GLOBAL OPTIMIZATION: FRACTAL APPROACH AND NON-REDUNDANT PARALLELISM

ROMAN G. STRONGIN AND YAROSLAV D. SERGEYEV

ABSTRACT. More and more optimization problems arising in practice can not be solved by traditional optimization techniques making strong suppositions about the problem (differentiability, convexity, etc.). This happens because very often in real-life problems both the objective function and constraints can be multiextremal, non-differentiable, partially defined, and hard to be evaluated. In this paper, a modern approach for solving such problems (called global optimization problems) is described. This approach combines the following innovative and powerful tools: fractal approach for reduction of the problem dimension, index scheme for treating constraints, non-redundant parallel computations for accelerating the search. Through the paper, rigorous theoretical results are illustrated by figures and numerical examples.

Everything should be a simple as possible, but not simpler.
Albert Einstein

CONSTRAINED OPTIMIZATION AS A DECISION MODEL

The world today exhibits booming innovations in the realm of technologies providing unbelievable diversity of products and services of high quality. There are two important sources for the best decisions leading to these impressive achievments.

It is very traditional that essential advancements in the modern practice of creating technical systems and technological processes of high efficiency are based on the employment of new principles, new materials, new physical effects and other *new discoveries*. And it is very often that these breakthroughs play the key role in the designation of the general structure of the objects to be designed.

Another typical source of significant improvements is in the *selection of the best combination* for the set of the object's parameters (with the general structure or the linkage being already set defined). Variations of these parameters (geometrical sizes, electrical and strength specifications, etc.) can substantially affect the performance characteristics of the object.

New discoveries (related to any particular area of design) are relatively rare events originated by some costly research activities. But the techniques for setting the best combinations for the values of the objects parameters can be

Roman G. Strongin is the Head of Software Department, University of Nizhni Novgorod, Russia (strongin@unn.ac.ru)

Yaroslav D. Sergeyev is the Senior Researcher, Institute of Systems Analysis and Information Technology, CNR, c/o DEIS, University of Calabria, Rende (CS), Italy and the Professor of the University of Nizhni Novgorod, Russia (yaro@si.deis.unical.it)

much more universal (not necessarily oriented towards some particularistic tasks) and noticeably less expensive. The information these techniques are calling into requisition is commonly obtainable as the result of observing some limited amount of *trial options*. Today it is typical that these trial options are not the real examples to be tested experimentally. Numerical (computer aided) analysis of the *mathematical model* simulating the performance of the real object is the modern tool for observing trial options.

The above model includes the set of n parameters

$$w = (w_1, \ldots, w_n) \in S$$

which are to be assigned with some particular values from the given *domain of search*

$$S = \{w \in R^n : \quad a_j \le w_j \le b_j, \quad 1 \le j \le n\} .$$

Performance characteristics of the object within the model are presented by the set of $m+1$ real functions $f_i(w), 1 \le i \le m+1$. We assume that these functions are selected in such a way that the diminution of their values corresponds to the better performance of the object.

One of the widely used concepts of the best decision linked with the above model classifies all the options $w \in S$ as either admissible or not admissible. The *admissible* options have to satisfy the *constraints* set by the inequalities

$$g_i(w) = f_i(w) - q_i \le 0, \quad 1 \le i \le m .$$

These requirements for the performance indexes $f_i, 1 \le i \le m$, to be below some preset levels $q_i, 1 \le i \le m$, usually reflect some necessary conditions for the object to actually function. Therefore, the admissibility of the choice $w \in S$ may be interpreted as the *feasibility* of the corresponding object.

Within the concept under consideration, one of the performance indexes is treated as the major *criterion* (we assume that the subscript $m+1$ is reserved for this purpose) and the *best solution* $w^* \in S$ has to minimize the function $\varphi(w) = f_{m+1}(w)$ over the set of all admissible options, i.e.

$$\varphi(w^*) = \min\{\varphi(w) : \ w \in S, \ g_i(w) \le 0, \ 1 \le i \le m\} .$$

Now the problem of the best decision may be interpreted as the mathematical problem of constrained optimization.

Any implementation of the decision w^* (if it does exist) could be carried out with some limited accuracy. Therefore, this decision may be of interest in applications only if it has an admissible vicinity in $S \subset R^n$ characterized by some positive *volume*. To meet this requirement we introduce the concept of an *ε-reserved solution* w_ε which (if it exists) wittingly has such a vicinity. This solution is defined by the relations.

$$\varphi(w_\varepsilon) = \min\{\varphi(w) : \quad w \in S, \quad g_i(w) \le -\varepsilon, \quad 1 \le i \le m\} ,$$

where $\varepsilon_R = (\varepsilon_1, \ldots, \varepsilon_m)$ is the vector of positive reserves ε_i for each i-th constraint, $1 \le i \le m$.

Trials and Partial Computability of Performance Characteristics

To select a trial option $w \in S$ within the above model (henceforth we will refer this operation as *executing a trial* at the point w) it is necessary to fugure out the corresponding values $f_i(w)$, $i \le i \le m+1$, of the performance characteristics. But it is could happen that some of the functions f_i, $1 \le i \le m+1$ are defined only at those points from S which satisfy a subset of several constraints. This is very often a case in the problems of optimal design, because if some necessary conditions for the object to function are not met, then some other characteristics of its performance may not be defined. Therefore, we admit that each function f_i, $1 \le i \le m+1$, is defined and computable only in the corresponding domain $Q_i \subset S$, where

$$Q_1 = S, \quad Q_{i+1} = \{w \in Q_i : \quad g_i(w) \le 0\}, \quad 1 \le i \le m \ .$$

This assumption, obviously, imposes the order in which the functions f_i, $1 \le i \le m+1$, are to be numbered. Note, that Q_{m+1} is the set of all admissible options.

