

Some Global Algorithms of Random Search. Examples of Solving Problems

George Filatov

Professor, Doctor of Techn. Sciences, Ukrainian State University of Chemical Technology, Ukraine

Abstract - This article in some degree is a review. In it is represented some global random search algorithms designed to solve complex problems of nonlinear programming. Is considered the setting of algorithms, testing on examples of solving specific problems for which the solution is known. Are adduced the independent and stepper algorithms of random search with different distribution of random trials, gullied algorithms and algorithm with the return after a failed step. Is considered the algorithm, that implements the synthesis of random search and dynamic programming at the optimization of multivariable systems.

Keywords — Nonlinear Programming, Random Search, Global Algorithms.

I. THE ALGORITHM WITH THE RELEASE OF "SUSPICIOUS" SUB-AREA ON THE PRESENCE OF GLOBAL EXTREMUM

The algorithm that is outlined below, allows reduction of number of trials on search using the preliminarily "reconnaissance" about the location of the global extremum. For a broad class of complex objects for optimization can build a distribution law of quality function $F(\mathbf{X})$ in the assumption that the states \mathbf{X} are selected in accordance with a uniform law of distribution across the search area. Thus, if the quality function can be provided in the form of:

$$F(\mathbf{X}) = \sum_{i=1}^m f_i(\mathbf{X}), \quad (1)$$

Where the functions $f_i(\mathbf{X})$ are loosely coupled, then at large number of variables in accordance with the limit theorem of probability theory, the value of F is distributed on a normal law [1].

One way or another, but it is assumed that the distribution law $p(F/\mathbf{A})$ is known to an accuracy of a number of parameters $\mathbf{A} = (a_1, a_2, \dots, a_k)$ of this

$$\mathbf{X}_{i+1} = \begin{cases} \mathbf{X}_i, & \text{at } f(\mathbf{X}_{i+1}) \geq f(\mathbf{X}_i) \\ \mathbf{X}_i + \Xi, & \text{at } f(\mathbf{X}_{i+1}) < f(\mathbf{X}_i) \end{cases}, \quad (2)$$

where: $\Xi f(\mathbf{X})$ – the penalty function in the form [4]:

distribution (e.g., for normal distribution, two parameters is necessary to know: the mathematical expectation and variance). This makes it possible to build such a global search procedure [2].

We divide the entire search area on sub-areas, and we find out in which of them is most advantageous to place a specified number of tests to determine the state in which the quality function has the smallest value. As the selection criterion sub-area naturally take the mathematical expectation of lower sample value corresponding to a given number of tests in this sub-area.

The estimation of unknown value M for each sub-area is required a certain number L of test trials are needed to assess the distribution of parameters $\mathbf{A}_i (i = 1, 2, \dots, s)$ in each sub-area. With this evaluation, it is possible to determine the mathematical expectation M_i of the smallest value of F at the distribution in each area of a specified number of tests N . Is required, at the minimum number of trials L find the sub-area which would meet with the probability of not less than a predetermined, lowest value M . The selected thus sub-area taken as the initial, etc.

The search process thus reduces to determining of the most promising sub-area that subsequently is divided on the following sub-areas or in it are accommodated all of the remaining trials. As you can see, the optimum search needs in an optimal dividing of the entire stock of trials on the "reconnaissance" that allows you to determine the most advantageous sub-area. Part of the remaining trials is evenly distributed in the selected sub-area.

II. THE ALGORITHM OF SLIDE ELLIPSE (ASE)

This algorithm has been suggested by the authors of article [3].

The algorithm is described of recurrent expression:

$$f(\mathbf{X}) = F(\mathbf{X}) + \chi \max_{i \in I} \{0, g_i(\mathbf{X})\} \quad (3)$$

$F(\mathbf{X})$ – quality function;

$g_i(\mathbf{X}) \leq 0, \quad i \in I^-$ – restrictions. Functions $F(\mathbf{X})$ and $g_i(\mathbf{X})$ assumed to be continuous in hyper-parallelepiped:

$$\mathbf{X}^- \leq \mathbf{X} \leq \mathbf{X}^+, \quad (4)$$

$\xi = \{\xi_{i1}, \xi_{i2}, \dots, \xi_{in}\}$ – random vector whose components are not correlated, and are determined by the relation

$$\xi_{ij} = \sum_{k=1}^n a_{jk}^{(i)} \mathbf{R}_k \mathbf{Z}_k (1 - e^{-|y|}), \quad (5)$$

where $\mathbf{Z} = (z_1, z_2, \dots, z_n)$ – the vector uniformly distributed on the unit n -dimensional sphere; $\mathbf{R} = (R_1, R_2, \dots, R_n)$ – vector lengths of the semiaxes of the ellipsoid given a priori; $\mathbf{A}^{(i)} = \{a_{jk}^{(i)}\}$ – transition matrix from the basis of the ellipsoid to the main basis of the space.

$$v_{li} = \frac{2}{\sqrt{2\pi}\sigma_i} \int_0^\infty x \exp\left(-\frac{x^2}{2\sigma_i^2}\right) dx = \sigma_i \sqrt{\frac{2}{\pi}}.$$

In equation (7) the parameters d_1 and d_2 – coefficients respectively of forgetting and learning ($0 \leq d_1 \leq 1; \quad d_2 \geq 0$). In the transition to the new center of search $\mathbf{X}_{i+1} = \mathbf{X}_i$ the first axis of ellipsoid is rotated in the direction:

$$\lambda_{i+1} = d_3 \frac{\lambda_i}{|\lambda_i|} + d_4 \frac{\mathbf{X}_{i+1} - \mathbf{X}_i}{|\mathbf{X}_{i+1} - \mathbf{X}_i|}, \quad (8)$$

where λ_i – the former direction of the axis R_1 . Use of (7) involves stretching along the axis of an ellipsoid R_1 . To build a complete orthonormal basis may be used the reception of orthogonalization [5] of the system of linearly independent vectors, since λ_{i+1} .

