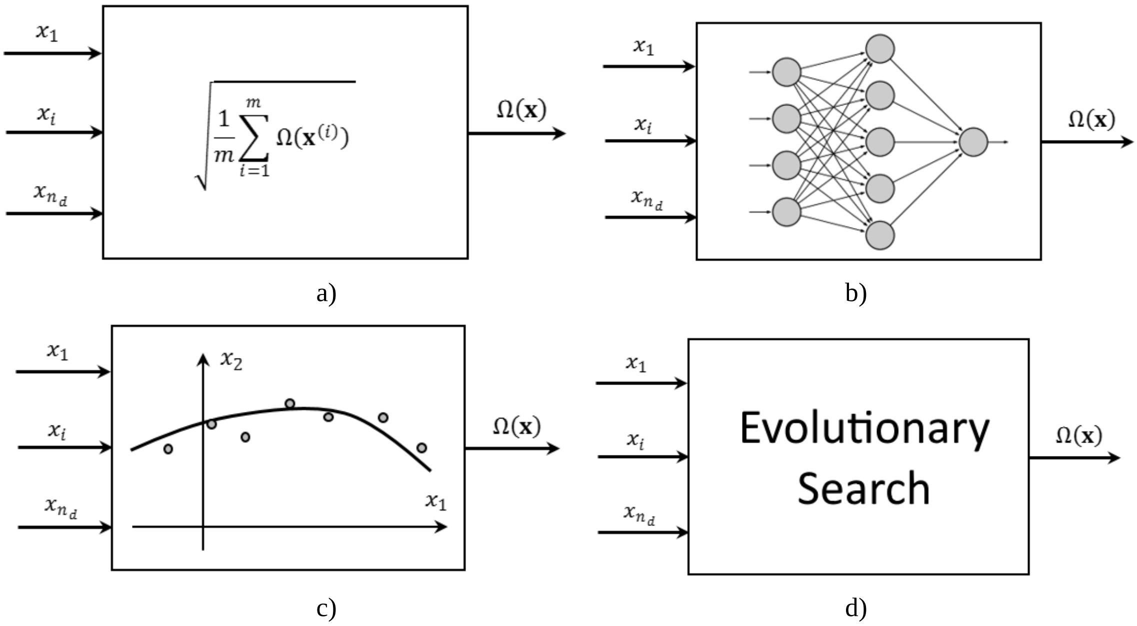

Figure 6. Schematic representation of objective function evaluation methods

Caching mechanism can have own implementation as dictionary with key value elements or dedicated tools such as Memcached [12] or Redis [13] can be used. The main advantage of the ready to use tools is the ability to manage the available memory allocation and using multiple physical machines connected in a cluster.

Numerical simulations of benchmark functions. Algorithm of evaluation and caching of objective function value has been tested by solving nonlinear optimisation of benchmark functions [14]. These functions have known solutions and they are continuous in the feasible set. Mentioned functions have either ordinary courses or many local extrema and saddle points. Determination of the global extrema is difficult task and it strongly depends on assumed starting point [15, 16]. For all analysed benchmark functions dynamic optimisation problem has been solved using various algorithms [15 – 19]:

- Powell Method (PM);
- Nelder-Mead Method (NM);
- Variable Metrics Method (VM);
- Hooke-Jeeves Method (HJ);
- Classical Genetic Algorithm (GA);
- Particle Swarm Optimisation (PSO).

Some of above methods, like classical genetic algorithm and particle swarm optimisation, have been taken from a computational intelligence. Other are classical gradient and gradientless optimisation methods. In all considered optimisation problems minimum of the objective function has been determined. Assumed parameters of the optimisation methods have been presented in Table 1.

Table № 1.
Optimisation methods parameters

Optimisation method	Parameters
Powell Method (PM)	$\varepsilon_{PM} = 10^{-10}$
Nelder-Mead Method (NM)	$\varepsilon_{NM} = 10^{-10}$
Variable Metrics Method (VM)	$\varepsilon_{VM} = 10^{-10}$
Hooke-Jeeves Method (HJ)	$\varepsilon_{HJ} = 10^{-1}$ $\tau = 10^{3}$ $N_{\max} = 1500$
Classical Genetic Algorithm (GA)	$N_{GA} = 30$ $p_C = 0.7$ $p_M = 0.2$ $N_{\max} = 50$
Particle Swarm Optimisation (PSO)	$N_{PS} = 10$ $N_{\max} = 50$

where: ε_{PM} – the fractional tolerance,
ε_{NM} – fractional convergence tolerance,
ε_{VM} – gradient convergence tolerance,
ε_{HJ} – iteration accuracy,
τ – initial iteration step,
$N_{\max}$ – maximum number of iterations,
N_{GA} – number of individuals in population,
p_C – crossover probability,
p_M – mutation probability,
N_{PS} – number of particles in the swarm.

