International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (ISSN 2250-2459, ISO 9001:2008 Certified Journal, Volume 6, Issue 9, September 2016)

Random Search as the Method of Nonlinear Programming. Algorithms of Random Search

George Filatov, Professor, Doctor of Techn. Sciences
Dnepropetrovsk State Agrarian-Economic University, Ukraine

***Abstract* — In the work is adduced the brief overview of existing algorithms accidentally search. Considered the definition of random search method as the method of nonlinear programming. Is discussed the strategy of search of, his opportunities of decisions one- and multi-extreme problems. Is considered two classes algorithms: independent and wandering. Some specific of local and global search algorithms are offered.**

***Keywords* — Random Search Method, Nonlinear Programming, Global Search**

I. Introduction

We formulate the task of search engine optimization as a multi-step process of gathering information and making a decision based on the information received. For this purpose, we shall introduce the concept of space and the concept of search area. Unlike the real concepts of space (one-, two-, or three-dimensional) the mathematics operates by $n-$dimensional space or hyperspace. The object which is subjected to optimization, normally represent the system that has some the inputs and outputs. The inputs are determined by $n-$dimensional vector of control $\mathbf{X}=(x_1,x_2,...,x_n)$. The dimension of the space defined by the number of independent control variables that change their values during the search. In the graphic sense, each of the control variables vector **X** defines one of the coordinate axes, whose beginning is in some initial point of the parameter space, and which can move in the permissible zone in the search process.

Optimized object is characterized by scalar output, the value of which is determined by the vector of control variables $\mathbf{X}$:

$$Q=F(\mathbf{X}), \tag{1}$$

where: F – is some scalar deterministic function of a vector argument. This function determines the quality score of the object, so this function is called the function of the target or the target function. The objective function with the help of the vector of control variables $\mathbf{X}=(x_1,x_2,...,x_n)$ can express the quality of the test process on the various stages of the search. Therefore, the objective function is also called the criterion of quality. Typically, the objective function is a mathematical model of the physical, energy, economic or other state of the object or process.

The permissible area of control variables is allocated in the space by restrictions. The restrictions can be geometric, physical, economic, energy or have the any other content in the dependence on the problem statement. We introduce the vector restrictions $\mathbf{G}=(g_1,g_2,....,g_m)$.

Under the search we will be called the process of selection from great set of elements $\mathbf{X}$ one element $\mathbf{X}^*$, which satisfy certain condition:

$$F(\mathbf{X})<F^*+\varepsilon, \tag{2}$$

where ε – a predetermined number.

The aim of the search is to reach an extreme state of the object and the determination of such a vector of control variables that determine this condition.

The search process is divided into stages. At each step, is executed gather information and deciding on further research direction. Thus, the search is a multi-stage process of gathering information and making a decision that is based on the information received [1].

The problem of mathematical programming can be formulated as follows: find a vector of control variables

$$\mathbf{X}^*=(x_1^*,x_2^*,....,x_n^*), \tag{3}$$

that deliver the extreme objective function:

$$F(\mathbf{X})=F(x_1,x_2,....,x_n) \tag{4}$$

under the condition of the execution of restrictions:

$$\mathbf{G}(\mathbf{X})=\{g_1(\mathbf{X}),g_2(\mathbf{X}),....,g_m(\mathbf{X})\}, \tag{5}$$

where restrictions $g_i(\mathbf{X})$ can take the form: $g_i(\mathbf{X})\le 0$; $g_i(\mathbf{X})=0$ or $g_i(\mathbf{X})\ge 0$.

The purpose of the search – condition X * – is not always achieved. However, this state is characterized by a limit, which can to solve the problem.

In Fig.1 shows an example of a graphical representation of two-dimensional problem of mathematical programming. Thin lines show the line-level of objective function, thick lines – restrictions. The same figure shows the trajectory of movement of the search system to the extreme point.

International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (ISSN 2250-2459, ISO 9001:2008 Certified Journal, Volume 6, Issue 9, September 2016)

In the process of collecting of information the search may to adapt. The role of adaptation perform the training, that is, to establishing of succession between two successive steps. The learning of search engine is the result of accounting prehistory factor that determines the impact of the previous steps to choose from the following search steps. The process of learning can take place on such a scheme: on stage m from the point are performed the tentative steps and is defined the objective function at a point chosen randomly.

The information thus obtained about the behavior of the objective function in the vicinity of the initial point $\mathbf{X}_{N-1}$ allows to take the decision to move the system to another point $\mathbf{X}_N = \mathbf{X}_{N-1} + \Delta\mathbf{X}_N$. At this displacement $\Delta\mathbf{X}_N$ is determined in accordance with the chosen function of learning [1]:

$$\Delta\mathbf{X}_N = \Phi\left(\mathbf{U}_{N-1}, \mathbf{X}_N^{(1)}, ..., \mathbf{X}_N^{(mN)}\right), \qquad (6)$$

where $\mathbf{U}_{N-1}$ – prehistory factor.

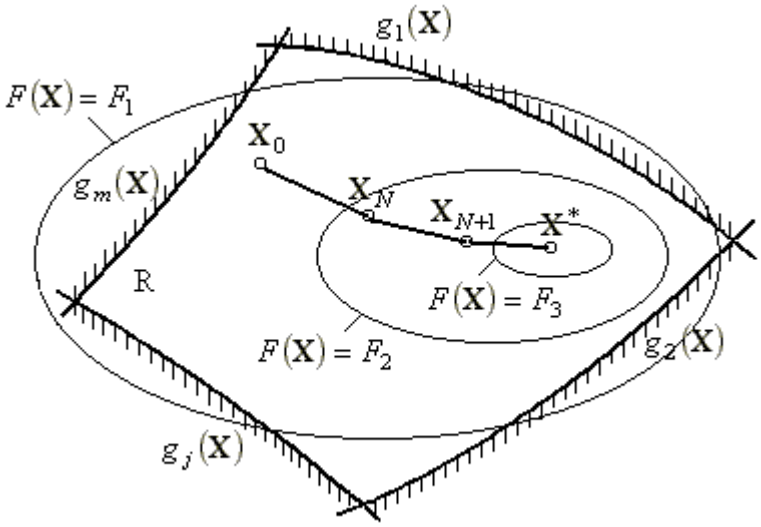

Fig. 1. The two-dimensional space of control variables of system, that is subjected to optimization

The random selection of system state, in which is determined the objective function occurs in accordance with the n –dimensional law of distribution:

$$p\left(\mathbf{X}_N^{(i)} / \mathbf{U}_{N-1}, \mathbf{X}_N^{(1)}, ..., \mathbf{X}_N^{(i)}\right), \qquad (7)$$

which may be formed during the search process.

The process of collecting information may also adapt. The adaptation occurs by the change in the distribution density (6) during collecting information. This means that the next steps in the parameter space are arranged so as to carry the most information about the gradient direction.

During the random search, the presence of restrictions (5) can be carried out by different methods. A universal method is called "the method of penalty functions" [2]. This method consists in the fact that the object of optimization is "penalized" by a corresponding increase of the objective function under the violation of the restrictions and, the greater is the violation of restriction, the more is "penalty". For this, is created a new Quality Score, which at violation of restrictions leads to the minimization of restrictions, i.e.

$$f(\mathbf{X},\mathbf{G}) = \begin{cases} F(\mathbf{X}) & \text{при} \quad \sum_{i-1}^{k}(1-\operatorname{sign} g_i) = 0 \\ F(\mathbf{X}) + \sum_{i-1}^{k}\lambda_i g_i(1-\operatorname{sign} g_i) & \text{при} \quad \sum_{i-1}^{k}(1-\operatorname{sign} g_i) > 0 \end{cases} \qquad (8)$$

the gap of function $f(\mathbf{X},\mathbf{G})$ on restrictions when $g_i = 0$, can be eliminated as follows:

$$f(\mathbf{X},\mathbf{G}) = F(\mathbf{X}) + \sum_{i-1}^{k}\lambda_i g_i(1-\operatorname{sign} g_i), \qquad (9)$$

International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (ISSN 2250-2459, ISO 9001:2008 Certified Journal, Volume 6, Issue 9, September 2016)

where λ_i – the weights coefficients of corresponding restrictions, which are chosen great enough.

The following are some of algorithms for local and global random search, which solve the problem of mathematical programming.