In case of violation of constraints (4), we obtain:

$$x_{ij} = \begin{cases} x_j^-, & \text{if } x_{ij} \leq x_j^-; \\ x_{ij}, & \text{if } x_j^- < x_{ij} < x_j^+; \\ x_j^+, & \text{if } x_{ij} \geq x_j^+ \end{cases} \quad (9)$$

It is envisaged that the center of the ellipsoid to the i -th step is located at the point \mathbf{X}_i , and the orientation of the axes depends on the results of previous steps. The value y is normally distributed with zero mean and variance σ_i :

$$f(y) = \frac{1}{\sqrt{2\pi}\sigma_i} \exp\left(-\frac{y^2}{2\sigma_i^2}\right). \quad (6)$$

The lengths of the semiaxes of the ellipsoid are selected as comparable to the size of the field (4), and the dispersion varies according to the recurrence relation [2]:

$$\sigma_{i+1} = \sigma_i \left(d_1 + \frac{d_2 \rho_i}{1 - \exp(-v_{li})} \right), \quad (7)$$

$$\text{Where: } \rho_i = \left\{ \sum_{j=1}^n \left(\frac{x_{ij} - x_{i+1,j}}{R_j} \right)^2 \right\}^{\frac{1}{2}}, \quad v_{ij} - \text{the}$$

absolute value of the central point y for σ_i :

The algorithm (2)-(9) is an algorithm of the slide ellipse (ASE). If the sizes of the ellipsoid are such that he completely covers the region (4), in view of that the function $f(\mathbf{X})$ is continuous then all the conditions of the theorem on convergence are performed [6] and the algorithm ASE at $i \rightarrow \infty$ with probability unit finds the δ – neighborhood ($\delta > 0$) of the global minimum of the function $f(\mathbf{X})$. The algorithm has the smoothing properties in comprehension of [7] and it is therefore is quite universal, is needed only the continuity of function $f(\mathbf{X})$. At the same time, as already noted, ASE allows to find a global extremum.

Consider the application of the algorithm of the sliding of the ellipse to the solution of the general problem of mathematical programming. Following [8], we can write the general problem of mathematical programming:

$$F(\mathbf{X} \rightarrow \min) \quad (10)$$

$$g_i(\mathbf{X}) \leq 0, \quad i \in I^- \quad (11)$$

$$g_i(\mathbf{X}) = 0, \quad i \in I^0, \quad \mathbf{X} \in E^n, \quad (12)$$

where $g_i(\mathbf{X})$ – a continuous function satisfying the Lipschitz condition with constant L : $|g_i(\mathbf{X}_1) - g_i(\mathbf{X}_2)| < L|\mathbf{X}_1 - \mathbf{X}_2|$, and $F(\mathbf{X})$ – unimodal function in space. To solve this problem is proposed in [15] apply the random search algorithm using ASE under the scheme:

1. The starting point \mathbf{X}_0 , which is the permissible solution of the problem, is selected. Let us call $\mathbf{X} \in E^n$ the solution ε -permissible for selected $\varepsilon > 0$, if $g_i(\mathbf{X}) \leq \varepsilon, i \in I^-; |g_i(\mathbf{X})| \leq \varepsilon, i \in I^0$.
2. In δ -neighborhood of point $(\delta \cdot L > \varepsilon) \cdot \mathbf{X}_k, \geq 0$ we find such m ε -permissible solutions of problem $\mathbf{X}_k^{(1)}, \mathbf{X}_k^{(2)}, \dots, \mathbf{X}_k^{(m)}$, such that

$$\sum_{j=1}^m \alpha_j (\mathbf{X}_k^j - \mathbf{X}_k) \neq 0. \quad (13)$$

The resulting system of linearly independent vectors is complemented by random vectors. This system of vectors with probability one is linearly independent. After the ortho-normalizing, we obtain a basis of space with the center in point. This basis we'll call temporary.

4. In the temporary basis we organize the search using ASE with the length of axis

$$R_1 > R_2 = R_3 = \dots R_m \gg R_{m+1} = \dots R_n$$

Where $R_n \leq \frac{\varepsilon}{L}$. As the initial position of ellipsoid's

basis is taken the point \mathbf{X}_k . In the process of the work of ASE his first axes remain in the subspace formed by the vectors (10). At this the penalty function is transformed to the form

$$f(\mathbf{X}) = F(\mathbf{X}) + \max(g_i(\mathbf{X})/\varepsilon)^{2p} \quad (14)$$

where p – a natural number.

5. Further the operation, is noted in paragraphs 2–4, is repeated. As a stopping criterion of the algorithm we can use the criterion proposed in [9].

Therefore, the introduction of ε -permissible solutions can be applied to the problem (10)-(12) the random search method, stipulating the less stringent conditions for the functions F and g_i (the differentiation of F , g_i and the continuity of f are not needed). Therefore, this algorithm is applicable to a wider class of problems than algorithms proposed in [8,10].

Introduction of ε -permissible solutions also allows to apply the theorem on convergence [6], and, as a consequence, with probability unit to find the solution of the problem (10)-(12).

Described algorithm is illustrated by the example of weight optimization of the rod system of frame type (including buckling of compressed rods). The task was formulated in this way in [11]: minimize:

$$F(\mathbf{X}) = \sum_{j=1}^n x_j, \quad n = 8. \quad (15)$$

At the performance of restrictions:

$$D(\mathbf{X}) = \begin{vmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{vmatrix} = 0. \quad (16)$$

Parameters r_{ki} , included in the determinant (16), is a complex transcendental function variables x_i and are defined by the formulas given in [11].

Table 1
The results of the weight optimization of rod system of frame-type

The optimum values of variables	Random search	The method of gradient projection
x_1	0,05	0,18731
x_2	0,05	0,18684
x_3	0,13637	0,218590
x_4	0,05	0,18865
x_5	0,05	0,21311
x_6	0,0593	0,20662
x_7	0,3049	0,39264
x_8	0,24056	0,19728
$F(\mathbf{X})$	0,94115	1,79111

The problem was solved with a computer at such search parameters $R_1 = 1; R_2 = \dots = R_7 = 0,2; d_1 = d_2 = 0,6; d_3 = d_4 = 1; \varepsilon = 0,005$; the initial value of the dispersion of search was taken 0.5; the starting point of search $x_i = 0.25, 1, 2, \dots, 8$. On variables were imposed the restrictions $0.05 \leq x_i \leq 10$. The decision, which was obtained through 31 iterations, is provided in Table 1. For comparison shows the data out of work [11], which were obtained by Rosen's gradient projection method.

As can be seen from Table 1, the criterion of quality function $F(\mathbf{X})$, that has been determined by random search method, is much of less than criterion of quality, that has been obtained by the method of gradient projection. The effectiveness of random search method is equal 47.5%.

III. THE ALGORITHM WITH A NON-UNIFORM STRAIN OF DENSITY OF DISTRIBUTION OF SAMPLING POINTS

The analysis of ASE work [2] allowed to establish one of the feature associated with the method of determining the function $p_N(\Xi)$. As already noted, for the interior points of the ellipsoid, the center of which is located at the point, corresponding to the last best sample, the function $p_N(\Xi) > 0$. The dimensions of ellipsoid, commensurability the length of its axes with dimensions of search area is a measure of inertia and globality of algorithm.