Under this suppositions the initial problem of constrained optimization has to be rewritten in the form

$$\varphi(w^*) = \min\{f_{m+1}(w) : \ w \in Q_{m+1}\}$$

which is to be referred as the problem with *partially defined objective function and constraints*. This new type of constrained problem notably differs from the more traditional models with everywhere computable objective function and the left-hand sides of the constraints (see, for instance, [1] and the references given therein).

Any trial in this new model carried out at the point

$$w \in Q_\nu, \quad w \notin Q_{\nu+1}, \quad 1 \le \nu \le m+1 \ ,$$

causes some successive evaluations of the functions $g_i(w)$, $1 \le i \le \nu$, yielding the values

$$g_i(w) \le 0, \quad 1 \le i < \nu, \quad g_\nu(w) > 0 \ ,$$

where the last inequality is unessential if $\nu = m+1$. To compact the notations we introduced the complementary empty set Q_{m+2} and assumed that $g_{m+1} = f_{m+1}$.

As follows from the above discourse each point $w \in S$ is characterized by the index $\nu = \nu(w)$ ant this *index* being detracted a unity gives the number of constraints met at this point. In the sequel, the dyad

$$z = g_\nu(w), \quad \nu = \nu(w) \ ,$$

produced by a trial executed at the point $w \in S$ will be referred as the *outcome* of this trial.

Next important feature of the problem under consideration is that functions f_i, $1 \le i \le m+1$, are commonly not given by analytical formulas (at least some of them are not). Raising complexity of the object to be designed causes sophistication of the corresponding mathematical models. Consequently, the main (and very often the only) available tool for simulating the object's behaviour and assessing its performance is the computer-aided numerical analysis. Thus, each trial may be interpreted as running some *black box* to produce the outcome for the preset input $w \in S$. These runs could require substantial computer resources and therefore

should not be too numerous. As a result we have a problem of assessing the best point w^* (which is also called the *minimizer*) and the minimal value $\varphi^* = \varphi(w^*)$ with some limited amount of trials.

This means that the satisfactory estimates may not be attainable by the *item-by-item examination* of all possible variants (which is typical for the search procedures based on testing all the nodes of some uniform grid embedded into the domain of search). The insufficiency of this brute force approach drives to the necessity for some *purposeful* selection of options while searching for the best solution. Some mathematical schemes aiming to achieve this economic selection are based on the assumption that it is possible to approach the best option by some small sequential improvements of the current variant. Each of this small improvements is to be achieved by selecting the better options arising as the results of some *local* variations of the parameters characterizing the current version (see, for instance, [2]).

But for the models with essentially nonlinear (and nonconvex) functions f_i, $1 \le i \le m$, it is very often the case that the domain of search contains the set of points which are admissible and are the best in some of their vicinities, however they are worse than the minimizer w^*. In this type of problems, which are usually referred as the *multiextremal* ones, the local procedures are likely to be entrapped in the vicinity of one of such *local minimizers*. Therefore, they need to be supplied with some initial points from the *region of attraction* to the *global* minimizer w^*. Nevertheless this information is mostly not attainable. Multiple local runs from different (somehow scattered) initial points are also not able to assure the attainment of the global optimum.

It should be mentioned that in some problems local solutions may be absolutely not appropriate. If, for example, φ is the indicator of the reliability and we are interested in assessing the worst case, then we definitely need the estimate for φ^*. Approaches for searching the global solutions in multiextremal problems with the unknown region of attraction and the trial outcomes produced by running the black box are often based on the quite natural assumption that any limited change in the parameters of the object yields some limited changes in the characteristics of its performance. This assumption can be justified by the fact that in real systems the energy of change is always limited. One of the most popular mathematical formulations of this property is the *Lipschitz continuity condition*

$$|f_i(w'') - f_i(w')| \le L_i \|w'' - w'\|, \qquad w', w'' \in Q_i$$

which assumes that the differences of the function f_i, $1 \le i \le m+1$, are majorized with some measure

$$\|w'' - w'\| \le \{\sum\nolimits_{j=1}^{n} (w''_j - w'_j)^2\}^{1/2}$$

of differences in the argument multiplied by factor L_i. The values L_i, $1 \le i \le m+1$, are referred as *Lipschitz constants*.

The existence of these conditions provides the possibility to use the already obtained trial outcomes for classifying some subareas in the domain of search as definitely not containing the minimizer. This allows to economize the search effort by concentrating further trials in other subareas. Search techniques successively selecting trial points with the account of Lipschitz conditions substantially outperform the item-by-item examination methods (see, for instance, [3], [4]).

In the sequel we admit that the constrained problem under consideration is the multiextremal one and the corresponding performance characteristics are Lipschitzian.

Reduction to Core Unconstrained Problem

We commence by considering the core univariate case with the only real parameter x and the range of search $S = [a, b]$. In this case we may interpret the outputs of trials introduced above as the values of some real function

$$f(x) = g_\nu(x), \qquad \nu = \nu(x) \ ,$$

defined everywhere in the range $[a, b]$. This function is either a value of the left-hand side of the first constraint violated at the point x (if $\nu \le m$), or $f(x)$ is the value of the function $\varphi = g_{m+1}$ to be minimized (if $\nu = m + 1$). Figure 1 presents such a function (with arcs depicted by solid lines) for the case $m = 2$. For the sake of illustration, indexes of the points from the subranges Q_i, $1 \le i \le 3$, are plotted at the top of the picture. Not that the admissible set Q_3 is of two disconnected parts (marked by thick lines) and, thus, the problem is multiextremal.