II. Local Random Search Methods

2.1. The algorithm on the best test with lowering

As a starting point in the parameter space R we select the point $\mathbf{X}_N^{(m)} \in R$. From the point $\mathbf{X}_N^{(m)}$ of radius equal a to the length of the working step, we build hyper-sphere and select on the surface of which the m random points $\mathbf{X}_{N+1}^{(m)} = \mathbf{X}_N^{(m)} + a\Xi^{(m)}$. From these points the system selects such points, those for which is performed the condition $\mathbf{X}_N^{(m)} \in R$. In each of the selected points we determine the value of the objective function $F(X_i)$ $(i = 1,2,....,m)$ and from these points select such that satisfy the condition:

$$F\left(\mathbf{X}_N^{(m)}\right) > F\left(\mathbf{X}_{N+1}^{(m)}\right). \qquad (10)$$

Finally, from the thus obtained points, we select one, for which a relative increase in the objective function is maximum:

$$\max_m \left| \frac{F_N^{(m)} - F_{N+1}^{(m)}}{\sum_{i=1}^{n} (a\xi_i)^2} \right|. \qquad (11)$$

The point $\mathbf{X}_{N+1}^{*}$ determines the best direction of descent from point $\mathbf{X}_N^{(m)}$. In this direction is performed descent, as long as till is satisfied the condition:

$$F\left\{\mathbf{X}_N^{(m)} + la\Xi^*\right\} < F\left\{\mathbf{X}_N^{(m)} + (l-1)a\Xi^*\right\}. \qquad (12)$$

In case it is not performed the condition in point $\mathbf{X}_N^{(m)} + (l-1)\cdot a\Xi^*$ again we build the hyper-sphere of radius and again we select on surface of hyper-sphere m points for retrieving the best direction. If it turns out that all these test steps satisfy to condition

$$F\left(\mathbf{X}_N^{(m)} + a\Xi_i\right) > F\left(\mathbf{X}_N^{(m)}\right), \quad (i-1,2,....,m; \quad N = 1,2,....,q), \qquad (13)$$

Then the size of working step a is halved and is repeated selection of best direction. The process of crushing of the value of working step is continued till fulfill the conditions

$$a \cdot 2^{-s} < \varepsilon, \quad (s = 0,1,2,....), \qquad (14)$$

where s – a predetermined small number. Search terminates when the condition (14).

As can be seen from the algorithm, learning is carried out only at the first stage and at some intermediate stages of the search when turning of trajectory of search. The algorithm works in condition of "punishment". Distinguished by the excessive straightforwardness, for retrieving extreme algorithm with high precision requires a significant amount of testing.

The considered algorithm of random search to some extent uses training to select the next step. The learning and the search in this case are of a statistical nature. This is reflected in the fact that out of precisely defined number of tests at each step we choose only one, which determines the best direction. The relationship between the states of the system is very strong. The good one direction that has been found on previous step determines the subsequent search and the next step is not changed. Restructuring of the search system in a new direction takes place only if the objective function is not reduced or on the path of search system appears the restriction. The search system is too straightforward and is not optimal from the point of view of search costs. Probabilistic linkage between different states is absent. Learning takes place through trial and error in finding the most promising directions. Obviously, such direction can be found in other way by adjustment of probabilistic search characteristics and by introducing a self-learning of search system. Essential for such training is not only an increase or decrease in the objective function, but also an increase or decrease of each independent variable in the selected direction. Consider one of these algorithms.

2.2. The algorithm of random search method with the self-learning

This algorithm [3] is a modification of the algorithm self-learning with forgetting. It got its name due to the fact that at every stage of the search repeats the same standard set of operations.

The scheme of algorithm is as follows: from a given point $\mathbf{X}_0$ is performed a series of test steps, is determined the point $\mathbf{X}^*$ at which the objective function reaches a minimum value at this search stage, and then from that point again is performed a series of test steps till the next point with an even lower value of the objective function, etc.

International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (ISSN 2250-2459, ISO 9001:2008 Certified Journal, Volume 6, Issue 9, September 2016)

The recurrence relation, with accordance to which are performed the work steps for each stage of search has the following form:

$$\mathbf{X}_{N+1} = \mathbf{X}_N + a_N \left[\beta \frac{\mathbf{U}_N}{\|\mathbf{U}_N\|} + (1-\beta)\boldsymbol{\Xi}_N \right], \quad (15)$$

where: a_N – the value of step , a scalar, which increases after a successful step and decreases after a bad step; $\mathbf{U}_N$ – vector of "prehistory", that indicate the average direction of the search on the previous steps:

$$\mathbf{U}_{N+1} = \gamma \cdot \mathbf{U}_N + (1-\gamma)(\mathbf{X}_{N+1} - \mathbf{X}_N)\mathbf{S}_N . \quad (16)$$

Here: $\boldsymbol{\Xi}_N$ – unit vector of normal deviations that is realized by the generator of random number ; β – the coefficient, that changing during the search; γ – the coefficient of forgetting ($0 \le \gamma \le 1$); $\mathbf{S}_i$ – vector of scale multipliers for scaling of space $\mathbf{X}$.

On N-th step at determining a random vector $\mathbf{X}_{N+1}$ and the vector of prehistory $\mathbf{U}_N$ are averaged, as is evident from (16). Step $\mathbf{X}_{N+1}$ is considered successful if in point $\mathbf{X}_{N+1}$ is performed the inequality $F(\mathbf{X}_{N+1}) < F(\mathbf{X}_N)$. If the step was successful, the length of step a_N increases. In the opposite case, the stride length is decreased.

This algorithm is not so rectilinear as a local algorithm on the best test with pull-down. The choice of a new direction at each step of this algorithm is carried out taking into account the experience gained in a form of vector of prehistory. Moreover, in order to avoid of superfluous determinism of search algorithm is introduced the forgetting of gained experience that could degrade the quality of its search. When choosing a new direction the algorithm uses a probabilistic relationship between the previous and the next state of the system and is classified as self-learning algorithm.

III. The Global Random Search Method

The above algorithms of "wandering" random search solve the problem of the definition of a local extremum of the objective function. Meanwhile, there are problems that can be attributed to a class of multi-extreme tasks. Among these tasks is, for example, the task with concave restrictions that are imposed on the area of permissible parameters of the search, or the multi-connection task that requires a global extremum search in the area with ruptures, for example, the resonance zone, etc.

In this case, the problem can be solved by global search, which is many times more difficult then the process of determining of local extreme. That is why almost all global search methods are statistical in nature.

Most of the existing methods of global search can be divided into two classes: the class of "independent" and the class of "wandering" search methods.

The search techniques relating to the first class, at each stage use independent sample random values, in accordance with a given distribution. By appropriate selection of the density of distribution the trials are concentrated in the area, "suspect" to the existence of a global extremum. In the process of search the system gains an experience based on which occurs the transition from one distribution to the other.

The second class includes methods that mimic the wandering in area of optimized parameters in order to collect information about the object of optimization and to finding the global extremum.

Consider some of the global search algorithms belonging to both classes.

3.1. "Independent" Global Search

3.1.1. The random sorting [1].

One of the easiest algorithm of "independent" global search is a random sorting. This search algorithm is as follows:

$$\mathbf{X}_i^0 = \begin{cases} \mathbf{X}_{i-1}^0, & \text{if} \quad F(\mathbf{X}_i) \ge F_{i-1}^0 \\ \mathbf{X}_i, & \text{if} \quad F(\mathbf{X}_i) < F_{i-1}^0 \end{cases}, \quad (17)$$

$$F_i^0 = \begin{cases} F_{i-1}^0, & \text{if} \quad F(\mathbf{X}_i) \ge F_{i-1}^0 \\ F(\mathbf{X}_i), & \text{if} \quad F(\mathbf{X}_i) < F_{i-1}^0 \end{cases}, \quad (18)$$

Where $\mathbf{X}_i$ - the *i*-th random state, selected in accordance with a predetermined distribution density $p(\mathbf{X})$; $\mathbf{X}_i, F_i^0 = F(\mathbf{X}_i^0)$ –; the value of the objective function on the *i*-th step of the search.

In accordance with this algorithm at each step of randomly determined value of the objective function calculated in this state $\mathbf{X}_i$ and the obtained value is compared with a stored in memory $F(\mathbf{X}_i)$. If $F(\mathbf{X}_i) \ge F_{i-1}^0$ is performed the next random experiment, and if $F(\mathbf{X}_i) < F_{i-1}^0$, then memorized the new value of the index of quality and the state $\mathbf{X}_i$, which provides the decrease of the objective function.

International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (ISSN 2250-2459, ISO 9001:2008 Certified Journal, Volume 6, Issue 9, September 2016)

After this the next random sample is performed in accordance with given density in the area of parameters.

At the constant density of distribution of trials $p(\mathbf{X})$ this algorithm guarantees the finding of global extremum when $N \to \infty$. The disadvantage of such an algorithm is the large losses on the search. Therefore, under the solving real-world problems such global search method has quite limited application. Obviously, the density distribution of trials at different stages of research is changing in such a way to increase the likelihood of accidental finding a state close to a global extremum. Thus search efficiency is greatly increased.

Consider a general statement of the problem and some algorithms independent global research with controlled density distribution within the permissible area.