With a relatively small size of the ellipsoid the search stops in a local minimum. If the axis sizes are much larger than dimensions of the area of search, then there is a high probability of the appearing of sampling points behind the borders of region and in the presence of the type of restrictions $0 \leq x_j \leq 1, j = 1, 2, \dots, n$, \square exists a high probability of selecting a limit point.

Based on these arguments, we construct an algorithm independent random search with a function $p_N(\Xi) \geq 0$ for internal points of the unit hypercube $x_j \in [0,1]$, and $p_N(\Xi) = 0$ for all the other points that do not satisfy the condition $x_j + \xi_j \in [0,1]$. The density $p_N(\Xi)$ is formed by of coordinatewise reflection the multidimensional normal law with zero mathematical expectation and identical variance σ_N in all axes. At this is displayed only a part of the initial of the normal law, that is enclosed in the cube with a side of $[-1,1]$. The boundary of this hypercube is displayed on the border with the side of $[0,1]$, and the center of initial distribution law \square is placed to a point \mathbf{X}_N , is received resulting from previous research stages.

Let $\Xi_N = (\xi_{N,1}, \xi_{N,2}, \dots, \xi_{N,n})$, the coordinates $\mathbf{r}_N = (r_{N,1}, r_{N,2}, \dots, r_{N,n})$ of a normal distribution with parameters $(0, \sigma_N)$ and coordinates of the point \mathbf{X}_N belong to interval $[0,1]$, then the reflection will look like this:

$$\xi_{N,j} = \begin{cases} x_{N,j}, r_{N,j} & \text{if } -1 < r_{N,j} \leq 0 \\ (1 - x_{N,j})r_{N,j} & \text{if } 0 < r_{N,j} \leq 1 \end{cases} \quad (17)$$

The function $p_N(\Xi)$, obtained at this, has the property that all direction coming out of the point \mathbf{X}_N , identical on probability, however, the probability density in any direction decreases depending on the distance to the boundary of the unit hypercube with sides [0,1].

$$\sigma_{N+1} = d_1 \sigma_N + d \| \mathbf{X}_{N+1} - \mathbf{X}_N \|, \quad 0 < d_1 \ll 1, \quad d_q > 0. \quad (18)$$

Since the initial normal law defined on the whole space, and displays only a part without "tails", let us limit from above the change $\sigma_N \leq \bar{\sigma}$ so that:

$$\frac{2}{\bar{\sigma} \sqrt{2\pi}} \int_{-\infty}^{-1} x \exp\left(-\frac{r^2}{2\sigma_N^2}\right) dr = \frac{\varepsilon_3}{2}, \quad 0 < \varepsilon_3 \ll 1. \quad (19)$$

At the practical implementation of the algorithm can be accepted $\bar{\sigma} = 0,3$ (from the condition $(3\bar{\sigma} < 1)$).

If $\sigma_N \geq \sigma(\varepsilon_1)$ it is easy to make sure in the validity of the assertion that a sequence $\{\mathbf{X}_N\}$ satisfies (1.42).

IV. THE GULLIED ALGORITHM OF INDEPENDENT GLOBAL SEARCH

The proposed global search algorithm with controlled density of distribution within allowable area as well as the algorithm of the sliding ellipse is the algorithm of independent search which is able to find a global extremum. However, unlike the ASE this algorithm allows to find extremum on the bottom of ravines arbitrary shape. For this purpose is introduced the basis, which is rotated, and allows to orient the axis along the bottom of curved ravine at each step. The difference between the considered algorithm of ASE is also in the selection of the distribution of test density. For ASE the distribution density $\varphi(\Xi)$ takes positive values at interior points of the ellipsoid with a center in the last better point of search. The measure of inertia and necessary condition of globality at the using of ASE are the dimensions of ellipsoid - the commensurateness the length of its axes with the dimensions of the search area. If the axis sizes are much larger than the area D (the area of the existence of the objective function), there is a high probability of appearance the trials overseas area, and at the presence of constraints such as:

$$x_j^- \leq x_j \leq x_j^+, \quad j = 1, 2, \dots, n, \quad (20)$$

a high probability of selecting a limit point of type x_j^\pm .

At each $\sigma_N > 0$ for all points of this hypercube $p_N(\Xi)$ is different from zero. In the process of search the dispersion is proposed to change by the rule:

In view of these considerations was created the algorithm, which uses $\varphi_N(\Xi) \geq 0$ the on inner point of the region D (20) and $\varphi(\Xi) = 0$ beyond [12].

Based on the statement of the problem of non-linear mathematical programming [13], we will look for a function $F(\mathbf{X}) = F(x_1, x_2, \dots, x_n)$ defined in a connected region D \square n -dimensional Euclidean space E^n and having in D finite number of extremums, the point \mathbf{X}^* of the minimum of the function or close to her the point \mathbf{X}^{**} , for which $\|\mathbf{X}^{**} - \mathbf{X}^*\| < \varepsilon$. General chart of algorithm is determined by the equations (2)–(3).

Below are two modifications of the algorithm.

Algorithm №1. Without loss of generality, we accept $x_j^- = 0$; $x_j^+ = 1$, ($j = 1, 2, \dots, n$). The density of distribution of the random vector Ξ_N we construct by means of coordinatewise reflection of n -dimensional normal law with zero mathematical expectation and variance σ in all axes, at this displays only a part of the normal law, enclosed in the cube with a side of [-1,1]. The boundary of this hypercube is displayed respectively on the Hyper-parallelepiped (20), and the center of the original law - \square at the point which was received at previous research stages.