FIGURE 1. Function $f(x)$ (the solid line) built with arcs of the partially defined functions $g_1(x)$, $g_2(x)$ and $\varphi(x)$. Function $\Phi(x)$ coincides with $f(x)$, while $x \in Q_1 \backslash Q_3$, and differs from $f(x)$ (the dotted line), while $x \in Q_3$

The main idea of our approach is to reduce the initial constrained problem to the unconstrained one

$$\Phi(x^*) = \min\{\Phi(x) : x \in [a, b]\} \ ,$$

where

$$\Phi(x) = \begin{cases} f(x), & \nu(x) \le m, \\ f(x) - \varphi^*, & \nu(x) = m + 1. \end{cases}$$

The sought value φ^* is not supposed to be known. At this stage of contemplation we consider just the existence of the function $\Phi(x)$ linked with the unconstrained problem. The new problem and the initial one have the coincident sets of minimizers, since the function $\Phi(x)$ is strictly positive at all points which are not solutions to the constrained problem, while it vanishes at the points of minimizers. This function (see the example presented on Figure 1) coincides with the function $f(x)$, while $x \in Q_1 \backslash Q_3$, and it does not coincide (see the dotted line) with $f(x)$, while $x \in Q_3$.

Note that, owing to our assumptions, the arcs $g_\nu(x)$, $x \in Q_3$, of the function $f(x)$ are Lipschitzian with constants L_ν. The same is true for the arcs of the

function $\Phi(x)$. Our next step is to annihilate these differences between the arcs by introducing the divided function

$$\psi(x) = \begin{cases} g_\nu(x)/L_\nu, & \nu \le m, \\ (g_\nu(x) - \varphi^*)/L_\nu, & \nu = m+1, \end{cases}$$

with the arcs being Lipschitzian in each set Q_ν, $1 \le \nu \le m+1$, with the unique constant $L = 1$. This function may have some discontinuities at the a priori unknown boundary points of the sets Q_ν. These points are jump discontinuities also of $f(x)$ and of $\Phi(x)$ (see Figure 1). Nevertheless it is still possible to evaluate the location of the global minimizer x^* having the outcomes of k trials executed at the points $x^1, \ldots, x^k$ from $[a, b]$.

Indeed, due to the Lipschitz condition, the value $\psi(x)$ is strictly positive if x satisfies the inequality

$$\min\{\psi(x^i) - |x - x^i| : \ 1 \le i \le k\} > 0 \ .$$

Figure 2 gives a particular example for the case $k = 4$, where the set of points satisfying the above inequality is presented by the union of 4 open subintervals in $[a, b]$ shaded below the real axis. Note that the function $\psi(x)$ on this picture corresponds to the problem represented on Figure 1.

FIGURE 2. The union of 4 open subintervals (marked by thick lines) does not contain the global minimizer

As a consequence, we obtain the inclusion

$$x^* \in \{x \in [a, b] : \ |x - x^i| \ge \psi(x^i), \ 1 \le i \le k\}$$

which can serve as an estimate for the minimizer. Actually, this estimate employs an additional assumption that each subset of arcs $\psi(x)$, $x \in Q_i$, of the function $\psi(x)$ has Lipschitzian (with the unique unity constant) continuation $\psi_i(x)$ throughout $[a, b]$ (i.e. $\psi(x) = \psi_i(x)$, $x \in Q_i$, $1 \le i \le m+1$) an we are to keep this in mind in further consideration.

Under the above conditions it is possible to assess the global minimizer by effectively exploring the function $\psi(x)$ with trials. To do so we have to assign some values to the Lipschitz constants (which are mostly unknown) and to the sought optimal value φ^*. We shall overcome these difficulties in some *adaptive way* by replacing all the above unknowns with their running estimates.

The suggested scheme for reducing the constrained problems to the unconstrained ones substantially differs from the approaches based on the introduction of *penalty functions* (which is typical for local optimization techniques; see, for instance, [5], [6]). This difference is not only in the possibility to treat partially defined constraints and the objective function, but also in the separate account of each constraint and in the termination of any trial with the advent of the first

violated constraint. Therefore, the suggested approach may be efficient in handling constrained problems also in the case when all functions $g_i(x)$, $1 \le i \le m+1$, are defined all over the range $[a,b]$. This is due to the fact that, within the suggested scheme, the trials omit evaluation of the values $g_{\nu+1},\ldots,g_{m+1}$ if $g_\nu > 0$, which reduces the necessary computational effort. Moreover, it is possible to additionally accelerate the solution of the constrained problem by numbering the left-hand sides g_i, $1 \le i \le m$, of the constraints in such an order that small numbers will correspond to the constraints which are less cumbersome to verify.

The exposed above ideas are employed in the algorithm described below. This algorithm selects the trial points reposing on the already obtained outcomes.

Index Method for Global Optimization

Algorithm. The first two trials are carried out at the points $x^0 = a$ and $x^1 = b$. The selection of any subsequent point x^{k+1}, $k \ge 1$, is determined by the rules:

1. Renumber the points $x^1,\ldots,x^k$ of the previous trials by subscripts in increasing order of the coordinate, i.e.,

$$a = x_0 < \cdots < x_i < \cdots < x_k = b\ ,$$

and juxtapose them the values $z_i = f(x_i)$, $1 \le i \le k$, computed at these points.

2. Construct the sets I_ν containing subscripts of all the trial points in which the values of the corresponding functions g_ν were estimated, i.e.,

$$I_\nu = \{i:\ 1 \le i \le k,\ \nu(x_i) \ge \nu\}, \quad 1 \le \nu \le m+1\ .$$

3. Compute the running lower bounds

$$\mu_\nu = \max\{|g_\nu(x_i) - g_\nu(x_j)|/(x_i - x_j):\ i,j \in I_\nu,\ i > j\}$$

for the unknown Lipschitz constants L_ν of the functions g_ν, $1 \le \nu \le m+1$. If the set I_ν contains less than two elements or if the above formula yields a zero value, assume that $\mu_\nu = 1$.