Suppose that in a compact, connected domain D n-dimensional Euclidean space E_n exists the bounded function $F(\mathbf{X}) = F(x_1,....,x_n)$, which has in D finite number of extremum values, ravines and saddle points. Thus $D = \{\mathbf{X} \in E_n : g_i(\mathbf{X}) \le 0, i = 1,2,...,q\}$ where $g_s(\mathbf{X})$ – a continuous function. Suppose also that $F(\mathbf{X}^*) = \min\limits_{X \in D} F(\mathbf{X})$ is the minimum point of $f(\mathbf{X})$ in D, and the function $f(\mathbf{X})$ is such that for all $C > f(\mathbf{X}^*)$ the set $W(C) = \{\mathbf{X} \in D : f(\mathbf{X}) \le C\}$ non-empty. The challenge is to find a point of $\mathbf{X}^*$, for which $\|\mathbf{X}^{**} - \mathbf{X}^*\| < \varepsilon$.

In the basis of the following algorithms laid modification of the algorithm of random search with the return after an unsuccessful step [4] of the following type:

$$\mathbf{X}_{N+1} = \begin{cases} \overline{\mathbf{X}}_{N+1}, & \text{if} \quad f(\overline{\mathbf{X}}_{N+1}) < f(\mathbf{X}_N) \\ \mathbf{X}_N, & \text{if} \quad f(\overline{\mathbf{X}}_{N+1}) \ge f(\mathbf{X}_N) \end{cases}, \quad (19)$$

$$f(\mathbf{X}) = F(\mathbf{X}) + \lambda \cdot \max\{0, g_i(\mathbf{X})\}, \quad (20)$$

where λ – the coefficient of fine for violation of restrictions; $\overline{\mathbf{X}}_{N+1} = \mathbf{X}_N + \mathbf{\Xi}_N$, $\mathbf{\Xi}_N \in E_n$, $\mathbf{\Xi}_N = (\xi_1, \xi_2, ..., \xi_n)$ – random vector with distribution density $p_N(\mathbf{\Xi})$.

At each step of the vector components of the search $\mathbf{\Xi}$ are not correlated, and the parameters of function $p_N(\mathbf{\Xi})$ attunes based on the results of the previous steps in such a way that for all $\mathbf{X}_N$ who have $f(\mathbf{X}_N) \ge f(\mathbf{X}^*) + r\varepsilon_1$ and $r > 0$

$$P_F(\mathbf{X}_N) = \int\limits_{W\{F(\mathbf{X}_N) - \varepsilon_1\}} p_N(\mathbf{\Xi}) d\mathbf{\Xi} \ge \varepsilon. \quad (21)$$

Here is used the notation, similar entries, that applied in [5]. It is assumed that for all $\mathbf{X} \in E_n$, for which $f(\mathbf{X}) - f(\mathbf{X}^*) < \varepsilon_1$ is fair $\|\mathbf{X} - \mathbf{X}^*\| < \varepsilon$, $\varepsilon_1 > 0$, $\varepsilon > 0$. Then, for a series of points $\{\mathbf{X}_N\}$, which are constructed by algorithm (17) with the distribution density satisfying the condition (19), at each step, there is a theorem that is presented below.

Theorem 3.1. Suppose that for any $C > f(\mathbf{X}^*)$ set $W(C)$ has interior points, and the density distribution of random tests $p_N(\mathbf{\Xi})$ satisfies the condition at each step (21). Then the sequence $\{\mathbf{X}_N\}$ which is produced by algorithm (19) contains in itself a sequence $\{\mathbf{X}_{N_K}\} \xrightarrow{\text{п.н.}} \mathbf{X}^*$ and $\{f(\mathbf{X}_{N_K})\} \xrightarrow{\text{п.н.}} f(\mathbf{X}^*)$.

Is proved [27] that at this $\{\mathbf{X}_N\}$ on the probabilities coincides with the optimum point. Using the theorem of Lebesgue-Rees, from the sequence $\{\mathbf{X}_N\}$, we select the sequence $\{\mathbf{X}_{N_K}\}$ that converges almost surely to $\mathbf{X}^*$.

At this the peculiarity of the construction of algorithms is stipulated by method of forming density, which satisfies the conditions of Theorem 3.1. Below adduced several of these algorithms.

3.1.2. Algorithm of "independent" global search with adaptation of the distribution of trials

Assume that the search area is the n dimentional parallelepiped (hyper-parallelepiped) [1]:

$$a_{i1}^{(1)} \le x_i \le a_{i2}^{(1)} \quad (i = 1,2,.....,n), \quad (22)$$

and the initial density distribution $p_i(\mathbf{X})$ is uniform throughout the volume of this parallelepiped $V_1 = \prod\limits_{i=1}^{n} (a_{i2}^{(1)} - a_{i1}^{(1)})$, i.e. $p_i(\mathbf{X}) = 1/V_1$ for $\mathbf{X} \in V_1$ and $p_1(\mathbf{X}) = 0$ for $\mathbf{X} \notin V_1$.

Let us divide the search process on k stages, each of which consists of N_j trials $(j = 1,2,...,k)$. At each stage we will determine the state that corresponds to the smallest value of the quality index.

International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (ISSN 2250-2459, ISO 9001:2008 Certified Journal, Volume 6, Issue 9, September 2016)

Suppose that after the first stage the minimum quality index value corresponds to a state $\mathbf{X}_{m_1} = \left(x_1^{(m_1)}, x_2^{(m_1)}, \ldots\ldots, x_n^{(m_1)}\right)$,

$$f\left(\mathbf{X}_{m_1}\right) = \min_{i=1,2,\ldots,N_1} \left\{f\left(\mathbf{X}_i\right)\right\}, \qquad (23)$$

In the second step the trials are performed in a similar parallelepiped with center in point $\mathbf{X}_{m_1}$, but each side is reduced in $c > 1$ in comparison with the original parallelepiped i.e.

$$a_{i1}^{(2)} \le x_i \le a_{i2}^{(2)} \quad (i = 1,2,\ldots\ldots,n), \qquad (24)$$

$$\text{where : } a_{i1}^{(2)} = x_i^{(m_1)} - \frac{a_{i1}^{(1)} - a_{i1}^{(1)}}{2c},$$

$$a_{i1}^{(2)} = x_i^{(m_1)} + \frac{a_{i1}^{(1)} - a_{i1}^{(1)}}{2c}.$$

In this case the trials $\mathbf{X}$ should satisfy the inequalities of both stages (22) and (24), i.e. the sample density distribution in the second stage is of the form:

$$p_2(\mathbf{X}) = \frac{c^n}{V_1}. \qquad (25)$$

As can be seen, the density has increased at least in c^n times. At the second stage we distribute in parallelepiped the N_2 trials, we determine the state $\mathbf{X}_{m_2}$ that corresponds to the smallest value of the quality index, and then we pass to the next stage.

At the *j*-th stage of the search the density of distribution the trials equals:

$$p_j(\mathbf{X}) = \frac{c^{jn}}{V_1}, \qquad (26)$$

and the search area is defined by inequalities

$$a_{i1}^{(k)} \le x_i \le a_{i2}^{(k)}; \quad i = 1,2,\ldots\ldots,n; \quad k = 1,2,\ldots\ldots,j, \qquad (27)$$

Where

$$a_{i1,i2}^{(k)} = x_i^{(m_{k-1})} \pm \frac{a_{i1}^{(k-1)} - a_{i1}^{(k-1)}}{2c}.$$

Here: $x_i^{(m_{k-1})}$ – the coordinate of best point on $(k-1)$-th stage.

This algorithm at every stage of the research clarifies the result obtained in the previous step. By the narrowing the search area (24), we are increasing the distribution density of trials and, therefore, increase the likelihood of finding of global extremum. However, at this increases the likelihood of "loss" of the global extremum on each stage, which in turn depends on the number of trials on stage. If in order to reduce this probability to significantly increase the number of tests, we would must confine ourselves by one step, that is, the search would to degenerate into a random sorting. Thus, the above algorithm does not guarantee of finding of global extremum, but can find a state when Quality Score is sufficiently close to the value of the global extremum.

3.1.3. The algorithm with a controlled variance of trials distribution

In some cases, the distribution of trials is expedient take not uniform, but the normal with the center in best point of previous step [1]. In this case, a search algorithm is formulated as such:

The random trials are defined with the help of n-dimentional normal distribution:

$$p(\mathbf{X}) = \frac{1}{(2\pi)^{n/2}\sigma_i^n} \exp\left\{-\frac{1}{2\sigma_i^2}\left|\mathbf{X} - \mathbf{X}_i^0\right|^2\right\}. \qquad (28)$$

In the algorithm memory on each step are stored X_i^o, F_{i-1}^0. The recurrence formulas for X_i^o and F_{i-1}^0 store the form (17) and (18), but σ_i, the middle square deviation, is changed in accordance with rule:

$$\sigma_i = \begin{cases} \sigma_0, & \text{if} \quad f(\mathbf{X}_i) < f_{i-1}^0 \\ \sigma_{i-1} - f(\sigma_{i-1}), & \text{if} \quad f(\mathbf{X}_i) \ge f_{i-1}^0 \end{cases}, \qquad (29)$$

where σ_0^2 – initial dispersion; $f(\sigma)$ – a function which determines a decrease σ_i (for example, $f(\sigma_i) = q\sigma_i$, where $0 < q < 1$).