Let the vector components $\mathbf{T} = (\tau_1, \tau_2, \dots, \tau_n)$ are normally distributed with parameters $(0, \sigma)$, and \mathbf{X}_N - \square internal point \square n -dimensional of open unit cube $x_{N,j} \in [0,1]$, ($j = 1, 2, \dots, n$). Then the mapping of the constructing Ξ_N is:

$$\xi_{N,j} = \begin{cases} x_{N,j}\tau_j, & \text{if } -1 \leq \tau_j < 0 \\ (1-x_{N,j})\cdot\tau_j, & \text{if } 0 \leq \tau_j \leq 1, j=1,2,\dots,n. \\ 0, & \text{if } |\tau_j| > 1 \end{cases} \quad (21)$$

As for the algorithm with non-uniform deformation of the density distribution of sampling points (17)-(19), is introduced by the mapping (21) the function $\varphi_N(\Xi)$, which has the same property, namely: all destinations that go from point \mathbf{X}_N have the same probability. The permissible area (20), the probability density in any direction will decrease depending on the distance to the border. For all interior points (20) for each value $\sigma > 0$, the function of probability distribution $\varphi_N(\Xi) > 0$. We describe the sequence of operations required to move out of point \mathbf{X}_N to point \mathbf{X}_{N+1} .

$$\sigma_{N+1} = d_1\sigma_N + d_2\|\mathbf{X}_{N+1} - \mathbf{X}_N\| + \Delta_N \quad (22)$$

$$0 < d_1 < 1; \quad d_2 > 0; \quad \lim\|\Delta_N\| = 0; \quad \sum_{N=1}^{\infty}\|\Delta_N\|.$$

When implementing the algorithm on a computer the law has been chosen $\Delta_N = d_3 / N$, where $d_3 > 0$. Since the original law of the distribution of the random vector \mathbf{T} defined on the whole space, and displays only a part of it, then we'll introduce the limitation from above $\sigma_{N+1} \leq \bar{\sigma}$ so, that the following equation was performed:

$$\frac{1}{\sqrt{2\pi\bar{\sigma}}} \int_{-\infty}^{-1} \exp\left(-\frac{\eta}{2\bar{\sigma}^2}\right) d\eta = \frac{\varepsilon}{2}$$

For the numerical implementation of the algorithm is enough to choose $\bar{\sigma} = 0,3$.

Algorithm №2. The algorithm №1, described above, ensures reliable operation when searching for the global extreme of the function which has such lines of level of local extremes which can be approximated quite well with the help of hyper-sphere, for example

$$F(\mathbf{X}) = \sum_{i=1}^n x_i^2 + \sum_{i=1}^n \cos(Ax_i). \text{ This way, during the}$$

motion out of the point of local extreme in any direction the rate of increment $F(\mathbf{X})$ obtained approximately identical. In case of problems with the level lines of local extremes having a ravine, it is more expedient to introduce the basis which turns and different values of parameters along each axis.

2. With the help of the reflection (1.71) is calculated vector Ξ_N .
3. The test point $\mathbf{X}_{N+1} = \mathbf{X}_N + \Xi_N$ in which is measured by the value of the objective function $F(\hat{\mathbf{X}}_{N+1})$, is determined.
4. With the help of (1) is calculated \mathbf{X}_{N+1} , when subsequently is displayed the center of the normal law.
5. The magnitude of the variance is changed σ_N :

Let us designate as $\sigma_N = (\sigma_{N,1}, \sigma_{N,2}, \dots, \sigma_{N,n})$ the vector of parameters display of the law along the respective axes in the rotating basis, which on N – step is determined by rotation matrix $B_N = \{b_{i,j}^N\}$ ($i, j = 1, 2, \dots, n$). The introduction of the rotating basis allows the best way track the curved ravines, and the introducing a vector $\square\sigma_N$ – increase the probability of movement along the bottom of the ravine. These two operations are in a sense equivalent to single procedure of tension of space (20) in the direction of the last successful step. Therefore, this algorithm is a definite sequence of operations:

1. Is ruffled vector \mathbf{T} , at this $-1 \leq \tau_j \leq 1; \quad j = 1, 2, \dots, n.$
2. Using the coordinates of the vector \mathbf{T} is determined the trial point $\hat{\mathbf{Y}}_{N+1}$ in the ortho-normal basis of the space, which is determined by the matrix B_N . To be specific, can to accept as $B^{(0)}$ the unit matrix. At the same time such calculations are performed:

$$\hat{\mathbf{Y}}_{N+1,j} = \begin{cases} \delta_j^{(0)}\tau_j + \mathbf{Y}_{N,j}, & \text{if } \tau > 0 \\ \delta_j^{(1)}\tau_j + \mathbf{Y}_{N,j}, & \text{if } \tau \leq 0 \end{cases}$$

Where $\mathbf{Y}_N = B_N \mathbf{X}_N$, a $\delta_j^{(0)}$ and $\square \delta_j^{(1)}$ – respectively distances from the point \mathbf{X}_N along the

positive and negative directions of the j – axis of the rotating basis till the surface of the unit hypercube (20).

$$\delta_j^{(0)} = \min \delta_{kj}^{(0)}; \quad \delta_j^{(1)} = \min \delta_{kj}^{(1)}; \quad (1 \leq k \leq n);$$

$$\delta_{jk}^{(0)} = \begin{cases} (1 - x_{N,j}) : b_{jk}^N & \text{for } b_{jk}^N < 0 \\ -x_{N,j} : b_{jk}^N & \text{for } b_{jk}^N > 0 \end{cases};$$

$$\delta_{jk}^{(1)} = \begin{cases} -(1 - x_{N,j}) : b_{jk}^N & \text{for } b_{jk}^N < 0 \\ x_{N,j} : b_{jk}^N & \text{for } b_{jk}^N > 0 \end{cases}.$$

At $b_{jk}^N = 0$ is necessary accept $\delta_{jk}^{(0)} = \delta_{jk}^{(1)} = 0$.

3. We find the coordinates of trial points in a fixed basis $\hat{\mathbf{X}}_{N+1} = B_N^Y \mathbf{Y}_{N+1}$.
4. In the transition from the mobile to the fixed coordinate basis some coordinates of points $\hat{\mathbf{X}}_{N+1}$ may be outside the interval [0,1], so for these coordinates we install the nearest limit values. To navigate to a point \mathbf{X}_{N+1} we use the algorithm (1).
6. In the case, when $\|\mathbf{X}_{N+1} - \mathbf{X}_N\| > 0$, the position of rotating basis is changed. For this the new direction of first axis of basis is formed:

$$b_1^{N+1} = \frac{k_1 b_1^N + k_2 \frac{\mathbf{X}_{N+1} - \mathbf{X}_N}{\|\mathbf{X}_{N+1} - \mathbf{X}_N\|}}{\left\| k_1 b_1^N + k_2 \frac{\mathbf{X}_{N+1} - \mathbf{X}_N}{\|\mathbf{X}_{N+1} - \mathbf{X}_N\|} \right\|},$$

where b_1^N – the direction of the first axis of basis on N -th step; $\square k_1, k_2 > 0$ – factors that determine the inertia of the axis rotation b_1^N .

$$U(d_1, d_2, k_1, k_2, \mathbf{X}^{(0)}, \Omega) = k_3 N_\psi(\Omega) + k_4 \left(\|\mathbf{X}^{**}(\Omega) - \mathbf{X}^*\| - \delta \right), \quad (25)$$

where: $N_\psi(\Omega)$ – the number of trial points needed to reach the point $\mathbf{X}_N = \mathbf{X}^{**}$ at which $\sigma_N < \bar{\sigma}$; Ω – \square is a set of random numbers, with help of which was performed this implementation; $\square \delta > 0$ – the necessary search accuracy; $k_3 > 0$; and $\square k_4 > 0$; – weighted coefficients which establish the equivalence of sampling points, which are calculated to further reduce the

Construction of axis b_1^{N+1} , ($i \geq 2$) performed by the orthogonalization procedure [5].