In the case of optimisation with application of genetic algorithms it is assumed real-number representation of genes in chromosomes and the following genetic operators have been used [11]: natural selection, arithmetical one-point crossover, in which a new chromosome is a linear combination of two vectors and non-uniform mutation.

Easom's function is first analysed function (Figure 7). This function is described by formulae:

$$\Omega(x_1, x_2) = -\cos x_1 \cos x_2 e^{-(x_1-\pi)^2-(x_2-\pi)^2} \tag{5}$$

It has only one global extrema obtained for $0 \le x_1, x_2 \le 5$ which equals $\Omega(x_1, x_2) = 0$ in point $x_1 = x_2 = \pi$.

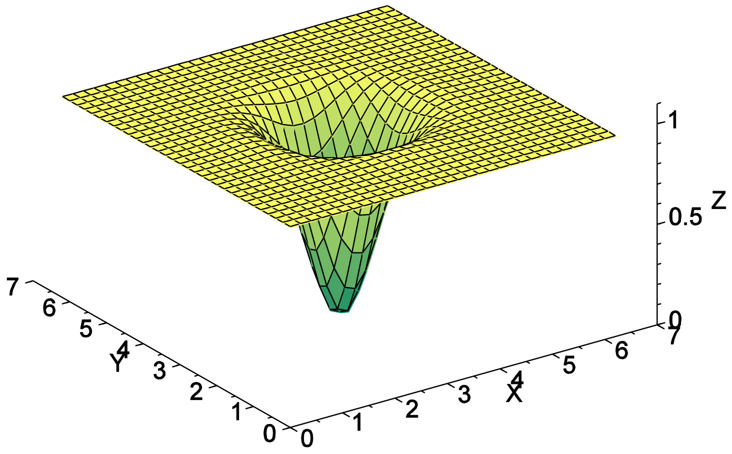

Figure 7. Easom's function

Results of nonlinear optimisation task have been summarized in Table 2. It can be seen that evaluating and caching mechanisms have been often used for stochastic optimisation methods like GA and PSO. These mechanisms are less important for classic algorithms of nonlinear optimisation. Analysing the obtained solution, it can be concluded that only two methods have been determined a solution different than the global optimum. It can be due to an incorrectly selected starting point. Other methods have been determined the solution close to the optimal point.

Table № 2.
Result of simulation for Easom's function

Method	n_C	n_E	n_Ω	w_C [%]	w_E [%]	x_1^{opt}	x_2^{opt}	$\Omega(x_1^{opt}, x_2^{opt})$
GA	1208	399	2500	48.320	15.960	3.141	3.140	0.000002434
NM	2	37	2692	0.080	1.480	3.141	3.141	0.000000001
PM	19	17	2775	0.760	0.680	1.305	1.305	0.999918898
HJ	4	21	2906	0.160	0.840	3.141	3.141	0.000000000
VMM	1	0	3911	0.040	0.000	1.017	1.017	0.999966802
PSO	59	442	6959	2.360	17.680	3.141	3.141	0.000000000

Second analysed function has been axis parallel hyper-ellipsoid function (Figure 8), which written in the following form:

$$\Omega(x_1, x_2) = x_1^{\,2} + 2x_2^{\,2} \tag{6}$$

It has only one global extrema obtained for $0 \le x_1, x_2 \le 5$ which equals $\Omega(x_1, x_2) = 0$ in point $x_1 = x_2 = 0$.