4. For all non-empty sets I_ν, $1 \le \nu \le m+1$, find the values

$$z^*_\nu = \begin{cases} \min\{z_i:\ i \in I_\nu\}, & I_{\nu+1} = \varnothing, \\ -\varepsilon_\nu, & I_{\nu+1} \ne \varnothing, \end{cases}$$

where $\varepsilon_R = (\varepsilon_1,\ldots,\varepsilon_m)$ is the preset vector of positive reserves introduced in the definition of an ε-reserved solution and $I_{m+2} = \varnothing$ (by definition). The running integer value ω meeting the conditions $I_\omega \ne \varnothing$, $I_{\omega+1} = \varnothing$ may be interpreted as the subscript of the 'running objective function' $g_\omega(x)$ the algorithm is minimizing. Thus, z^*_ω is an upper bound for the minimal value of this function. Some different functions may play this role in the course of search until ω attains the value $m+1$. In this last case z^*_{m+1} is the running estimate for φ^*.

5. Compute the characteristic $R(i)$ for each interval (x_{i-1}, x_i), $1 \le i \le k$, where

$$\begin{aligned} R(i) = \Delta_i + (z_i - z_{i-1})^2/r^2\mu_\nu^2\Delta_i - \\ -2(z_i + z_{i-1} - 2z^*_\nu)/r\mu_\nu, \qquad \nu = \nu(x_{i-1}) = \nu(x_i)\ , \\ R(i) = 2\Delta_i - 4(z_i - z^*_\nu)/r\mu_\nu, \qquad \nu = \nu(x_i) > \nu(x_{i-1})\ , \\ R(i) = 2\Delta_i - 4(z_{i-1} - z^*_\nu)/r\mu_\nu, \qquad \nu = \nu(x_{i-1}) > \nu(x_i)\ , \end{aligned}$$

$\Delta_i = x_i - x_{i-1}$, $1 \le i \le k$. The preset parameter $r > 1$ serves to ensure that the product $r\mu_\nu$ may be used as an upper bound for the Lipschitz constant L_ν (see Theorem 1 below).

6. Execute the next trial at the point

$$x^{k+1} = (x_t + x_{t-1})/2 - (z_t - z_{t-1})/2r\mu_\nu \ ,$$

where $\nu = \nu(x_{t-1}) = \nu(x_t)$; if $\nu(x_{t-1}) \ne \nu(x_t)$, then the second term in the right-hand side of the above equation has to be omitted. Integer t corresponds to the interval with the maximal characteristic $R(t) \ge R(i)$, $1 \le i \le k$.

These rules may be appended with the termination criterion which truncates the sequence of trials if $\Delta_t \le \varepsilon$, where $\varepsilon > 0$ is the preassigned accuracy of search. We will refer the described algorithm as *Index Method with Reserves* (IMR for short).

For the sake of illustration Figure 3 presents a constrained problem (case $m = 2$) solved by the IMR in the range $[0, 0.8]$ at $r = 3$, $\varepsilon_1 = \varepsilon_2 = 0.01$ and $\varepsilon = 0.0001$. The points of 50 trials executed in this run are marked by vertical strokes beneath the real axis. They are separated in three series with regard to the values of indexes; top, middle and bottom series respectively correspond to $\nu = 1$, $\nu = 2$ and $\nu = 3$. Observable concentration of trials corresponds to the vicinity of the global minimizer.

FIGURE 3. Objective function φ and functions g_1, g_2 from the left-hand sides of the constraints. Vertical strokes mark the points of trials generated by the IMR. Strokes separated in three series with regard to the value of indexes (increasing from top to bottom)

Theorem 1. (convergence conditions; see [7]). *Assume that the constrained problem has an ε-reserved solution and the following conditions are satisfied:*

1. *Each subrange Q_i, $1 \le i \le m+1$, is the finite union of positive length x-intercepts.*
2. *Each function $g_i(x)$, $x \in Q_i$, $1 \le i \le m+1$, admits some Lipschitzian (with the constant L_i) continuation throughout $[a, b]$.*
3. *Reserves corresponding to the constraints active at the minimizer (i.e., $g_\nu(x^*) = 0$) satisfy the inequalities*

$$0 < 2\varepsilon_\nu < L_\nu(\beta - \alpha) \ ,$$

where $[\alpha, \beta]$ is the admissible interval containing x^* (see the right one of the two shaded intervals on the Figure 1).

4. *Reserves corresponding to the constraints not active at the minimizer (i.e. $g_\nu(x^*) > 0$) satisfy either the above inequalities or the inequalities*

$$0 < \varepsilon_\nu < |g_\nu(x^*)| \ .$$

5. *Since some step, the inequalities*

$$r\mu_\nu > 2L_\nu, \qquad 1 \le \nu \le m+1 \ ,$$

are met for the running values μ_ν.

Then:

1. *The minimizer x^* is the limit point of the sequence $\{x^k\}$ generated by the IMR for the constrained problem at $\varepsilon = 0$ in the termination criterion.*
2. *Any limit point of the sequence $\{x^k\}$ is a solution to the constrained problem.*
3. *If the above conditions for the reserves are not met, then for any limit point $\bar{x}$ of the sequence $\{x^k\}$ is true that*

$$\varphi(\bar{x}) = \inf\{\varphi(x^k) : 1 \le k < \infty,\ x^k \in Q_{m+1}\} \le \varphi(x_\varepsilon) .$$

It is also proved (see [7]) that augmentation of the reserves (while the conditions of Theorem 1 are still met) speeds up the convergence process. For example, the solution of the problem from Figure 3, which terminated after 50 trials at $\varepsilon_1 = \varepsilon_2 = 0.01$, requires 42 trials at $\varepsilon_1 = \varepsilon_2 = 0.05$ and 37 trials at $\varepsilon_1 = \varepsilon_2 = 0.1$. But the interval $[\alpha, \beta]$ and the values $g_\nu(x^*)$ introduced in the conditions of Theorem 1 are mostly unknown which tangles implementation of these conditions for selecting reserves. To overcome these difficulties it is possible to propose some mechanism for adaptive selection of the running values for the reserves; see [7].