At unsuccessful step the search continues in the same basis with a modified vector σ_N .

Example №1. Consider the problem of determining the global minimum of the function [2]:

$$F(\mathbf{X}) = x_1^2 + x_2^2 - \cos 18x_1 - \cos 18x_2 \quad (23)$$

$$-1 \leq x_1; \quad x_2 \leq 1 \quad (24)$$

and explore on it the search properties of algorithms described above.

The function (23) in (24) has 25 local minima and 10 gullies, the global minimum $\mathbf{X}^* = 0,0$.

The adduced above algorithms of random search type smoothing [7] in the integral form adapted for storage and use of previous experience, i.e. have the property of adaptation. In particular, the algorithm 2 has four parameters d_1, d_2, k_1, k_2 defining the probabilistic properties of directions and selecting the length of trial step. For each set of parameters, for a certain starting point $\mathbf{X}^{(0)}$ and a priori given the lower limit of the norm of vector $\sigma_N : \|\sigma_N\| \geq \bar{\sigma}$ we define the loss function:

difference $\|\mathbf{X}^{**} - \mathbf{X}^*\| - \delta$ on per unit ($k_4 = 0$ if $\|\mathbf{X}^{**} - \mathbf{X}^*\| \leq \delta$).

As the experience of numerical experiments, the value of the function (25), quite strongly depends on the sequence Ω and the starting point for a fixed set d_1, d_2, k_1, k_2 .

International Journal of Emerging Technology and Advanced Engineering

Website: www.ijetae.com (ISSN 2250-2459, ISO 9001:2008 Certified Journal, Volume 6, Issue 12, December 2016)

Therefore, as the objective function for the problem of choice of rational parameters of training we use $U(d_1, d_2, k_1, k_2) = M_{\Omega, X^0} \{U(d_1, d_2, k_1, k_2)\}$.

Then the problem of choosing the optimal training parameters for the considered object (23), (24) can be reduced to a mathematical programming problem: find a minimum of the function

$$U(d_1, d_2, k_1, k_2) \quad (26)$$

with restrictions:

$$d_i^- \leq d_i \leq d_i^+; \quad k_i^- \leq k_i \leq k_i^+; \quad i=1,2. \quad (27)$$

To solve this problem was used algorithm №1. The search process took place at such initial data:

$$\sigma = 10^{-3}; \delta = 10^{-2}; d_1^- = 0,2; d_2^- = 0,4; k_1^- = 0,1;$$

$$k_2^- = 0,1; d_1^+ = 5; d_2^+ = 10; k_3 = 10^{-2}; k_4 = 10.$$

$$\text{for } \|\mathbf{X}^{**} - \mathbf{X}^*\| > 10^{-2}$$

For each set d_1, d_2, k_1, k_2 the search $\mathbf{X}^{**}(\mathbf{X}^{(0)}, \Omega)$ has began out of five different starting points, evenly distributed along the constraints (24). At this with the purpose of obtaining the averaging over the random number sequence, this procedure was repeated 10 times for each of the starting point. As a result were found these parameters:

$$d_1 = 0,976; \quad d_2 = 0,38; \quad k_1 = 1,79; \quad k_2 = 0,47.$$

Using the procedure described above were chosen the same settings for ASE algorithm for object (1.23)–(1.24). The numerical values of the parameters were respectively ASE:

$$d_1 = 0,753; \quad d_2 = 0,621; \quad k_1 = 0,91; \quad k_2 = 0,83$$

With these sets of parameters for the algorithm №2 was obtained \mathbf{X}^{**} for 290 trial points on average (standard middle-quadratic deviation was 93 trials) and for ASE – 320 (standard middle-quadratic deviation – 78 trials).

$$F(\mathbf{X}) = 100(x_1 - x_1^2)^2 + (1 - x_1)^2 + 90(x_4 - x_3^2)^2 + (1 - x_3)^2 + 10,1[(x_2 - 1)^2 + (x_4 - 1)^2] + 19,8(x_2 - 1) \times (x_4 - 1) \rightarrow \min; \quad -10 \leq x_i \leq 10, \quad (i=1,4).$$

The equivalence of the losses on the search for algorithms is not accidental and is explained by a certain similarity of density distributions $\varphi_N(\Xi)$, as well as sufficiently close values of the factors of forgetting the prehistory of search.

Example №2. The choice of rational parameters of learning of algorithm №2 for gullied function is carried by the example of the optimization of function

$$F(\mathbf{X}) = \alpha_1(x_2 - \alpha_2 x_1^2) \beta + (x_1 - \alpha_2^2)^2$$

at different values $\alpha_1, \alpha_2, \beta$, which respectively define the steepness of the walls of the ravine, the measure of non-linearity the lines of level of the bottom of and the degree of reduction of curvature $f(\mathbf{X})$ near the line

$x_2 = \alpha_2 x_1^2$. Calculations were performed for $10 \leq \alpha_1 \leq 1000$; $0,1 \leq \alpha_2 \leq 5$ with $\beta = 2$ and $\beta = 4$.

In the result of studies on the rational minimize of losses were found the intervals of parameters of the algorithm №2 for the function with gullied lines of level:

$$0,95 \leq d_1 \leq 0,965; \quad 1,5 \leq d_2 \leq 1,8; \quad (28)$$

$$2 \leq k_1 \leq 2,7; \quad 0,55 \leq k_2 \leq 0,7 \quad (29)$$

The losses on the search practically did not depend on the parameter β , and for $100 \leq \alpha_1 \leq 1000$; $0,5 \leq \alpha_2 \leq 3$ are accounted for 350–500 computing of function. When $\alpha_1, \alpha_2 \leq 0,5$ or $\alpha_2 \geq 4,5$ the losses decreased and, for example, accounted for $\alpha_1 = 10$; $\alpha_2 = 0,1$ only 70–120 trials. With an increase the losses on the search also decreased, as the approach of the branches of the parabola $x_2 = \alpha_2 x_1^2$ the search has "jumped" from one branch to another without moving along the bottom of the ravine.