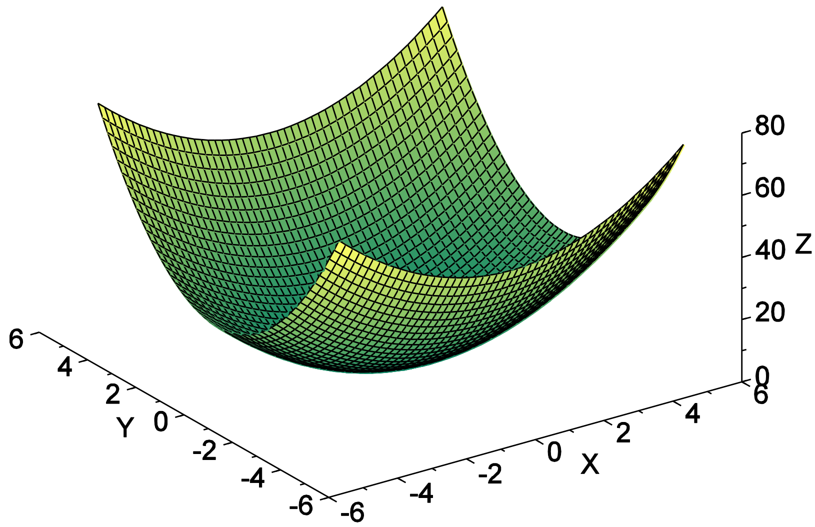

Figure 8. Axis parallel hyper-ellipsoid function

Results of optimisation obtained with various nonlinear optimisation algorithms have been shown in table 3. Analysing the results, it can be concluded that PSO and GA method use most frequently estimating and caching mechanisms. It can be noted that usage of caching mechanism for GA method is over 50%. In the case of PSO method usage of caching mechanism is negligible, but number of evaluations is more than 30%. Optimal solution for analysed benchmark function has been correctly determined by all optimisation methods.

Table № 3.
Result of simulation for axis parallel hyper-ellipsoid function

Method	n_C	n_E	n_Ω	w_C [%]	w_E [%]	x_1^{opt}	x_2^{opt}	$\Omega(x_1^{opt}, x_2^{opt})$
GA	1402	0	2500	56.080	0.000	-0.004	0.0002	0.000016844
NM	3	44	2707	0.120	1.760	2.93e-5	5.83e-6	0.000000001
PM	58	65	2891	2.320	2.600	1.25e-5	0.0	0.000000000
HJ	3	38	3027	0.120	1.520	1.56e-5	7.83e-5	0.000000000
VMM	1	25	3067	0.040	1.000	1.67e-6	8.56e-7	0.000000000
PSO	32	813	6103	1.280	32.520	9.31e-7	2.05e-7	0.000000000

Another considered function has been Rosenbrock’s valley (Figure 9), which is described by the following formulae:

$$\Omega(x_1, x_2) = 100\left(x_2 - x_1^{\,2}\right)^2 + (1 - x_1)^2 \qquad (7)$$

It has only one global extrema obtained for $-2 \le x_1, x_2 \le 2$ which equals $\Omega(x_1, x_2) = 0$ in point $x_1 = x_2 = 1$.

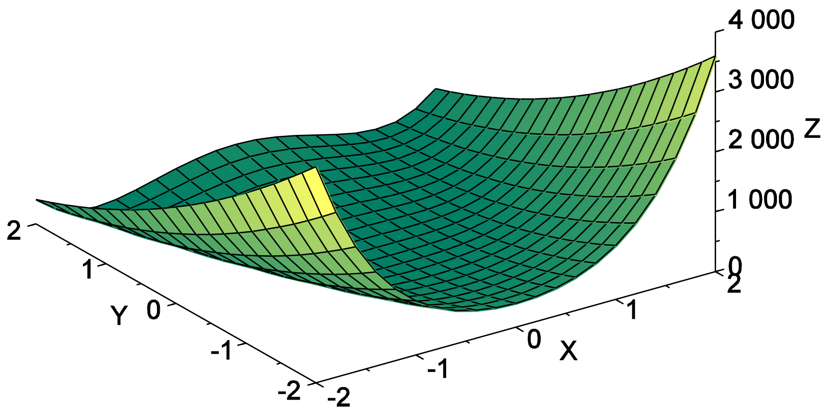

Figure 9. Rosenbrock's valley

Analysing the results, presented in the Table 4, it can be concluded that usage of evaluating for all considered optimisation methods is small. It can be also noted that usage of caching mechanism for GA method is over 50%. This algorithm gives the least accurate solution in relation to other methods. However, this is a feature of genetic algorithms which can point the area where the global solution is located.

Table № 4.
Result of simulation for Rosenbrock's valley

Method	n_C	n_E	n_Ω	w_C [%]	w_E [%]	x_1^{opt}	x_2^{opt}	$\Omega(x_1^{opt}, x_2^{opt})$
GA	1356	0	2500	54.240	0.000	0.926	0.843	0.028491650
NM	13	48	2719	0.520	1.920	1.000	1.000	0.000000000
PM	19	7	2758	0.760	0.280	0.999	0.999	0.000000000
HJ	4	8	2812	0.160	0.320	1.000	1.000	0.000000000
VMM	1	7	2823	0.040	0.280	1.000	1.000	0.000000000
PSO	0	60	5865	0.000	2.400	1.000	1.000	0.000000000

The last tested function has been Rastrigin's function (Figure 10) which can be written as follow:

$$\Omega(x_1, x_2) = 20 + x_1^2 - 10\cos(2\pi x_1) + x_2^2 - 10\cos(2\pi x_2) \qquad (8)$$

It has only one global extrema obtained for $-2 \le x_1, x_2 \le 2$ which equals $\Omega(x_1, x_2) = 0$ in point $x_1 = x_2 = 0$.