The algorithm works in this way. Random trials are carried out in accordance with the normal distribution (28), the mathematical expectation of which correspond to the best trial for all previous trials and the variance is reduced at failed tests and is increases at successful tests. Reducing the variance in this case related to the need to specify the found local extreme, and increase (with the need to study the new situation for retrieving best point.

International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (ISSN 2250-2459, ISO 9001:2008 Certified Journal, Volume 6, Issue 9, September 2016)

This algorithm can also find a global extremum per a finite number of steps. However, it is more "cautious" algorithm, and due to the fact that the normal law has the endless "tails" it does not rule out in principle retrieving the global extremum and when $N \to \infty$ and appropriate choice of function $f(\sigma)$. For example, when $f(\sigma) = 0$ the considered search degenerates into a random sorting with a normal distribution of the tests, the convergence of which to the global extremum is intuitively obvious and proved in [5].

Global search methods discussed above have asymptotic character, that is, it is assumed that (at least in principle) is possible to do infinitely many tests, it is possible to be considered $N \to \infty$. However, in practical calculations, when each test, i.e. each definition of quality indicators $f(X)$ requires expenditures, the number of experiments N is always limited $N \le N_{\max}$. So naturally when determining the search strategy is necessary to take into account the value $N_{\max}$, that substantially affect the process of searching. Thus, is necessary to get the greatest effect per $N \le N_{\max}$ trial steps. This statistical approach to the process of receipt of global extremum requires of some additional information about the object of optimization.

3.2. The "wandering" global random search method

3.2.1. The theoretical grounding of wandering" global random search method

The "wandering" random global search method is a statistical development of regular gradient method [6, 7, 8]. In order to impart for search the global character on the gradient moving of system imposed random influences, creating a regime of random walk [1].

In minimizing the quality function $f(\mathbf{X})$ according to (1) in the case of a continuous gradient method reduces to the movement of point $\mathbf{X}$ in n-dimensional space of optimization settings under "force" aimed in the direction opposite to gradient of Quality Score. The movement of the point i $\mathbf{X}(t)$ in this case is described by the equation:

$$\frac{d\mathbf{X}}{dt} = -a\,\mathrm{grad}\,F(\mathbf{X}). \qquad (31)$$

The trajectory of movement $\mathbf{X}(t)$ is caused by this equation leads to the extreme point of the system (in the open zone or on the edge of the feasible region $g_j(\mathbf{X}) > 0, \quad j = 11,2,...m$).

If the starting point $\mathbf{X}^{(0)}$ is in the area of attraction of the global extremum, then the corresponding trajectory of the equation (31) would lead the system to a global minimum of the function $f(\mathbf{X})$. If the point $\mathbf{X}^{(0)}$ does not belong to the zone of attraction of the global extremum, then the motion in the direction of the greatest decrease $f(\mathbf{X})$ will result in a system that features a local minimum from which it is impossible to get under the influence of forces directed in the direction opposite to the gradient function $F(\mathbf{X})$. In such cases it is useful to include in the search system a random mechanism. Random bumps can help the point $\mathbf{X}(t)$ to overcome the barrier that separates the local minimum, which hit the system, from the area in which may still decrease. Such motion of a point under the influence of deterministic drift towards anti gradient and random impacts is determined by the following differential equation:

$$\frac{d\mathbf{X}}{dt} = -a\,\mathrm{grad}\,F(\mathbf{X}) + \Xi(t), \qquad (32)$$

where $\Xi(t)$ – n-dimensional normal random process with zero expectation, delta similar autocorrelation function (normal white noise), and spectral density σ^2.

The process described by this equation is a random Markov's process of diffusion type. The probability density of transition $p(\mathbf{X},t)$ from the initial state $\mathbf{X}^{(0)}$ to a point $\mathbf{X}$ in time t satisfies the Kolmogorov equation − Focker □ Planck [9]

$$\frac{\partial p(\mathbf{X},t)}{\partial t} - a\sum^{n}\frac{\partial}{\partial x_i}\left(p(\mathbf{X},t)\frac{\partial F}{\partial x_i}\right) = \frac{\sigma^2}{2}\sum_{i=1}^{n}\frac{\partial^2 p(\mathbf{X},t)}{\partial^2 x_i} \qquad (33)$$

with a natural condition $\int_{\Omega} p(\mathbf{X},t)d\mathbf{X} = 1$, where □ F – the domain of the quality function.

Equation (33) has the stationary solution [36]:

$$p(\mathbf{X}) = \frac{1}{c}\exp\left\{-\frac{2aF(\mathbf{X})}{\sigma^2}\right\}, \qquad (34)$$

where

$$c = \int_{\Omega}\exp\left\{-\frac{2aF(\mathbf{X})}{\sigma^2}\right\}d\mathbf{X}.$$

International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (ISSN 2250-2459, ISO 9001:2008 Certified Journal, Volume 6, Issue 9, September 2016)

This can easily be verified by direct substitution (34) into the equation (33).

The solution of this equation for any initial probability density $p(\mathbf{X},0)$ tends at $t \to \infty$ to the stationary solution (34).

From equation (34) obvious that the maximum value $p(\mathbf{X})$ corresponds to the point of global extremum of function $f(\mathbf{X})$. Thus, the most probably that the point $\mathbf{X}$ for quite a long time of search process is the position of a global extremum, and, as can be seen from the equation (34), the density $p(\mathbf{X})$ is determined by the equation:

$$\lim_{\sigma \to 0} p(\mathbf{X}) = \delta(\mathbf{X} - \mathbf{X}^*), \quad (35)$$

where δ – the generalized delta function, $\mathbf{X}^*$ – the position of global minimum of function $f(\mathbf{X})$. This means that with a decrease of variance of distribution of in the random trials probability of finding the system in an arbitrarily small neighborhood of a global minimum $\mathbf{X}^*$ tends to unity.

The reaction on the unsuccessful result obtained at random step may be bilateral. On the one hand, it is an immediate reaction to the elimination of failure is to return, recalculation or extrapolation. On the other hand, with the help of the mechanism of self learning eliminates the possibility of such failures in the future, that is appropriately rearranges the probability characteristics of a random step. If the step was successful, then at the relevant law of self learning occurs the registration of this experience in the form of certain adjustment of properties of the probability of random step. Thus, the part of the result of the optimization is fall on one and the other part of this effect - on another type of reaction. Exclusion of any of these reactions do not deliver search engine optimization abilities.

Consider the global search obtained by eliminating the immediate reaction to a unsuccess step [1]. Let us assume, that the direction of test random steps in the area of parameters of optimization is determined by the given multi-dimentional distribution $p(\Xi / \mathbf{U})$ depends on some n − dimentional unit vector $\mathbf{U} = (u_1, u_2, ..., u_n)$ as a parameter.

Distribution $p(\Xi / \mathbf{U})$ must have such property in order the direction of the mathematical expectation Ξ over all possible implementations conformed to the direction of the vector $\mathbf{U}$. Thus, $\mathbf{U}$ defines the average direction of the search. On the other hand, it must depend on the direction of the search history, that is, to be the best in terms of the previous work. So is naturally to call vector the vector of experience.

As you can see, this vector is very similar to the vector memory in the process of continuous self learning. The only difference is that in this case the vector of previous experience indicates only the direction but its magnitude does not carry information, as opposed to a vector memory, the module of which determines dispersion property of random test.

The search process we divide into stages. During the first phase (analysis) from the starting point, which determines the state of optimization in the parameter space of the system in - i -th moment and is executed m independent random tests in accordance with the existing density. The length of the test steps is constant and equal. In addition, each time determined the quality function

$$f_j = f(\mathbf{X}_i + h\Xi_j) \quad (j = 1,2,...,m). \quad (36)$$

At the second stage of the search (the decision) is determined by the direction of the working step $\Delta\mathbf{X}_{i+1}$, which depends on the results of the analysis made in the first phase and from the decision rule F, which connects the previous experience and the received information:

$$\Delta\mathbf{X}_{i+1} = a\mathbf{L}(\Xi_1, ..., \Xi_m;\ F_1, ..., F_m, \mathbf{U}_i), \quad (37)$$

Where $\mathbf{L}$ – a vector unit function; a – the length of the working steps in the parameter space. The function $\mathbf{L}$ is thus in a sense determines the best direction for next step. Therefore, a further search is necessary to execute in this direction. Based on the fact that the search direction is determined by the vector at the third stage (training) will naturally to change direction $\mathbf{U}$ according to the new results, for example, as follows:

$$\mathbf{U}_{i+1} = \frac{\Delta\mathbf{X}_{i+1}}{a}, \quad (38)$$

That is a new vector direction $\mathbf{U}_{i+1}$ determined by the direction of made by the working step $\Delta\mathbf{X}_{i+1}$.