Multidimensional Case: Fractal Approach

As it was already mentioned, optimization techniques based on the assumption of Lipschitzian conditions and selecting each subsequent trial through analysis of all previously computed (and recorded) functions' values require substantially fewer trials than the plain item-by-item examination on some uniform grid of trial points. Unfortunately, such a selection turns into solving an auxiliary multidimensional problem of increasing multiextremality (along with the accumulation of trial outcomes) at each step of the search process. I.e., the *decision rules* of any such algorithm aiming to effectively use the acquired *search information* for reducing the amount of trials needed to estimate the sought optimum include an inherent multiextremal optimization problem of the same dimensionality. But, as it was demonstrated above, the univariate case is effectively solvable. Therefore it is of interest to present the multivariate constrained optimization problem by its univariate equivalent which may be effectively solved using the technique similar to the presented above.

A possible way to do so is to employ a single-valued Peano curve $y(x)$ continuously mapping the unit interval $[0, 1]$ on the x-axis onto the unit hypercube D and, thus, yielding the equality

$$\{y(x) : x \in [0, 1]\} = D ,$$

where

$$D = \{y \in R^n : -2^{-1} \le y_j \le 2^{-1},\ 1 \le j \le n\} .$$

These curves, first introduced by Peano (1890) and Hilbert (1891), 'fill' the cube D, i.e. they pass through every point of D, and this gave rise to the term *space filling curves*; see survey [8]. It is important to lay stress on the fact that while x passes an intercept in the interval $[0, 1]$ the corresponding image $y(x)$ covers some *volume* (but not a *line*) in the hypercube D. These curves having dimensions exceeding unity are typical examples of the so-called *fractal* objects.

By introducing transformation

$$y_j = (w_j - (a_j + b_j)/2)/\rho \ ,$$
$$\rho = \max\{b_j - a_j : \ 1 \le j \le n\} \ ,$$

and the extra constraint

$$g_0(y) = \max\{|y_j| - (b_j - a_j)/2\rho : \ 1 \le j \le n\} \le 0 \ ,$$

it is possible to present the initial constrained problem defined in the hyperinterval S as the one defined in the 'standard' domain of search D:

$$\varphi(y^*) = \min\{\varphi(y) : \ y \in D, \ g_i(y) \le 0, \ 1 \le i \le m\} \ .$$

Now it can be rewritten in the equivalent univariate form

$$\varphi(y^*) = \varphi(y(x^*)) = \min\{\varphi(y) : \ x \in [0,1], \ g_i(y(x)) \le 0, \ 1 \le i \le m\} \ .$$

Notations $g_i(y)$, $1 \le i \le m$, are used for brevity; to be precise we had to write $g_i(w(y))$. The employed transformation of the hyperinterval S into the unit cube D aims not to alter the relations of Lipschitzian properties in dimensions.

Space-filling curves $y(x)$ are not presentable by any analytical formula. They are defined as the limit objects linked with some successive joint partitioning of the ranges for the argument ($x \in [0,1]$) and for the value ($y \in D$). Let us consider an important example of such a scheme ascending to Hilbert (see [7]).

Partition Scheme. Divide the cube D into 2^n equal hypercubes of the 'first partition' by cutting D with n mutually orthogonal hyperplanes (each plain is parallel to one of the coordinate ones and passes through the middle points of the D edges orthogonal to this hyperplane; note that each of these subcubes has the edge length equal to 2^{-1}. Use index z_1, $0 \le z_1 \le 2^n - 1$, to number all the subcubes obtained in the above partitioning; each particular subcube is, henceforce, designated $D(z_1)$. Then divide (in the above manner) each of the obtained first-partition cubes into 2^n second-partition subcubes numbered with the index z_2, $0 \le z_2 \le 2^n - 1$. Each particular subcube obtained by partitioning of $D(z_1)$ is designated $D(z_1, z_2)$ and it has the edge length equal to 2^{-2}; see Figure 4 (case $n = 2$). Consequently cutting each hypercube of a current partition into 2^n subcubes of the subsequent partition (with a twice shorter edge length) obtain hypercubes $D(z_1, \ldots, z_M)$ of any M-th partition satisfying the inclusions

$$D \supset D(z_1) \supset D(z_1, z_2) \supset \cdots \supset D(z_1, \ldots, z_M) \ .$$

FIGURE 4. Subcubes of the second partition (case $n = 2$, $M = 2$)

Next, cut the interval $[0,1]$ into 2^n equal parts. Each particular part is designated $d(z_1)$, $0 \le z_1 \le 2^n - 1$. The numeration streams from left to right along the x-axis. Then, once again, cut each of the above parts into 2^n smaller (equal) parts. etc. Designate $d(z_1,\ldots,z_M)$ the subinterval of the M-th partition. The length of such an interval is equal to 2^{-Mn}, i.e., it is equal to the volume of the subcube of M-th partition. Assume that each interval contains its left end-point, but it does not contain its right end-point. The only exception is the right end-point equal to unity: it belongs to its interval. Obviously,

$$[0,1] \supset d(z_1) \supset d(z_1,z_2) \supset \cdots \supset d(z_1,\ldots,z_M) \ .$$

Definition of $y(x)$. Present the left-end point v of the subinterval

$$d(z_1,\ldots,z_M) = [v, v + 2^{-Mn})$$

in the binary form

$$0 \le v = \sum_{i=1}^{Mn} \alpha_i 2^{-i} < 1 \ ,$$

where $\alpha_1, \alpha_2, \ldots, \alpha_{Mn}$ are binary digits. Then the indexes z_j, $1 \le j \le M$, are presentable as

$$z_j = \sum_{i=0}^{n-1} \alpha_{jn-1} 2^i, \qquad 1 \le j \le M \ .$$

This allows to refer the subinterval $d(z_1,\ldots,z_M)$ also as $d(M,v)$. Now establish the mutually single-valued correspondence between all the subintervals of any particular M-th partition by accepting that $d(M,v) = d(z_1,\ldots,z_M)$ corresponds to $D(z_1,\ldots,z_M)$ and vice versa. With account of the accepted correspondence, the above subcube may also be referred as $D(M,v)$.