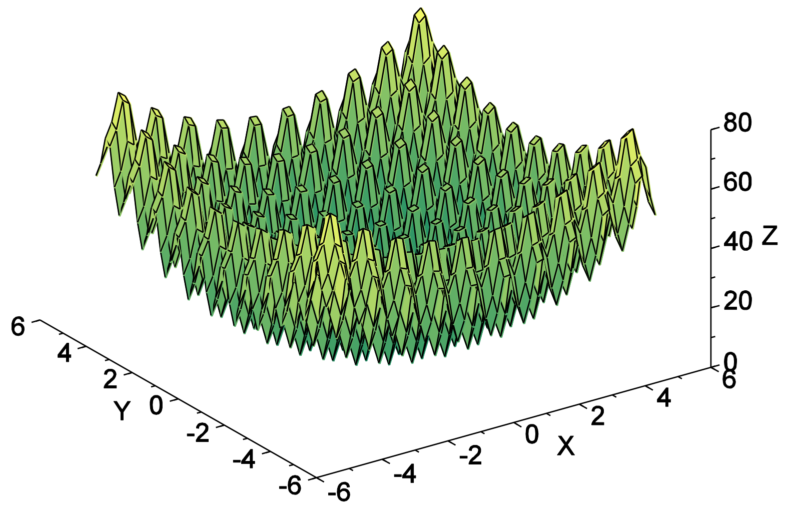

Figure 10. Rastrigin's function

This function is the most difficult to solve from among all analysed functions. It has many local extrema which can be a big obstacle for the classical algorithms of nonlinear optimisation. Results presented in the table 5 confirm this hypothesis. Only VMM and PSO method correctly pointed the global minimum. Usage of evaluating and caching mechanisms is similar to previous analysed benchmark functions (about 50%). Value of the objective function obtained from GA method is close to global minimum, but its coordinates show that it is only local extrema.

Table № 5.
Result of simulation for Rastrigin's function

Method	n_C	n_E	n_Ω	w_C [%]	w_E [%]	x_1^{opt}	x_2^{opt}	$\Omega(x_1^{opt}, x_2^{opt})$
GA	1404	0	2500	56.160	0.000	-1.326	-0.009	0.018753945
NM	3	71	2748	0.120	2.840	0.995	0.995	1.989918232
PM	25	30	2841	1.000	1.200	0.995	0.995	1.989918114
HJ	4	28	2945	0.160	1.120	0.995	0.995	1.989918119
VMM	1	0	2956	0.040	0.000	0.000	0.000	0.000000000
PSO	32	692	5991	1.280	27.680	-2.54e-6	-3.26e-6	0.000000000

Usage of evaluating and caching mechanisms for all considered functions is negligible in the case of classical gradient and nongradient nonlinear optimisation methods. This usage is considerable when optimisation problem is solved with using stochastic methods. According to the authors a EVCA algorithm can make a big difference in the case of optimisation of physical systems, for which a time of the objective function calculation can be long. For example, mechanical systems can need to integrate dynamic equations of motion in the whole time interval when the objective function is determined. In order to prove this hypothesis, in the next chapter results of numerical simulations of the double pendulum on the cart have been presented.

Numerical simulations of physical system. Pendulum systems are often used as examples for highly nonlinear underactuated mechanical systems. Stabilization of the pendulum is still a challenging problem and can be used to show the effectiveness of control systems in analogy with the control of many real systems. It can also be used for the verification of the designed control systems or for control education in laboratories. Researchers have been analyzing the stabilization of single and multiple pendulum systems in the instable inverted position, the swing-up or pendulums mounted on a cart which moves on a horizontal rail [20].

The double pendulum on a cart is modeled by two rigid links connected by rotational joints which lengths are described as l_1 and l_2. First link is connected with the cart which is modeled as rigid body (Figure 11). The general coordinates φ_1 and φ_2 describe absolute rotation angles of the links. Friction in the joints have been taken into account in order to obtain more realistic system. All physical parameters of the links have been presented in table 6. The cart's movement is described by x_0. It has been assumed that mass of the cart is $m_0 = 4[kg]$.