This algorithm has a global character. Indeed, the random tests $\Xi_1, ..., \Xi_m$ are carried out is not anywhere, but only in a particular sector, the best direction determined by the vector, that is a new vector direction determined by the direction of made by the working step

This algorithm has a global character. Indeed, the random tests are carried out is not anywhere, but only in a particular sector, the best direction determined by the vector $\mathbf{U}$. This vector, or more precisely, the distribution $p(\Xi / \mathbf{U}_i)$ restricts the distribution of random tests only in a particular sector of space of optimized parameters.

International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (ISSN 2250-2459, ISO 9001:2008 Certified Journal, Volume 6, Issue 9, September 2016)

The direction of working steps is chosen according to certain rule $\mathbf{L}$, based on the information thus obtained.

On the other hand, such a search has a certain "inertia", because the direction of the search $\mathbf{U}$ is not significantly changes per one step, at least on average. Availability of distribution $p(\Xi / \mathbf{U}_i)$, determines the direction of search steps, provides in some sense "smooth" motion of the system, which can be considered inertial. The system in the search process moves along the trajectory of moving of heavy point.

If there is a ravine quality function search the system of search will move along ravine regardless of whether he or falls rises.

As you can see, the considered algorithm stimulates movement of the of search optimization system in direction ravine quality function. This allows the system to overcome the "ridges" on the "passes" of the objective function and find new areas of its "global lowlands." Further analysis of these areas in order to find a local extremum can be performed by any other local search method. Set out a search algorithm not finds the global extremum, but picks out from the parameter space the areas, where there may be a global extremum of the objective function.

Now consider and analyze a few specific search algorithms.

3.2.2. Algorithm with directing sphere [1].

Suppose that the random points are selected on the surface of *n*-dimentional hyper-sphere, and this very hyper-sphere slightly pushed in the direction of the vector memory $\mathbf{U}$. Then the thus formed random directions tend shifts toward Ξ^0, and this tendency is more strongly expressed than on a large value the hyper-sphere is shifted along the vector of memory $\mathbf{U}$. This scheme forms the algorithm with the guide sphere. The direction of the test step in the algorithm is determined by the following formula:

$$\Xi = \frac{\mathbf{U} + c\Xi^0}{\left|\mathbf{U} + c\Xi^0\right|}, \qquad (39)$$

Where Ξ^0 – random unit vector, distributed with equal probability in all directions of the parameter space; c – a constant magnitude (radius of hyper-sphere). Fig.2 shows the interaction of vectors Ξ^0 and $\mathbf{U}$ in the process of formation Ξ. By a dotted line is indicated the hyper-sphere of possible realizations of a random vector $c\Xi^0$. As can be seen, at $c < 1$ for all test steps are inside hyper-cone with the axis $\mathbf{U}$ and angle of opening $2\arcsin c$.

The smaller the value, the more narrow is the cone and the closer one test to another. The direction of the working steps in this case it is possible to determine the best test

$$\mathbf{L} = \Xi^* \qquad (40)$$

or if the value of the quality function in the area of test steps, that is, in the area of the hyper-sphere radius c can be approximated with sufficient accuracy by the linear shape, the direction of working steps can be defined by the following formula:

$$\mathbf{L} = \frac{\sum_{j=1}^{m} \Delta F_j \Xi_j}{\left|\sum_{j=1}^{m} \Delta F_j \Xi_j\right|}, \qquad (41)$$

where Δf_j – an increase in the quality of function caused by the *j*-th random test Ξ_j.

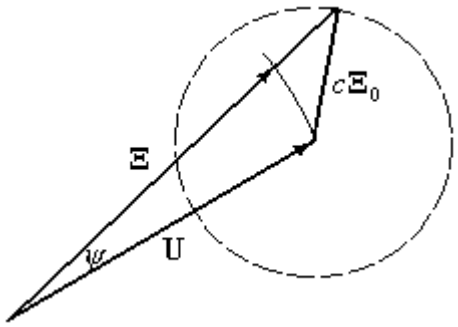

Fig. 2. To definition of the direction in algorithm with a guide sphere

This equation is obtained by "weighing" of test directions. By "weight direction" is to be understood an increase in the objective function resulting from the test step along this direction. The larger the modulus of this increase, the more "significant" is the direction. Thus, the direction of the working step is a weighted average of all the random tests. A modification of this algorithm is an algorithm with a guide cone.

3.2.3. Algorithm with guide cone [1]

Suppose that in a certain parameter space there is the hyper-cone with the apex $\mathbf{X}$ of which coincides with the direction of the axis of the vector memory $\mathbf{U}$, and the angle at the vertex is 2ψ. Around the top of the cone as around the center, we hold the hyper-sphere of radius c. The cone cuts off from this sphere the part of surface on which randomly we choose m independent points $\Xi_1, \ldots\ldots, \Xi_m$.

International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (ISSN 2250-2459, ISO 9001:2008 Certified Journal, Volume 6, Issue 9, September 2016)

Calculating in them the objective function $Q(\mathbf{X}+g\Xi_j)$, we define the point corresponding to the minimum value of the penalty objective function in accordance with (20):

$$f(\mathbf{X}+c\Xi^*)=\min_{j=1,\ldots,m} f(\mathbf{X}+c\Xi_j). \qquad (42)$$

In this direction performed the step which we will call the "working step". The search direction is determined entirely by this cone as since the random tests are selected inside. Therefore it is natural to call this guide cone.

The direction of the vector memory in this case it is possible to determine by the best test (40), and by the weighted average direction (41).

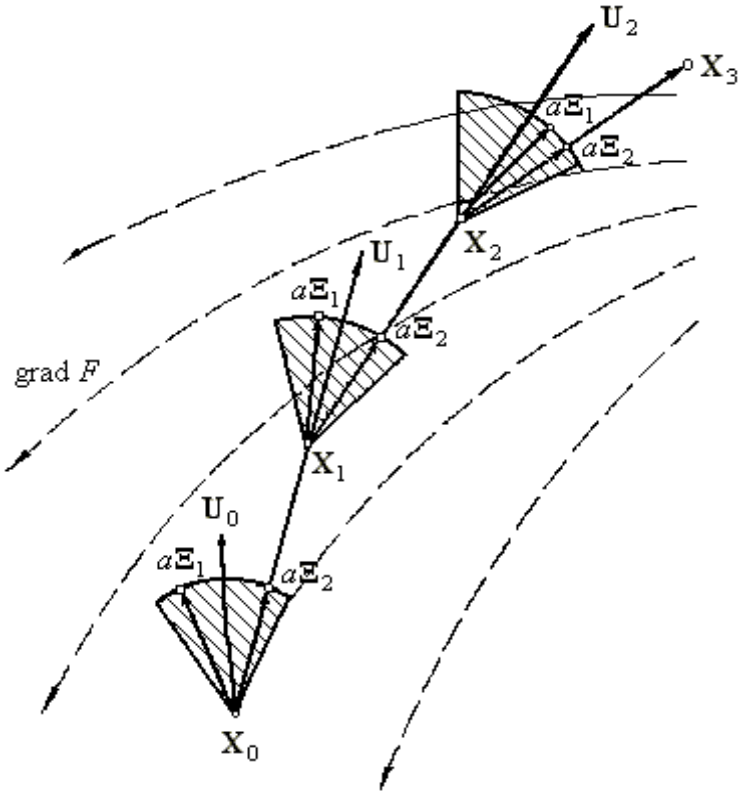

Fig.3. The algorithm with the guide cone. The motion of system in the space parameters of optimization

In Fig. 3 is shown a number of steps of search for $c<1<a$ and $n=m=2$ out of the state $\mathbf{X}_0$ with an arbitrary initial vector the direction of memory $\mathbf{U}$, which varies according to the best test in the search process. One can see that with the accumulation of information about the behavior of the objective function, this vector $\mathbf{U}$ turns in the direction opposite to the gradient. It is obvious that with a decrease in the opening angle of the cone of the vector $\mathbf{U}$ rotation opportunities in one working step are reduced, which is a consequence of the "inertness" of the search mentioned above. This means that at the sudden change in the direction of the gradient, the system will move in the same direction, and then gradually will rebuild the vector of memory $\mathbf{U}$ to a new state.