Theorem 2. *Assume that:*

1. *$y(x)$ is a correspondence defined by the assumption that for any $M \ge 1$ the image $y(x) \in D(M,v)$ if and only if $x \in d(M,v)$.*
2. *Two subintervals $d(M,v')$ and $d(M,v'')$ have a common end-point (which is either v' or v'') if and only if the corresponding subcubes $D(M,v')$ and $D(M,v'')$ have a common face (i.e., these subcubes must be contiguous).*

Then:

1. *$y(x)$ is a single-valued continuous mapping of the unit interval $[0,1]$ onto the hypercube D.*
2. *If $g(y)$, $y \in D$, is Lipschitzian with some constant L, then the univariate function $g(y(x))$, $x \in [0,1]$, satisfies the univariate Hölder conditions:*

$$|g(y(x'')) - g(y(x'))| \le 2L\sqrt{n+3}(|x'' - x'|)^{1/n}, \quad x', x'' \in [0,1] \ .$$

Second condition of the theorem is possible to assure by using some special scheme for the numeration of subcubes $D(z_1,\ldots,z_M)$; see [7]. Figure 5 demonstrates the ordering of the subcubes of the fourth partition provided by this numeration

FIGURE 5. Subcubes of the fourth partition (case $n = 2$, $M = 4$). Broken line stitches the subcubes in the order of numbering passing through the common faces of the adjacent subcubes

(case $n = 2$). The broken line stitching the subcubes in the order of numbering goes through the common faces of the adjacent subcubes.

To compute the image $y(x)$ with some preset accuracy $\varepsilon > 0$ (in coordinates) it is possible to determine the subcube $D(M, v)$, $M > \log_2(\varepsilon^{-1}) - 1$, containing this image and then to use the central point (center of gravity) of this subcube as the approximation for $y(x)$. Algorithms for constructing such approximations, their theoretical justification, and the standard routines actualizing these algorithms are given in [7].

Space-filling curves are not smooth in any x-intercept. But (as it was already discussed) we do not rely on the local behaviour of the performance characteristics. More important is that limitations set by Lipschitz conditions are kept under Peano transform as uniform Hölder conditions (see the second statement of Theorem 2). This allows to tune the already considered algorithm (IMR) for the univariate problems for solving the (reduced to one dimension) multidimensional problems. The main adjustment is in the substitution of the new measure of distances

$$\Delta_i = (x_i - x_{i-1})^{1/n}$$

in the formulas reentrant in the decision rules of the IMR.

Figure 6 presents the example of the constrained problem (case $n = 2$, $m = 3$) solved by the described technique. Admissible points are reposing in the outlined circle (the left-hand part of the first constraint is a quadratic function), on and above the ellipse line (the left-hand part of the second constraint is also a quadratic function), on and below the sine line (the left-hand part of the third constraint is proportional to the sine function of the first argument). These points conform the three disconnected parts constituting the admissible set. Level lines of the objective function are also plotted within the quadratic domain of search. Points of 141 executed trials are marked by the dark dots. Concentration of trials in the vicinity of the minimizer is noticeable. Trials (65) outside the circle terminated after checking the first constraint. Those (30) in the circle (but below the ellipse line) required checking of two constraints and the ones (20) in the crescent (but not admissible) - checking of three constraints. Only for trials (26) from the three admissible subsets all the performance characteristics were computed.

FIGURE 6. Level sets of the objective function and zero-level sets of the left-hand sides of the constraints (case $n = 2$, $m = 3$). Dark dots mark the trial points

Non-Redundant Parallel Algorithms

The advent of parallel computers has created conditions for the elaboration of methods which can accelerate the finding of a solution to many applied problems (see [9], [10],[11]). In the case of global optimization an usage of parallel computation is extremely attractive because the solving of global optimization problems is very time consuming: the time taken to evaluate the objective function at a point being long and the number of trials needed being high even when the most efficient numerical techniques are used (see [7], [10]).

As a rule, when parallel methods are proposed, either the problem to be solved or some sequential method, have an inherent parallelism and are used as the source of elaboration. Since the first type of parallelism depends greatly on the specific nature of the problem being solved and must be defined separately for every single case, attention is concentrated on the second way of parallelization. Naturally, the sequential method taken as the basis for parallelization must have elevated estimates of the convergence speed for the class of problems under examination, because there is no sense in using a parallel method if a more rapid sequential one already exists. That is why the new fast sequential global optimization algorithm introduced previously can be used as a basis for the elaboration of parallel methods. Of course, parallel computations can also be used for an acceleration in the analysis of the mathematical model of the object being optimized, i.e., for an acceleration of the computations needed to evaluate the objective function at a given point. But the organization of such an acceleration has its own particularity for every class of models, as the construction of the principles of a parallel choosing of trial points refers properly to the optimization algorithms.

The functioning of the sequential global optimization methods analyzed as the basis for the creation of parallel algorithms in general can be described as follows. At every step, $k+1$, of the sequential method, the new trial point, x^{k+1}, is chosen by using the search information obtained during k previous trials executed at the points $x^1, \ldots, x^k$, where the value of the objective function has been evaluated.

As has already been mentioned, in many real problems executing the trial at even one point takes too much time and the number of trials can itself be very high even when the most efficient sequential methods are used. In solving these problems, parallel (simultaneous) trials on the parallel processors can significantly accelerate the search. This means that at every step, $l+1$, of the parallel method, on the basis of information obtained during the l previous iterations (when $k(l)$ trials at the points $x^1, \ldots, x^{k(l)}$ were made), p points $x^{k(l)+1}, x^{k(l)+2}, \ldots, x^{k(l)+p}$ are chosen and the trial at each of them is carried out on a separate parallel processor. Characteristics for evaluating the efficiency of such parallelization can now be introduced.