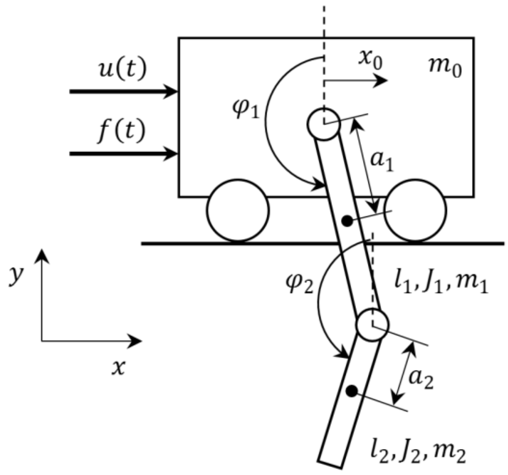

Figure 11. Double pendulum on a cart

Table № 6.
Mechanical parameters of the double pendulum links [20]

Name	Symbol	Unit of measure	Link 1	Link 2
Length	l_i	m	0.3560	0.3560
Distance to center of gravity	a_i	m	0.1800	0.1480
Mass	m_i	kg	0.7750	0.6540
Moment of inertia	J_i	Nms^2	0.0224	0.0179
Friction constant	d_i	Nms	0.0050	0.0050

Dynamic equations of motion of the double pendulum on a cart have been derived from Lagrange equations of the second kind which can be written as follows [20]:

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{q}_i} - \frac{\partial L}{\partial q_i} = f_i \tag{9}$$

where: L – Lagrangian function,

$\mathbf{f} = (f_i)_{i=1,2,3}$ – vector of generalized forces which describes influence of external forces and forces which result from friction in the joints,

$\mathbf{q} = (q_i)_{i=1,2,3} = [x_0 \quad \varphi_1 \quad \varphi_2]^T$ – vector of generalized coordinates.

Finally, equations of motion can be written in the matrix form [20]:

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{g}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{f} \tag{10}$$

where: $\mathbf{M}(\mathbf{q})$ – mass matrix,

$\mathbf{G}(\mathbf{q}, \dot{\mathbf{q}})$ – vector of centrifugal and Coriolis forces.

Optimisation with using EVCA algorithm has been made for two test cases of double pendulum on the cart. The aim of the optimisation is to determine the force acting on the cart which can restore equilibrium of the system. This force has been modeled by means of spline functions of 1st order. For time period $t \in \left\langle t^{(i)}, t^{(i+1)} \right\rangle$ force acting on the cart can be written in the form:

$$u(t) = a^{(i)} + b^{(i)}\left(t - t^{(i)}\right) \tag{11}$$

where $a^{(i)}$, $b^{(i)}$ are coefficient of the spline functions. Decisive variables in the analysed problem are values of the force in interpolation nodes. Vector of decisive variables can be written as follow:

$$\mathbf{X} = [X_1 \quad \dots \quad X_i \quad \dots \quad X_n] \tag{12}$$

where: $X_i = u\left(t^{(i)}\right)$ – value of the force in the time stamp $t^{(i)}$,

n – number of discrete time stamps (number of interpolation nodes).

Case 1

In analysed case, link 1 has been tilted so that it lost static equilibrium. Initial conditions assumed in case 1 have been presented in table 7.

Table № 7.
Parameters of the double pendulum on a cart for case 1

Symbol	Unit of measure	Value
φ_1	rad	$1.2 \cdot \pi$
φ_2	rad	π
x_0	m	0
$f(t)$	N	0

In order to solve dynamic optimisation problem Powell method and variable metric method have been used. During exact calculation of the objective function value dynamic equations of motion (10) are integrated in whole time interval. This operation involves a large computational effort and is time-consuming. Fourth order Runge-Kutta [16] method has been applied to integrate dynamic equations of motion. It has been assumed that course of the force $u(t)$ is described by 10 interpolation nodes. Results obtained from optimisation have been shown in Figure 12.