On the other hand, the search expenditures for such algorithm at the correct direction of vector of memory $\mathbf{U}$ (or in the direction opposite the gradient) decrease with decreasing taper angle.

As the increasing of angle the system becomes more mobile, but by increasing the expenditures on search .

Mobility of system and the expenditure on search are investigated in [1]. As a measure of mobility of search is the middle angle of turn of the vector of memory $\mathbf{U}$ per one step search or the average curvature of the trajectory in the parameter space. Due to the random nature of the search the angle turning of trajectory Ω (Fig.3) is also a random variable.

If to sum, we can say that this algorithm has the necessary and global property, but it is quite sensitive to the choice of search parameters − working step length a, the opening of angle ψ of the guide cone and the number of tests m. These options is necessary to choose during the search, i.e. the search must be with self-learning.

International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (ISSN 2250-2459, ISO 9001:2008 Certified Journal, Volume 6, Issue 9, September 2016)

This remark also applies to the algorithm of guide sphere, which is also three parameters □− the working length of the step a, the value of the parameter c and the number of tests m. These parameters are determined in the same way the inertness of the search. Indeed, the greater the step and the number of tests or less than the parameter, the more inertness has the system of search , and vice versa. Obviously, the optimum values of search parameters depend entirely on the type of the objective function $F(\mathbf{X})$ and its features. In problems where this function has a lot of extremums, due much difficulty, obviously, the only practical way of finding the optimal values of the search parameters is the self-learning of search.

3.2.4. The global method of random search with controlled boundaries of the interval of parameters (SGEF)[10]

The proposed algorithm is a statistical method and is classified as one of wandering algorithms [1]. However, unlike the methods of random search with a return after an unsuccessful step this algorithm always produces a step:

$$\mathbf{X}_{N+1} = \begin{cases} \overline{\mathbf{X}}_{N+1}, & if \quad Q(\overline{\mathbf{X}}_{N+1}) < Q(\mathbf{X}_N), \\ \mathbf{X}_N, & if \quad Q(\overline{\mathbf{X}}_{N+1}) \geq Q(\mathbf{X}_N), \end{cases} \tag{43}$$

$$\text{where: } Q(\mathbf{X}) = f(\mathbf{X})Q(\mathbf{X}) + \lambda \cdot \max\{0, \quad g_s(\mathbf{X})\}, (s = 1,2,...,q) \tag{44}$$

Here: λ − penalty coefficient for violation of the restrictions; g_s (**X**) – □ functional and geometric constraints (s = 1,2, ..., q).

Let x_i^-, x_i^+ – □ respectively the lower and upper limits of the control variables. Then geometric constraints take the form of:

$$G_s(\mathbf{X}) = (x_i - x_i^- \geq 0;\ x_i^+ - x_i \geq 0),\ (i = 1,2,....,n;\ s = 1,2,...,q) \tag{45}$$

$\mathbf{X}_{N+1} = \mathbf{X}_N + \Delta\mathbf{X}_N$; $\Delta\mathbf{X}_N \in E_n$; $\Delta\mathbf{X}_N = \{\Delta x_1, \Delta x_2, ..., \Delta x_m\}$ – vector increments of coordinates, correlated the results of previous steps. The process of correlation of increments of parameters to be optimized x_i is executed through the management by the boundaries the intervals of changing of variables x_i.

This method was used in the numerical experiments for confirmation of evolutionary theory of identification of mathematical models of corrosion destruction, the main results of which are published in [11-16]. Therefore, this algorithm will be considered in more details.

Algorithm. Without loss of generality, we assume y_i^- = 0 ; y_i^+ = 1, (i = 1,2,...,n). The density distribution of the random vector $\Delta\mathbf{Y}_N$ is built by the each coordinative reflection of the n - dimensional uniform law in hypercube with a side of [-1, 1]. The boundaries of this hypercube are reflected accordingly on the hyper-parallelepiped $\Delta\mathbf{X}_N$. From any point $\mathbf{X}_N$ for each coordinate x_i is performed step:

where: □ $\mathbf{X}_N(x_1, x_2, ..., x_n)$ – n- dimensional vector of parameters to be optimized on the N - th search step; α_i,β_i − respectively the lower and upper boundaries of i – th random variable ξ_i, uniformly distributed on the interval [α_i, β_i]; Ξ_i – a random variable uniformly distributed on the interval [0,1]; y_i – normalized value of i- th parameter in the previous step; y_i^{N+1} , x_i^{N+1} – respectively normalized and non-normalized value of the i -th parameter in the $N+1$ search step; □ c – a step size ($0 \leq c \leq 1$). After a successful step the value of step increases after unsuccessful □ reduced.

After each procedure, check the implementation of restrictions and if the restrictions are met, the next step is carried out from the point $\mathbf{X}_N$ according to the procedure (46)−(51). A similar procedure is performed m times. The thus obtained vectors $\mathbf{X}_{N+1}$ are stored, and the objective function (44) is calculated for each of them. From the thus found values of the function Q is selected best Q_{min} and appropriate vector $\mathbf{X}_{min}$ is remembered. Such a step is considered as working, and the direction along which it is made, is the best. Further search strategy will be such, in order that the next steps were performed in a direction close to best. It should be noted that in any case, the new point is found, even if there is a value of the objective function worse than in the previous point.

$$\xi_i = (\beta_i - \alpha_i)\,\Xi_i + \alpha_i,\ (i = 1,2,...,n); \tag{46}$$

$$y_i^N = (x_i^N - x_i^-) / (x_i^+ - x_i^-); \tag{47}$$

$$\Delta y_i^{N+1} = c\,\xi_i; \tag{48}$$

$$y^{N+1} = y_i^N + \Delta y_i^{N+1}; \tag{49}$$

$$\Delta x_i^{N+1} = \Delta y_i^{N+1}(x_i^+ - x_i^-); \tag{50}$$

$$x_i^{N+1} = x_i^N + \Delta x_i^{N+1}, \tag{51}$$

International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (ISSN 2250-2459, ISO 9001:2008 Certified Journal, Volume 6, Issue 9, September 2016)

The latter fact attach a global sense to algorithm of search, significantly reduces the impact of before-history of search, provides the flexibility to find direction in a difficult multi-extreme area, in case of the discontinuous objective function , in the presence of saddle points.

The mechanism of tracking successful search direction consists in the management change i - th variable in accordance with the law:

$$\alpha_i^{N+1} = \|\xi_i\| - (\beta_i - \alpha_i)/2;\ \beta_i^{N+1} = \|\xi_i\| + (\beta_i - \alpha_i)/2, \quad (52)$$

where $\|\xi_i\| = \xi_i / \Sigma\xi_i^2$ – □ norm of ξ; $\alpha_i \geq -1.0$; $\beta_i \leq 1.0$.

As follows from expression (52), the boundaries of the variables α_i and β_i draw together with each successful step interval $[\alpha_i, \beta_i]$ decreases. With the drawing together the boundaries of the variable x_i the influence of the second term in expression (52) becomes insignificant and the value of the variable is determined by the boundaries of the vector norm ξ. Thus, moving in the selected direction is stabilized. At the violation of any of restrictions is provided the crushing of step:

$$c^{N+1} = c^N / 2 \geq \varepsilon, \quad (11)$$

where ε – □ the set small value.

In the case, if not crushing of step do not leads to further displacement of the search system, the boundary variables α_i, β_i restored till the initial values α_0, β_0 and retrieval system makes a U-turn. If such a situation arises near the restriction, the search system moves away from restriction and finds a new way to improve the objective function.

To improve the convergence of the algorithm search is carried out from different starting points. The starting point can not be selected in advance, but formed by the search. As a starting point adopted the so-called "point that is the suspicious on the extreme value". It is the point at which the objective function as a result of the last step "impairs" own value.

Typically, these points lie on the boundary of permissible values of variables. The number of points is due in advance. According to the exhaustion of "point that is the suspicious on the extreme value" search stops. The algorithm operates in the global and local search mode. The search engine provides two modes of global research: basic and additional.

In the course of the main global search system performs the optimization of objective function, explores the area of permitted parameters and determines the "point of suspicious on extreme value ". In the course of additional search the system determines the extreme of objective function and refines the solution found in the local search mode. Then there is a return to the mode of the main global search with the restoration of all the parameters that the system was possessed before the additional search mode.

In further, the search is in native mode from the last starting point at previous basic mode, defined a new "point of suspicious on the extreme value of objective function", defines a new extremum, that is refined by the local search mode, etc. The best result is stored as a global extremum. Search stops on the reaching the limit on the number of starting points, which is due in advance.

The algorithm works in reverse mode and is capable find a maximal and minimal values of objective function. The initial starting point may be taken in the area of permitted parameters and outside of this area. In the latter case, the system of search includes the mode of fines at violation of restrictions. This mode is switched off as soon as the system of search return in the area of permitted parameters. Subsequently, the system of search can not go out from the permissible area.