One of the most important estimates of the efficiency of the parallel method in problem solving is speed up in time obtained due to parallelization. Let $T(1)$ be the time required to solve the problem by use of a sequential algorithm and $T(p)$ the solution time of the same problem by the multiprocessor system with p parallel processors. Let us call the value

$$S(p) = T(1)/T(p)$$

the *speed up in time* obtained by application of the parallel algorithm when compared with the sequential one. The other useful way of estimating the parallel

algorithms efficiency is the *speed up in iterations*

$$s(p) = n(1)p/n(p) \ ,$$

where $n(1)$ is the number of trials required by the sequential method in the problem solving, and $n(p)$ the number of trials required in using the parallel algorithm with p processors to solve the same problem.

Many applications exist where the time taken in evaluating the objective function $\varphi(x)$ is constant. Taking advantage of this peculiarity helps to organize parallelization in the most efficient way and later investigations will consider this kind of global optimization problems.

We can now correlate the values $S(p)$ and $s(p)$. Let T be the time required to evaluate $\varphi(x)$ at one point and t the time taken in realizing the algorithm's decision rule, i.e., the time required to choose the new trial point. In the methods analyzed, this time depends on the quantity of accumulated search information and on the number of parallel processors (see [7]) and can be approximately described by the following formula

$$t \cong t(n(p)) = \alpha n(p)/p \ ,$$

where $\alpha > 0$ is a parameter. The value $S(p)$ can then be calculated as follows

$$S(p) \cong \frac{Tn(1) + \alpha(1 + 2 + \cdots + n(1))}{Tn(p)p^{-1} + \alpha(p + 2p + \cdots + n(p))p^{-1}} \ .$$

In the case $T \gg t$ we obtain that

$$S(p) \to n(1)p/n(p) = s(p) \ .$$

In the case $t \gg T$, it follows that

$$S(p) \to s(p)^2 \ .$$

This means that $s(p)$ has a decisive influence on $S(p)$ and conditions which ensure the maximal levels of $s(p)$ will be discussed further.

Naturally, in using p parallel processors it is desirable to obtain the speed up $s(p) = p$ but this, unfortunately, is not always possible. The problem consists in the fact that the accepted parallelization scheme is not trivial from the information point of view (as happens, for example, when a cycle of assignment in programming is parallelized). This point will now be discussed in detail.

Consider a sequential method (SEQ) and its parallel version (PAR). In choosing the point x^{k+1} at the $(k+1)$-th iteration, the SEQ has all the information hitherto obtained during the previous k iterations at the points $x^1, \ldots, x^k$. On the basis of the same information, the PAR chooses not one but $p > 1$ points $x^{k+1}, x^{k+2}, \ldots, x^{k+p}$. This means that the choice of the point x^{k+j} is made *in the absence* of information concerning the results of the trials at the points $x^{k+1}, \ldots, x^{k+j-1}, 2 \le j \le p$. Consequently, only the first point, x^{k+1}, is defined on the basis of full information and coincides with the point chosen by the sequential method. The other $p-1$ trials of the parallel method can be made at those points which differ from those at which the SEQ effects its trials. These trials can slow

down the search and so reduce the efficiency of use of the parallel processors. Let us call these trials *redundant*, and the value

$$z(p) = \begin{cases} (n(p) - n(1))/n(p), & n(p) > n(1), \\ 0, & n(p) \leq n(1), \end{cases}$$

the *redundancy* of the parallel method. It is evident that

$$s(p) = p(1 - z(p)) \ .$$

It can be seen from these formulas that in minimizing the redundancy speed up is maximized. In the non-redundant search case, i.e., when $z(p) = 0$, a speed up equal to the number of parallel processors used is obtained.

Let us consider a parallel version of the method introduced above. It can be taken as a basis for parallelization because of the following reasons:

1. it belongs to the class of information methods having quite satisfactory estimates of the convergence speed (see [7]) when compared with other methods working with the Lipschitz functions;
2. the probabilistic nature of the algorithm allows its natural generalization to the parallel case;
3. the accepted idea of parallelization can be generalized and applied to a wide class of sequential global optimization algorithms (see [7]);
4. the same way of parallelization can be generalized for the elaboration of multi-dimensional parallel global optimization methods using the fractal approach introduced previously.

In order to concentrate our efforts on parallelization, the simplest case - where the feasible region is an interval (a, b), i.e., the problem of finding

$$\varphi(x^*) = \min\{\varphi(x) : x \in (a, b)\} \ ,$$

where the objective function $\varphi(x)$ is multiextremal and satisfies the Lipschitz condition - will be considered. Let us describe the information algorithm with parallel trials (IAPT).

Algorithm. The trials of the first iteration are made at $p = p(1) \geq 1$ arbitrary internal points of the interval (a, b). Suppose, now, that $n \geq 1$ iterations of the method have already been executed. The trial points of the next $(n+1)$-th iteration are then chosen by using the following rules:

1. Order trial points $x^1, \ldots, x^{k(n)}$ of the previous $k(n)$ trials by increasing their coordinates, where

$$k(n) = p(1) + p(2) + \cdots + p(n),$$

$$a = x_0 < x_1 < x_2 < \cdots < x_i < \cdots < x_k < x_{k+1} = b \ ,$$

and the points $a = x_0$, $x_{k+1} = b$ have been included in the row to assure the unique description of all the subintervals $[x_{i-1}, x_i], 1 \leq i \leq k+1$, of the interval $[a, b]$.

2. Calculate the value

$$M = \max\{|z_i - z_{i-1}|/(x_i - x_{i-1}) : 1 < i \leq k\} \, ,$$

where $z_i = \varphi(x_i)$ are the results of the trials effected at the points $x_i, 1 \leq i \leq k$, and the values z_0 and z_{k+1} are not defined.