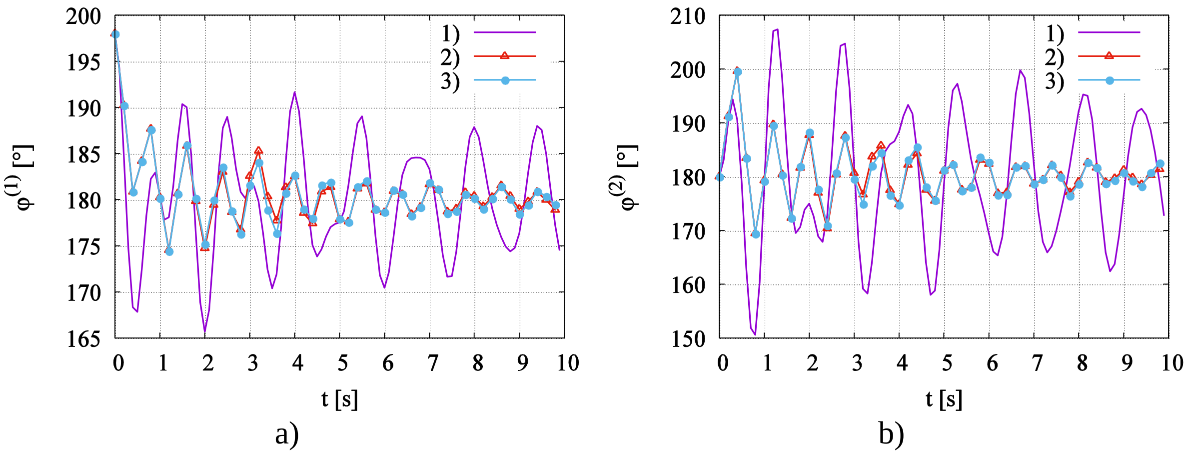

Figure 12. Courses of rotation angle of first (a) and second (b) link for case 1
1) before optimisation, 2) after PM optimisation, 3) after VMM optimisation

Results confirm that the optimisation process has been performed successfully. It means that magnitudes of rotation angle $\varphi^{(1)}(t)$ and $\varphi^{(2)}(t)$ are significantly smaller than before optimisation. In order to assess influence of evaluating and caching mechanism on results obtained from optimisation quantitive indicators have been calculated and summarized in table 8.

Table № 8.
Result of simulation for double pendulum on a cart for case 1

Method	n_C	n_E	n_Ω	w_C [%]	w_E [%]
PM	318	54	837	37.99	6.45
VMM	7	228	348	2.01	65.51

Above results confirm hypothesis, that in physical systems mechanisms contained in EVCA algorithm are frequently used. In the case of Powell method usage of caching mechanism is close to 40%, while usage of evaluating mechanism for variable metric method is over 60%. Results show that when the evaluating and caching mechanism are applied, the time of optimisation calculations is significantly smaller. Analysing summarized values from the Table 8 it can be concluded that the time of calculation has been reduced about 45% in Powell method and nearly 70% in the case of variable metric method.

Case 2

In the second case pendulums are in static equilibrium at the initial time, while the variable in time external force acting on the cart. Initial conditions assumed during simulations have been presented in table 9.

Table № 9.
Parameters of the double pendulum on a cart for case 2

Symbol	Unit of measure	Value
φ_1	*rad*	π
φ_2	*rad*	π
x_0	*m*	0
$f(t)$	*N*	$10+10\sin t$

The aim of the optimisation task is to reduce influence of the external force acting on the cart, in order to preserve position from static equilibrium for both pendulums. Dynamic optimisation task has been solved with using Powell and Hooke-Jeeves method. It has been assumed that course of the external force $u(t)$ acting on the cart is described by 20 interpolation nodes. Results obtained for optimisation have been shown in figure 13.