Example №3. The influence of parameters (28), (29) on the quick of action of the algorithm №2 is investigated by the example of the following mathematical programming problem:

The condition of stop has been selected $F(\mathbf{X}) \leq 10^{-2}$ In calculations, performed without using of intervals (28), (29) at different sets of parameters d_1, d_2, k_1, k_2 , the averaged losses on the search made up 13000-18000 trials and at set of parameters with account (28), (29) the averaged losses on the search made up 5000-7000 trials.

Analyzing the results, we can draw the following conclusions:

1. Displaying the normal distribution of the law of random trials on the entire range of variation of the control parameters increases the reliability of the search at each step.
2. The using of the basis with rotating axes was allowed to construct an algorithm independent global search, designed to optimize the gullied-type functions with local extremes, lying at the bottom of the gully curved form.
3. Introduction of the operation of preliminary settings of gullied algorithm on the test function improves the probability properties of the selection of a successful direction, and the length of the working steps and thus increases the speed of search.

$$\begin{cases} \tilde{\mathbf{X}}_{i+1} & \text{if } F(\tilde{\mathbf{X}}_{i+1}) < F(\mathbf{X}_i) \\ \tilde{\mathbf{X}}_{i+1} & \text{if } F(\tilde{\mathbf{X}}_{i+1}) \geq F(\mathbf{X}_i) > F(\bar{\mathbf{X}}_{i+1}) \\ \mathbf{X}_i & \text{if } F(\bar{\mathbf{X}}_{i+1}) \geq F(\mathbf{X}_i) \end{cases} \quad (33)$$

Where: $\hat{\mathbf{X}}_{i+1} = \tilde{\mathbf{X}}_i + H_i \Xi$; $\hat{\mathbf{X}}_{i+1} = \tilde{\mathbf{X}}_i - 2H_i \Xi$
 - Ξ random vector whose components are distributed uniformly over the unit n -dimensional hyper-sphere;
 $H = (h_1, h_2, \dots, h_m)$ - the value of the test step.

The algorithm consists of two sub-algorithms. Sub-algorithm №1 chooses the starting point and sub-algorithm №2 finds the solving of problem.

Sub-algorithm №1. Selection of the starting point.

The difficulty of solving the problem (70) for a sufficiently complex constraints (31) and (32) significantly is increased due to the need to "manual" selection of the initial (starting) point belonging to the permissible area.

In [14] it is proposed automatic selection of the starting point of the search process using the following procedure. In the region Q , which belongs to the space E_n , randomly selected the starting point, which generally does not satisfy the constraints (31) and (32). Out of this point as center, is constructed hyper-sphere of predetermined radius R^{m_0} .

V. GLOBAL SEARCH ALGORITHM "WITH THE RETURN AFTER A FAILED STEP"

This algorithm has been suggested by the authors of article [14].

Formulation of the problem. Let \square in n -dimensional Euclidean space E_n in some domain given a continuous function $F(\mathbf{X}) = F(x_1, x_2, \dots, x_n)$, i.e. $Q = (\mathbf{X} \in E^n)$.

Is considered the solution of the nonlinear programming problem: find the extreme of function

$$F(\mathbf{X}) \rightarrow \min, \quad \mathbf{X} \in Q \quad (30)$$

at the performance of restrictions:

$$g_{k_1}(\mathbf{X}) = 0, \quad (k_1 = 1, 2, \dots, m_1) \quad (31)$$

$$g_{k_2}(\mathbf{X}) = 0, \quad (k_2 = 1, 2, \dots, m_1) \quad (32)$$

Consider a modification of the algorithm of random search "with the return after a failed step" [2] \square - the randomized algorithm with "double return", which can be represented as follows [14]:

After that, by using the pseudorandom numbers produced \square -dimensional realization of random vector $\Xi_i^{(1)} = (\xi_{i1}^{(1)}, \xi_{i2}^{(1)}, \dots, \xi_{in}^{(1)})$ and in the area $Q \subset E^n$ is determined the random point:

$$\begin{aligned} \mathbf{X}_{i+1} &= \mathbf{X} + \Xi R_i^{(m)}; \\ R_i^{(m)} &= (r_{i1}^{(m)}, r_{i2}^{(m)}, \dots, r_{in}^{(m)}) \end{aligned} \quad (34)$$

If the point \mathbf{X}_{i+1} satisfies the constraints (31) and (32), this point is accepted as the starting point; otherwise are performed consistently all the given S realizations of the random vector $\Xi_i^{(2)}, \dots, \Xi_i^{(S)}$ with the calculation of the formula (34), and is checked constraints (31) and (32). If all the attempts of S have not been successful, then the "contraction" of the hyper-sphere radius by the formula

$$R^{m_{i+1}} = R^{m_i} K, \quad (0, 1, 2, \dots), \quad (35)$$

wherein $\square K$ – contraction coefficient, which is selected depending on the complexity of constraints and the sizes of the search area for each of the variables (usually ranges from 0.8 to 0.98). Likewise, \square is selected the number of tests: $S = (10-30)$. Thereafter is performed a new series of tests of the random point on the surface of the new hyper-sphere. If during the search starting point $R^{m_j} - R^{m_{j+1}} < K_R$ (K_R – criterion values hyper-sphere radius), then the process of variation of hyper-sphere radius stops. Then, according to the formula

$$R^{m_{j+1}} = R^{m_j} K, (0, 1, 2, \dots, t-1) \quad (36)$$

is constructed a new sequence of hyper-spheres. Each time at the changing of the radius are performed the calculations according to the formula (34) with check of constraints (31) and (32). If in the process of construction of $i - j$ hyper-sphere has not been determined the starting point of satisfying the given restrictions, then out of the last of S \square point, lying on the surface of the i -hyper-sphere, is done the "big step" in a random direction, i.e.

$$R^{m_0} = R^{m_t} + L^k, (k = 0, 1, 2, \dots, q). \quad (37)$$

The resulting random point is the center of the creation of a new hyper-sphere and a new cycle of the search of starting point. If there is a restriction only of the type (31), the system changes the choice of starting point: at first by program way is formed the auxiliary objective function in the form

$$F(\mathbf{X}) = \sum_{k_1=1}^{m_1} f_{k_1}^2(\mathbf{X}), \quad (38)$$

and then is performed the search of the starting point, i.e. minimum of the function (38), and is performed sub-algorithm №2.