To illustrate the proposed algorithm is considered the optimization of the objective function (Table 1) of special structure set out in [17].

International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (ISSN 2250-2459, ISO 9001:2008 Certified Journal, Volume 6, Issue 9, September 2016)

Table 1
The objective function

$f(\mathbf{X}) = B_1 + B_2x_1 + B_3x_1^2 + B_4x_1^3 + B_5x_1^4 + B_6x_2 + B_7x_1x_2 + B_8x_1^2x_2 + B_9x_1^3x_2 +$ $B_{10}x_1^4x_2 + B_{11}x_2^2 + Bx_2^3 + B_{13}x_2^4 + B_{14}\frac{1}{x_2+1} + B_{15}x_1^2x_2^2 + B_{16}x_1^3x_2^2 + B_{17}x_1^3x_2^3 +$ $+ B_{18}x_1x_2^2 + B_{19}x_1x_2^3 + B_{20}\{\exp[0{,}0005x_1x_2]\}$	
$B_1 = 75{,}1963666677$	$B_{11} = 0{,}2564581253$
$B_2 = -3{,}8112755343$	$B_{12} = -0{,}0034604030$
$B_3 = 0{,}1269366345$	$B_{13} = 0{,}0000135139$
$B_4 = -0{,}0020567665$	$B_{14} = -28{,}1064434908$
$B_5 = 0{,}0000103450$	$B_{15} = -0{,}0000052375$
$B_6 = -6{,}8306567613$	$B_{16} = -0{,}0000000063$
$B_7 = 0{,}0302344793$	$B_{17} = 0{,}0000000007$
$B_8 = -0{,}0012813448$	$B_{18} = 0{,}0003405462$
$B_9 = 0{,}0000352559$	$B_{19} = -0{,}0000016638$
$B_{10} = -0{,}0000002266$	$B_{20} = -2{,}8673112392$

This function of two independent variables has single peak and a saddle point. A number of specific tasks, each of which is to maximize the objective function in the presence of a set of constraints-inequalities (Table 2) are considered. Three tasks such concretized by selecting a subset of constraints-inequalities are shown in Table 3.

The peak of the objective function is in the point with coordinates x_1 = 81,154841 and x_2 = 69,135588; at this point the objective function takes the value 61,9059345. Conditional maximum located at coordinates x_1 = 75,000000 and x_2 = 65,000000; at this point the objective function value is equal to 58,9034360. The search carried out from different starting points, lying in the area of permitted parameters, and outside this area. The trajectories of search are shown in Fig. 4 and Fig. 5.

Table 2
Constraints-inequalities, used for the create of two-dimensional models

$g_1(\mathbf{X}) = x_1 \geq 0$	$g_7(\mathbf{X}) = 65{,}0 - x_2 \geq 0$
$g_2(\mathbf{X}) = x_2 \geq 0$	$g_8(\mathbf{X}) = x_2 - 5{,}0\ \{x_1/25{,}0\}^2 \geq 0$
$g_3(\mathbf{X}) = 95{,}0 - x_1 \geq 0$	$g_9(\mathbf{X}) = (x_2 - 50{,}0)^2 - 5{,}0\ (x_1 - 55{,}0) \geq 0$
$g_4(\mathbf{X}) = 75{,}0 - x_2 \geq 0$	$g_{10}(\mathbf{X}) = x_1 - 54{,}0 \geq 0$
$g_5(\mathbf{X}) = x_{1*}x_2 - 700{,}0 \geq 0$	$g_{11}(\mathbf{X}) = 1{,}5\ (x_2 - 45{,}0) - (x_1 - 45{,}0) \geq 0$
$g_6(\mathbf{X}) = 75{,}0 - x_1 \geq 0$	$g_{12}(\mathbf{X}) = x_1 - 35{,}0 - 1{,}6\ (x_2 - 40{,}0) \geq 0$

Table 3
Variants for two-dimensional tasks of nonlinear programming with restrictions

Number of task	Number of Figur	Number of restrictions
1	1	1, 2, 3, 4, 5
2	2	1, 2, 3, 4, 5, 8, 9, 10
3	3	1, 2, 3, 4, 5, 6, 7, 8, 9, 10,11,12

Task №1. Find the maximum of the objective function given in Table 1, when the number of restrictions is 5 (Table 3). The search is carried out from different starting points lying in the permissible area.

Coordinates of the first initial starting point is $\mathbf{X}(0) = [90.0,\ \ 10.0]$.

International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (ISSN 2250-2459, ISO 9001:2008 Certified Journal, Volume 6, Issue 9, September 2016)

Search options accept the following: the number of attempts at the basic and additional global search $m = 3$; step size at basic global search takes the $h = 0{,}1$ side of the unit hypercube and at additional global search and at local $h = 0{,}05$ side of the unit hyper-cube; the number of "suspicious points on the extreme value of objective function" in the major global search $n_1 = 3$; the number of "suspicious points of the extreme value" at additional global $n_2 = 5$; number of attempts to refine the global extremum $n_3 = 3$, at the stage of local search $n_4 = 3$. As can be seen from Fig. 1 the system of search at each step selects a direction close to a positive gradient.

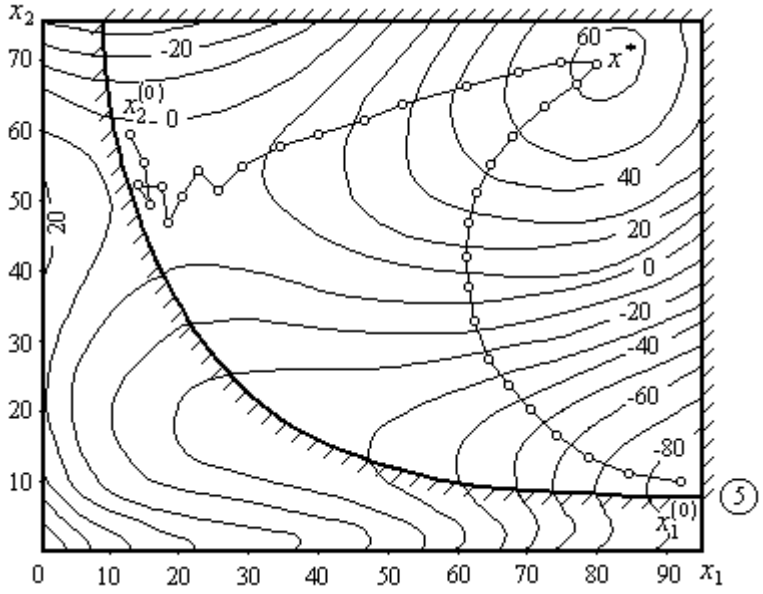

Fig. 4. The task 1. The trajectories of system of search

In the linear field, away from the extreme point, the search system is stabilized, the borders interval α_i,β_i, for each of the variables draw together and slowly change during the motion of search system along the statistical gradient in the region of extreme value of objective function. In the central field, or in "points that is the suspicious on extreme value of objective function " gradient changes and the system begins to unlearn. Interval α_i,β_i is increased, the density of attempts decreases and the search system unfolds. Extreme point of **X** = [81,163; 69,143] has been found for the 217 working steps at total loss on the search for 960 attempts. The value of the objective function in the extreme point was 61,906. Search system stopped and when $n_1 = 4$ и $h < \varepsilon = 2^{-7}$. The second initial starting point ($\mathbf{X}^{(0)}$ =[10,0; 60,0]) was chosen specifically in the area of the saddle point.

Number of intermediate points on the stage of the main global search was increased to n_1=10. The inertia of the system at the saddle area was small, the system for a long time wandering slowly moved to the border area of the saddle, and then got out of it and found a peak of 61,906 objective function with coordinates x_1 = 81,162 and x_2 = 69,141. Total losses amounted to 7504 attempts, the number of working steps amount 2033. The main losses occurred in the saddle area. The trajectories of search are shown in Fig.4.

Task №2. Find the maximum of the objective function (Table 1) when the number of restrictions 1−5 and number 8−10 (Table 3). The initial starting points, $\mathbf{X}(0) = [90.0,\ 10.0]$ are not in permissible area. The objective function takes the form (2).