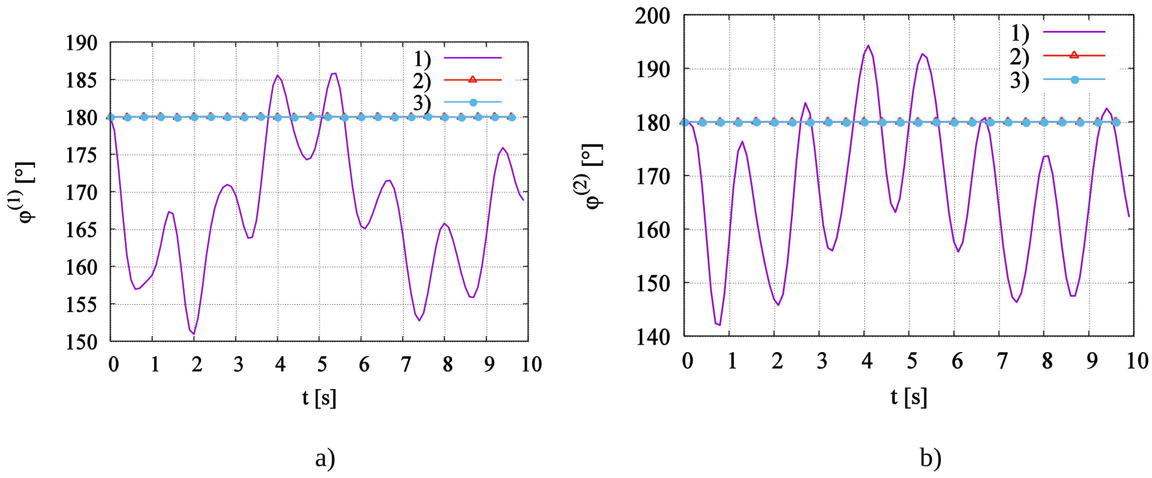

Figure 13. Courses of rotation angle of first (a) and second (b) link for case 2
1) before optimisation, 2) after PM optimisation, 3) after HJ optimisation

Analysing obtained results it can be seen that angles of rotation $\varphi^{(1)}(t)$ and $\varphi^{(2)}(t)$ are constant and both pendulums preserve initial position. In order to assess influence of evaluating and caching mechanisms on dynamic optimisation process quantitive indicators have been calculated and summarized in table 10.

Table № 10.
Result of simulation for double pendulum for case 2

Method	n_C	n_E	n_Ω	w_C [%]	w_E [%]
PM	8713	2051	21514	40.50	9.53
HJ	55	5794	11131	0.49	52.05

Above results confirm again hypothesis, that in physical systems mechanisms contained in EVCA algorithm are frequently used. In the case of Powell method usage of caching mechanism is close to 40% and usage of evaluating mechanism for Hooke-Jeeves method is over 50%. Analysing summarized values from the table 10 it can be concluded that the time of calculation has been reduced about 50% for both analysed methods.

Conclusions. In the paper EVCA algorithm which allow to reduce optimisation calculations time has been presented. This algorithm is useful for the problem, in which time of calculation of the objective function is long. Analysed algorithm uses two mechanisms: evaluating and caching. During optimisation exact values of the objective function are collected in the cache. The first mechanism is applied when analysed point lie close to another stored in the cache. If such point can’t be taken from the cache, value of the objective function can be evaluated with using appropriate estimator. In the paper the results of using mentioned mechanisms to benchmark functions and physical system like double pendulum on the cart have been shown. The Tests have been performed with using classical gradient and nongradient optimisation methods and stochastic methods like genetic algorithms and particle swarm optimisation method. Obtained results show that usage of evaluating and caching mechanisms for all analysed benchmark functions is negligible if classical optimisation methods have been used. Huge usage of these mechanisms can be seen when the methods from computational intelligence have been taken into account. Very interesting results from dynamic optimisation have been obtained for double pendulum on the cart. Significant usage of evaluating and caching mechanisms have been noted for all considered optimisation methods. When the Powell method has been applied, the usage of caching mechanism is nearly 50%. In the case of Hooke-Jeeves and variable metric method usage of the evaluating mechanism is over 50%. It means that the time of optimisation process for double pendulum on the cart and other physical system can be half as long if EVCA algorithm has been applied.

References

1. Warwas K., Augustynek K. Dynamic optimisation of articulated vehicle motion for control of stability in critical situation. IDAACS'2015: 8th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications: Vol. 1, 2015, pp. 232 – 237.

2. Warwas K., Augustynek K. An Application of Neural Networks to Control Stability of an Articulated Vehicle in Real Time, Information Systems Architecture and Technology: Proceedings of 36th International Conference on Information Systems Architecture and Technology – ISAT 2015 – Part I, Springer, 2016, pp. 135-146.

3. Anastassiou G. Intelligent Systems: Approximation by Artificial Neural Networks, Intelligent Systems Reference Library, Volume 19, 2011.

4. Zhang J., Zhu H., Zhao C. Combined finite element analysis and subproblem approximation method for the design of ultrasonic motors, Sensors and Actuators a Physical 163 (2), 2010, pp. 510 – 515.

5. ANSYS, Academic Research, Release 11, Help System, Advanced Analysis Techniques Guide, ANSYS, Inc.

6. Chen C., Yeh M. Optimum structural design of composite xylophone bars, CD-ROM Proceedings of The Sixteenth International Conference on Composite Materials, 2007.

7. Patnaik S., Guptill J., Hopkins D. Subproblem optimization with regression and neural network approximators, Comput. Methods Appl. Mech. Engrg. 194, 2005, pp. 3359 – 3373.