Sub-algorithm №2. Solution of non-linear programming problem. The work of sub-algorithm №2 starts after finding the starting point. Out of it in a random direction are performed successively the series of S of test steps, the length and the number of which is selected depending on the complexity of the objective function and constraints: the more considerable is their nonlinearity, the greater is selected the value of initial step, which is called the "small step".

$$\min F(\mathbf{X}) = x_1^2 + x_2^2 - \cos(18x_1) - \cos(18x_2), \quad -1 \leq x_i < 1, (i = 1, 2). \quad (42)$$

The example was taken from monograph [2]. Function (42) has 10 gullies and 25 lows (the global minimum is located at the point with coordinates $x_1 = x_2 = 0$).

The first from series of steps leading to a decrease in the objective function is called the "working step", and thus obtained was a point taken as the new starting point (34). If neither step in the series of random sampling is not successful, then there is variation in the length of a "small step" towards its reduction:

$$H_i^{(j+1)} = H_i^{(j)} K \quad (39)$$

where $i = 1, 2, \dots$ – the number of working step; $j = 1, 2, \dots, S-1$ – \square number of variations. If $H_i < k_m$ (k_m – "criterion of the length of "small step") or $|F(\mathbf{X}_{i+1}) - F(\mathbf{X}_i)| < \varepsilon$ (ε \square – specified accuracy), then out of the point \mathbf{X}_{i+1} is performed the "big step" in a random direction:

$$\hat{\mathbf{X}}_0 = \mathbf{X}_{i+1} + L\mathbf{E}, \quad (40)$$

\square where $L = (l_1, l_2, \dots, l_n)$ – the value of this step. At this the length L is selected so as to re-cover approximately 3/4 of the area:

$$\mathbf{A} \leq \mathbf{X} \leq \mathbf{B}, \quad (41)$$

Where \mathbf{A} and \mathbf{B} – the vectors \square of geometric constraints imposed on the control parameters (vector \mathbf{X}).

In the absence of the condition (41), the length of the "big step" is defined arbitrarily. Obtained in this way point $\mathbf{X}^{(0)} \in Q$ is checked whether it belongs to the permissible region. The production of a certain number of random tests at failed "small step" in both directions allows a retrieval system, while keeping the same step, rise up the slope, and thus overcome the mountain ranges of the objective function. Consequently, the search itself thus gains global properties. In practice, there is a situation when the function (30) is multi-extreme and the restrictions (31) and (32) are complicated. To overcome this type of obstacles the sub-algorithm №2 includes addition, based on the use of (41). This allows in case of hit of the search system in any of local extremes get out in another zone using the "big step".

In order to evaluate the effectiveness of the algorithm it was conducted numerical experiment on a computer. The task of mathematical programming if formulated as follows: find

The search character is shown in Fig. 1. Search was carried using two algorithms – ASE and the algorithm "with the return of a failed step."

The solid line shows the trajectory algorithm ASE with variable axial correlation. The minimum of objective function was achieved per 142 steps, 10 trials each. The initial starting point was selected randomly. Bold dotted line shows the trajectory of the finding of the minimum of function (42). The search was performed out of the randomly selected point with the initial vector

$$\alpha_0 = \left(\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}} \right); R_1 = 0,7; R_2 = 0,2. \quad \text{The}$$

minimum of function was found per 27 steps to 10 trials for each step.

The trajectory of search engine using the algorithm "with the return after a failed step" is shown in broken lines. The initial length of "small steps" has been selected $h_1 = h_2 = 0,5$, and the length of the "big step" $l_1 = l_2 = 10$ (this was done in order to verify the response of algorithm on the going out the random point out of the permissible area). The number of "big steps" is five, that is, while six options were examined simultaneously. Table 2 shows the number of working steps required in order to find a global minimum in each of the options.

Table 2
The losses on search for a global extremum of the function (1.114) using the algorithm "with the return after a failed step"

Variants	1	2	3	4	5	6
Number of steps	21	18	21	21	14	19

At each step was given to 15 trials at $\varepsilon = 10^{-6}$. The studies have shown that due to the relative simplicity of the above adduced randomized algorithm (two parameters – the \square length of initial step and the number of random tests), it has a much greater adaptability in solving complex problems of nonlinear programming in comparison with known algorithms of global random search [15].

- a. ability to find the global extreme of nonlinear function;
- b. a relatively high speed;
- c. the ability to solve the multivariable problems of nonlinear programming with a large of computer memory savings.

VI. SYNTHESIS OF THE METHODS OF RANDOM SEARCH AND DYNAMIC PROGRAMMING IN THE OPTIMIZATION OF MULTIVARIABLE SYSTEMS

In addressing each specific optimization problem it is necessary to select such a mathematical method, which would lead to outcomes with the least losses in the calculations and, at the same time, would give the opportunity to get as much information about the desired solution.

Modern optimization methods quite well resolves the certain classes of problems. However not exists the universal method for the decision all tasks. For example, dynamic programming method [16] is well suited for solving integer optimization problems of multistage processes with a small number of control variables (for example, finding a one-dimensional profile of lengthy constructions). Dynamic programming is one of the few methods of optimization, at the application of which the resulting solution corresponds to global extremum. However, it should be understood that the method has the drawbacks caused by "curse of dimensionality" [16], and therefore its use is often difficult due to the limited memory volume and performance of computer devices.

The random search method, as mentioned above, is very effective for solving large-scale problems with complex nonlinear constraints written in the form of a system of inequalities. A random search is not always finds the global extremum, and finds a state close to it.

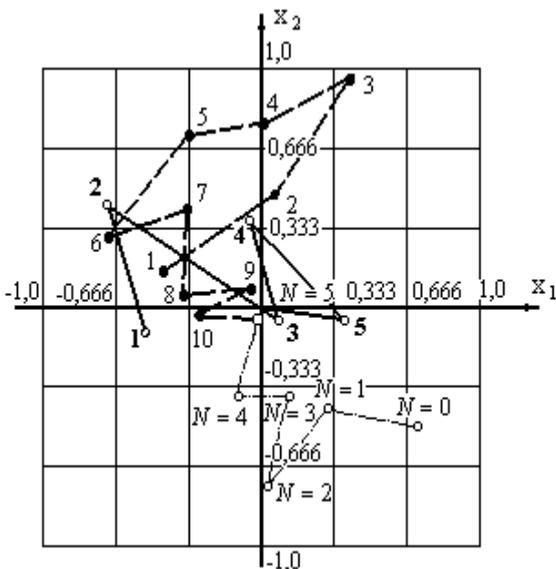


Fig. 1. The trajectory of the search engine of minimum function (42) using the algorithm ASE and algorithm "with the return after a failed step"

The authors of the article [14] claim that the software implementation of the proposed algorithm and the experimental verification on a computer allow to note the following main peculiarities of algorithm:

Therefore there is a need for such a combination of these techniques at certain stages of solving the optimization problem in which ideally would combine the main advantages of both methods [17].