International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (ISSN 2250-2459, ISO 9001:2008 Certified Journal, Volume 6, Issue 9, September 2016)

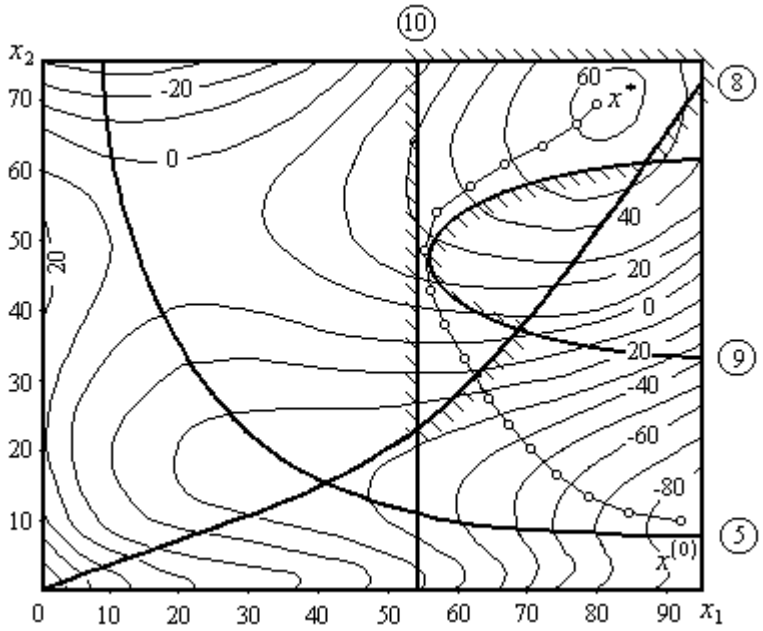

Fig 5. The task 2. The trajectories of system of search

The penalty coefficient was accepted $\lambda = 100$. Search options taken the same as in the problem №1. The system of search found the permissible area and, moving along the restriction №10, crossed a narrow isthmus that have been formed by restrictions number 9 and number 10, has moved further in the direction of the extremum and found the peak of the objective function $\mathbf{X}^*$ = [81,163; 69,143]. The value of the objective function at the extreme point was 61,906. Total losses amounted to 756 attempts, the number of working steps amounted 187. The trajectory of the search is shown in Fig. 5.

Task №3. Find conditional maximum of the objective function (Table 1) when all 12 restrictions are used (Table 3). The initial starting point, $\mathbf{X}(0) = [90.0,\ 10.0]$ was selected in unpermissible area.

Restrictions number 11 and №12 form a narrow ravine, the bottom of which is a permissible area. Constrained optimization lies at the intersection of restrictions № 6,7,11,12. Search options and the penalty coefficient taken the same as in Problem №2. The system of search entered in the permissible area and raised up along the bottom of the ravine to the conditional extremum. Total losses amounted to 1032 attempts, the number of working steps amounted 205. The value of objective function in an extreme point at the coordinates x_1 = 74,9998 and x_2 = 64,9998 amounted to 58,90325. The same solution has been found from a number of other randomly chosen initial starting points. The trajectories of search are shown in Fig. 6.

International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (ISSN 2250-2459, ISO 9001:2008 Certified Journal, Volume 6, Issue 9, September 2016)

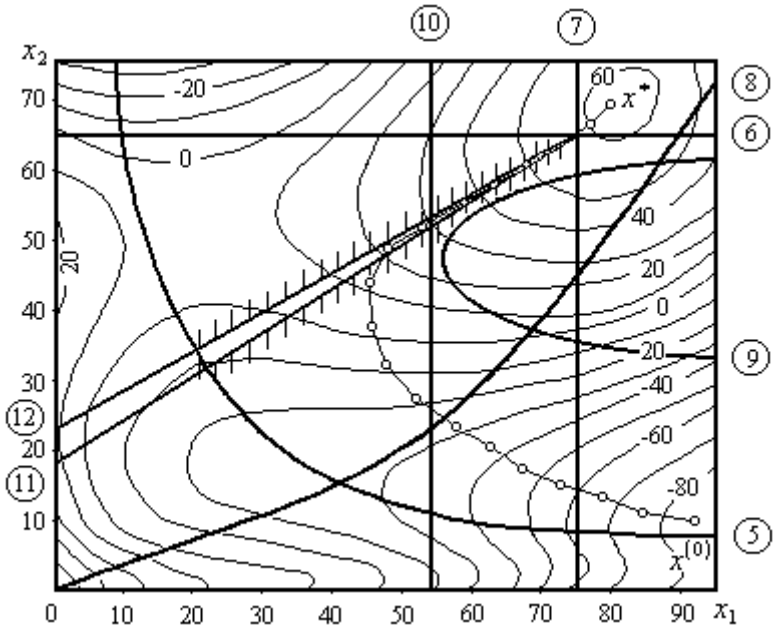

Fig 6. The task 3. The trajectories of system of search

IV. CONCLUSIONS

1. The management by the boundaries of controlled variables allows to stabilize the direction of the search in the linear field by increasing the distribution of attempts density in a narrow range specified by constantly changing boundaries of variables.
2. The global nature of algorithm is achieved by inertia of system of search and the introduction of a two-stage strategy of search. The inertia, which is the result of training, allows the system to search in difficult terrain conditions, typical for multi-extreme functions with saddle points and ravines. The search system is steadily moving along the gentle slope constraints, pushing off on them through intermediate starting points, in which the system of search is learning to make a choice of new directions. Two-stage system allows to search the area to identify the "points of suspicious on extreme value of objective function" and perform the search of global extremum from these points, by refining the previously found solutions.
3. The algorithm is reliable and easy in mathematically relation, requires little or no expenditures on the solution problem, accessible to a wide range of engineering and technical personnel who do not have special training, has an applied nature.

REFERENCES

[1] Rastrigin L.A. Stochastic search methods [Text] // L.A. Rastrigin / − M.: Nauka, 1968. − 376 p.

[2] Fiakko A., McCormick G. Nonlinear Programming. Methods consistent unconditional optimization.− M.:Mir, 1972. − 240 p.

[3] Kelly R. J,, Wheeling R. F., A Digital Computer Program for Optimizing Nonlinear Functions // Mobil Oil Corp., Research Dept., Central Research Div., Princeton, N. J., July 1962. – p. 627− 636.

[4] Rastrigin LA The systems of extremal managment .− M .: Nauka, 1974.− 630 p.

[5] Matyash J. Random optimization // Automation and telemehanika.−1965. №2. p.p. 246 −253.

[6] Hadley J. The non-linear and dynamic programing.− M .: Mir, 1967.− 506 p.

[7] Yudin D.B. The methods of quantitative analysis of complex systems // Izv. USSR Academy of Sciences, Technical Cybernetics. − № 1. − 1966.

[8] Yudin D.B., Hazen E.M. Some mathematical aspects of statistical methods to search // Collection "Automation and Computer Science". − Vyp.13. − Riga: Zinatne. − 1966.

[9] Kolmogorov. A.N. Probability theory and mathematical statistics.− M: Science, 1986 – 534p.

[10] Filatov G.V. The Stochastic Method of Search of Global Extreme Value of Function with the Guided Scopes of Interval of the Optimized Parameters [Text] // The Questions of Chemistry and Chemical Technology. Dnepropetrovsk: UGHTU. – 2000. p.p. 334−338.

[11] Filatov G.V. The Foundations of the Evolution Theory of Identification of Mathematical Models of Corrosion Destruction at the Optimal Planning of Constructions [Text] // G.V. Filatov / − International Journal of Emerging Technology & Advanced Engineering, Volume 6, Issue 3, March 2016, p.p.166-180.

[12] Filatov G.V. The Numerical Experimental Verification of Evolutional Theory of Identification of Mathematical Models of Corrosive Destruction under Stress. Compressed Shell [Text] // G.V. Filatov / − International Journal of Emerging Technology & Advanced Engineering, Volume 6, Issue 4, April 2016, p.p. 1-9.

[13] Filatov G.V. Application of Evolutional Theory of Identification of Mathematical Models of Corrosive Destruction at Optimum Designing of Weld-fabricated I-beam [Text] // G.V. Filatov / − International Journal of Emerging Technology & Advanced Engineering, Volume 6, Issue 5, May 2016, p.p.222-236.

[14] Filatov G.V. Optimal design of structures by the combined use of mathematical models of corrosion destruction [Text] // G.V. Filatov / − International Journal of Emerging Technology & Advanced Engineering, Volume 6, Issue 5, May 2016, p.p.6-15.

[15] Filatov G.V. The methods of Polynomial Approximation for the Determining the "Optimal" Coefficients of the infiuence of Stress-Strained State on the rate of Corrosion. Algorithm 1. [Text] // G.V. Filatov / − International Journal of Emerging Technology & Advanced Engineering, Volume 6, Issue 7, July, 2016, p.p.13-23.

[16] Filatov G.V. Application of the Method of Polynomial Approximation for the Determining the "Optimal" Coefficient of the Mathematical Model of Corrosive Destruction. Algorithm 2. [Text] // G.V. Filatov / − International Journal of Emerging Technology & Advanced Engineering, Volume 6, Issue 8, August, 2016, p.p.

[17] Himmelblau D. Applied Nonlinear Programming [Text] // David M. Himmelblau / − The University of Texas, Austin, Texas, McGraw-Hill Book Company, 1972. − 534 p.