8. Lagaros N., Charmpis D., Papadrakakis M. An adaptive neural network strategy for improving the computational performance of evolutionary structural optimization, Comput. Methods Appl. Mech. Engrg. 194, 2005, pp. 3374 – 3393.

9. Aittokoski T., Miettinen K. Efficient evolutionary approach to approximate the Pareto-optimal set in multiobjective optimization, UPS-EMOA, Journal Optimization Methods & Software. The International Conference on Engineering Optimization, Volume 25 Issue 6, 2010, pp. 841 – 858.

10. Mottaez A., Iyer M. Accurate approximation of the objective function for solving the gate-sizing problem using a numerical solver, US 8826218 B2, Google Patents, 2014.

11. Liang K., Yao X., Newton C. Evolutionary Search of Approximated N-Dimensional Landscapes, International Journal of Knowledge-based Intelligent Engineering Systems. Vol. 4, 2000, pp. 172-183.

12. Pandey S. Caching Using Memcached in Open Source Searching: Execution Time and Throughput Improvement Using Open Source Key-Value Caching Techniques-Memcached, Lap Lambert Academic Publishing, 2015.

13. Carlson J. Redis in Action, Manning Publications Co., 2013.

14. Molga M., Smutnicki C. Test functions for optimization needs, 2005, http://www.zsd.ict.pwr.wroc.pl/files/docs/functions.

15. Pedregal P. Introduction to Optimization, Springer-Verlag Inc., 2004.

16. Press W., Teukolsky S., Vetterling W., Flannery B. Numerical Recipes 3rd Edition: The Art of Scientific Computing, Cambridge University Press, Cambridge, 2007.

17. Luersen M., Le Riche R. Globalized Nelder–Mead Method for Engineering Optimization, Computers and Structures, 2004, pp. 2251 – 2260.

18. Sivanandam S., Deepa S. Introduction to Genetic Algorithms, Springer-Verlag Berlin Heidelberg, Berlin, 2008.

19. Clerc M. From Theory to Practice in Particle Swarm Optimization, Handbook on Swarm Intelligence. Vol. 8, 2010, pp. 3 – 36.

20. Timmermann J., Khatab S., Ober-Blöbaum S., A. Trächtler, Discrete Mechanics and Optimal Control and its Application to a Double Pendulum on a Cart, Preprints of the 18th IFAC World Congress Milano, August 28 – September 2, 2011, pp. 10199 – 10206.

21. Bauchau O. Flexible Multibody Dynamics, Solid Mechanics and Its Applications, Springer Netherlands, 2011

УДК 004.02

АЛГОРИТМ ОПТИМІЗАЦІЇ СКОРОЧЕННЯ ЧАСУ ОБЧИСЛЕНЬ

Кшиштоф Авґустинек; Корнель Варвас

Університет у Бєльсько-Бялій, Бєльсько-Бяла, Польща

Резюме. *В багатьох випадках вирішення проблем оптимізації, зокрема динамічних проблем оптимізації, вимагає багато часу. Причиною цього є тривалий період обчислення значення цільової функції. Наприклад, упродовж процесу оптимізації механічних систем може з'явитися необхідність інтегрування динамічних рівнянь руху протягом усього інтервалу часу. В зв'язку з цим, були розроблені спеціалізовані методи, які дозволяють обчислити приблизне значення цільової функції. Ці методи зазвичай ґрунтуються на алгоритмах повної оптимізації та містять методи розрахунку приблизних значень цільової функції. В цій статті запропоновано новий алгоритм EVCA (оцінювання і кешування), призначений зменшити час обчислень під час процесу оптимізації. Важливою особливістю запропонованого алгоритму є те, що він може бути застосований до будь-яких нелінійних методів оптимізації як градієнтних, так і неградієнтних та стохастичних. Для впровадження запропонованого підходу немає необхідності змінювати алгоритми та методи оптимізації, які застосовуються для обчислення значень цільової функції. В алгоритмі використовуються два етапи: оцінювання значень цільової функції та кешування отриманих значень для всіх попередньо розрахованих точок. Такий підхід дозволяє ефективно пришвидшити процес оптимізації, особливо для оптимізації фізичних систем. Представлено результати оптимізації для еталонних функцій та подвійного маятника з рухомою основою із застосуванням алгоритму EVCA.*

Ключові слова*: оптимізація, оцінювання цільової функції, генетичний алгоритм.*

Отримано 05.04.2016