Consider a dynamic system described by a nonlinear vector differential equation:

$$\frac{d\mathbf{X}(t)}{dt} = f[\mathbf{X}(t), \mathbf{U}(t)], \quad (43)$$

here: $\mathbf{X}(t) = \{x_1(t), x_2(t), \dots, x_n(t)\}$ – the state vector at time t and $x_i(t)$ – the state variables ($i = 1, 2, \dots, n$); $\mathbf{U}(t)$ r -dimensional vector of control, $r \leq n$ and f is a vector-valued function of $\mathbf{X}(t)$ and $\mathbf{U}(t)$. Regarding the components f let us assume that they are differentiable or piecewise-differentiable in all arguments. The initial state of the system at the time t_0 given as:

$$\mathbf{X}(t) = \mathbf{X}_0. \quad (44)$$

The certain restrictions on the influence of control is imposed in such a way that:

$$g[\mathbf{X}(t)] = \min \left\{ \int_{t_0+(k-1)\Delta}^{t_0+k\Delta} L[\mathbf{X}(t), \mathbf{U}(t)] dt + g[\mathbf{X}(t_0+k\Delta)] \right\}, \quad (47)$$

where $\mathbf{X}(t)$ is determined from the solution (43) on the interval $[t_0 + (k-1)\Delta, t_0 + k\Delta]$,

$$g_N[\mathbf{X}(t)] = \min_U \{ L[\mathbf{X}_N(t_j), \mathbf{U}_N(t_j)] \Delta + g_{N+1}(\mathbf{X}_{N+1}^*) \}, \quad N = 1, 2, \dots \quad (48)$$

$$\mathbf{X}_{N+1}^* = \mathbf{X}_N + \Delta f_N[\mathbf{X}_N(t_j), \mathbf{U}_N(t_j)], \quad (49)$$

Solution of recurrent functional equations (48)-(49) is performed by "inversion" method, starting from $N = N$ and ending $N = 1$. At this at each step of dynamic programming in the process of solving is performed the search of control parameters. Search can be performed by a simple scanning of control parameters with some sufficiently small pitch. Such a procedure is justified for vector control, consisting of a single component. For two or more components of the control vector, this method of specifying of changing controls is no effective, because it requires a huge number of different combinations. In this case, should use the algorithms of random search.

REFERENCES

[1] Mockus I.B. Multiextremal problems in the designing. – M.: Nauka. □1967. □ 276p.

$$\mathbf{U}(t) \in R \quad \text{for all } t \quad (45)$$

where R – space of permissible controls.

We formulate the following problem in terms of optimal control: it needs to find a permissible vector control $\mathbf{U}(t)$, which transfers the system from the initial state \mathbf{X}_0 to the final state $\mathbf{X}(t_j)$, belonging to the set of possible states S at a minimum of functional:

$$J(\mathbf{U}) = \int_{t_0}^{t_j} L[\mathbf{X}(t), \mathbf{U}(t)] dt. \quad (46)$$

Posed variational problem of optimal control will be considered as a multi-step process of making optimal decisions, to which then we'll apply the theory of dynamic programming [16].

Let $g[\mathbf{X}(t)] = \min J(\mathbf{U})$ – the minimum value of the functional (46) under the assumption that a certain initial state is characterized by the value (44), the conditions (45) are executed and the optimal strategy is used. Then, according to Bellman optimality principle [16]. The functional equation of dynamic programming can be written as follows:

($K = 1, 2, \dots, j$) ($k = 1, 2, \dots, j$), in which, moreover, is required the execution of conditions (45).

In the discrete setting equation (47) takes the form:

- [2] Rastrigin L.A. Statistical methods of search.– M.: Science, 1968. □ 376 p.
- [3] Volynskij E.L., Pochtman Y.M. About one algorithm of random search for solving multi-extremal problems // Math. USSR Academy of Sciences. Technical cybernetics. – 1974. □ № 1. □ p.55-60.
- [4] Kolmogorov H. On analytical methods in probability theory // UMN. □ T. V. □ Vol. 1. – 1938. – p.246-252.
- [5] Gantmakher F.R. The theory of matrices. □ Nauka, Moscow, 1967. □ 576 p.
- [6] Matyash I. Random optimization // Automation and telemekhanika. – 1965. □ №2. p.246-253.
- [7] Antonov G.E., Katkovnik V.Y. The method of synthesis of a class of random search algorithms // Automation and telemekhanika. – 1971. □ №6. p.154-157.
- [8] Pshenichny B.N., Danilin Y.M. Numerical methods for solving extreme problems. – M.: Nauka, 1975. – 317 p.
- [9] Rubinstein Y.S. Stopping random search algorithms // Automation and Computer Science. – 1971. □ Vol. 3. □ p.58-64.

International Journal of Emerging Technology and Advanced Engineering

Website: www.ijetae.com (ISSN 2250-2459, ISO 9001:2008 Certified Journal, Volume 6, Issue 12, December 2016)

- [10] Zoytendeyk G. The methods of possible directions. □ – M.: IL, 1961 □, 196 p.
- [11] Baublis P.S., Remishkauskas M.Y., Tsypinas I.K. Application of the method of gradient projection for optimization of elastic frame that exposed to buckling // Lithuanian mechanical collection. – Vilnius: Minthis, 1 (6). – □ 1970. □ p.234-□239.
- [12] Volynskij E.I., Filatov G.V., Khablo T.V. About one of gullied global search algorithm // Automation and Computer Science. – Riga: Zinatne. □1976. □ №2. □ p.50-□54.
- [13] Bocharov I.N., Fel'dbaum A.A. Automatic optimizer for finding the least minimum out of few lows // Automation and telemehanika. □ –1962. □ № 3. □ T.23. – p.67-□73.
- [14] Gurvich I.B., Zakharchenko V.G., Pochtman Y.M. Randomized algorithm for solving the problems of nonlinear programming // Math. USSR Academy of Sciences. Technical cybernetics. – 1975. □ № 5. □ p.30 □-33.
- [15] Rastrigin L.A. The systems of extreme control. □ M.: Nauka, 1974. □ 630p.
- [16] Bellman R. R., Dreyfus S. Applied dynamic programming problem. – M.: Nauka, 1985. – 446p.
- [17] Baranenko V.A., Pochtman Y. M., Filatov G.V. About the joint use of dynamic programming techniques and random search in the problems of optimal design // Building mechanics and calculation of constructions. □ M.: Stroyizdat. □ №1. – 1973. □ p.3-□6.