3. Calculate the characteristic $R(i)$ of every subinterval $[x_{i-1}, x_i]$, $1 \le i \le k+1$:

$$R(i) = \begin{cases} \Delta_i + \frac{(z_i - z_{i-1})^2}{M^2 \Delta_i} - \frac{2(z_i + z_{i-1})}{rM}, & 1 < i \le k, \\ \Delta_1 - 4z_1/(rM), & i = 1, \\ \Delta_{k+1} - 4z_k/(rM), & i = k+1, \end{cases}$$

where $r > 1$ is the reliability parameter of the method and $\Delta_i = x_i - x_{i-1}$.

4. Order the characteristics $R(i)$ by decreasing the indexes $t_j, 1 \le j \le k+1$:

$$R(t_1) \ge \cdots \ge R(t_j) \ge \cdots \ge R(t_{k+1}) \ .$$

5. Execute the next $p = p(n+1)$ trials of the $(n+1)$-th iteration at the points $x^{k+1}, \ldots, x^{k+p}$ where

$$x^{k+j} = (x_{t_j} + x_{t_j - 1})/2 - \begin{cases} 0, & t_j = 1 \text{ or } t_j = k+1, \\ (z_{t_j} - z_{t_j - 1})/(2rM), & 1 < t_j \le k, \end{cases}$$

and $t_j, 1 \le j \le p$, are indexes of the first p intervals having highest characteristics. It is assumed that

$$p = p(n+1) \le \min\{k(n) + 1, Q\}, n \ge 1 \ ,$$

where $Q \ge 1$ is the constant defining the maximum number of processors which can be used for the parallelization.

The stop condition will terminate the search when, for at least one index $t_j, 1 \le j \le p$, the inequality

$$x_{t_j} - x_{t_j - 1} \le \varepsilon \ ,$$

holds, where $\varepsilon > 0$ is a given search accuracy.

The algorithm described generalizes, for the parallel case, the sequential information algorithm (SIA), the calculation scheme of which can easily be obtained from the decision rules described above by assuming

$$p(n) = 1, \quad n \ge 1 \ .$$

This type of parallelization of the SIA is based on the following argument. The sequential information algorithm SIA was constructed on the basis of the stochastic model, according to which the interval having the maximal characteristic can be interpreted as being the interval with the highest probability of containing the global minimizer. In the stochastic model, the intervals $[x_{i-1}, x_i], 1 \le i \le k+1$, are ordered according to the decreasing probability of the global minimizers being localized in them.

It is natural to assume that, in the case when, at the $(n+1)$-th iteration, there are $p = p(n+1) > 1$ parallel processors, p parallel trials will simultaneously be carried out at the intervals having p highest characteristic; namely, at those p intervals having the highest probability of the global minimizers being localized in them. This idea of parallelization has shown itself to be very fruitful and has been

generalized for certain other global optimization algorithms (numerous examples can be found in [7]). It can be shown that this kind of parallelism does not produce limit points different from those of the purely sequential scheme.

Let us illustrate the algorithms by numerical experiments executed with Shubert's test function (see Figs. 7–9). In Fig. 7 behaviour of the SIA is presented. Vertical strokes under the graph of the test function represent trials done by the SIA. The diagram below shows (from top to bottom) dynamics of the search. The stop condition has been satisfied after 42 trials. Fig. 8 shows the behaviour of the IAPT with two parallel processors. The algorithm has done 39 trials during 20 iterations. Trials executed by the first processor are linked on the dynamics diagram. It can be seen from the diagram that after the first initial iteration two trials are executed during every iteration (these trials are linked by horizontal lines). Fig. 9 presents results of minimization with three parallel processors. During 15 iterations 42 trials have been done (one during the first iteration, two during the second one, and three trials during the remaining iterations). In all three experiments the first trial has been done at the same point. Thus, in both cases where the IAPT worked with parallel processors no redundant trials have been generated.

FIGURE 7. Dynamics of the global search executed by the sequential algorithm

FIGURE 8. Dynamics of the global search executed by the parallel algorithm working with two parallel processors

FIGURE 9. Dynamics of the global search executed by the parallel algorithm working with three parallel processors

Let us study theoretical conditions of non-redundant parallelism. We denote by $\{x^k\}$ and $\{y^m\}$ the trial sequences generated by the sequential and parallel algorithms, respectively, for the same function $\varphi(x), x \in (a,b)$, at $\varepsilon = 0$ in the stop condition (infinite search). The following theorems (see [7]) show how many parallel processors Q can be used in order to achive the non-redundant search and, consequently, high levels of speed up.

Theorem 3. *Let the number Q of parallel processors used during the search by the parallel algorithm IAPT be equal to 2. Assume that the following conditions are satisfied:*

1. *The initial trial points are the same for the sequential and parallel methods, i.e., $y^1 = x^1$.*
2. *In both algorithms for the value M a constant exceeding the Lipschitz constant L is used and the choice of the parameter r ensures sufficient convergence conditions of both methods, i.e.,*

$$M \geq L,\ r \geq 2\ .$$

Then:

1. *If x^* and x^{**} are global minimizers of the function $\varphi(x)$ and the first trial has been executed at a point y^1 such that $x^* < y^1 < x^{**}$, then $s(2) = 0$.*
2. *If the reliability parameter $r > 2 + \sqrt{5}$, then $s(2) > 1.66$.*

This theorem establishes the conditions in which use of the IAPT with two parallel processors accelerates the search twice. As the objective function $\varphi(x)$ is multiextremal, a quite natural supposition about existence of a local minimizer x' can be made. This supposition enables us to establish conditions in which it is possible to use more parallel processors with a bounded number of redundant trials and, therefore, to obtain a major speed up of the search.

Theorem 4. *Assume that conditions of Theorem 3 hold and let the inequality*

$$\varphi(x') - \varphi(x^*) \leq \delta, \quad \delta > 0,$$

be fulfilled. Then, if there exists a trial point $y^{\hat{n}} \in \{y^m\}$ such that $x' \leq y^n \leq x^*$ or $x^* \leq y^n \leq x'$ takes place, the following assertions hold for the IAPT and $n > \hat{n}$:

1. *For $Q = 4, r > 2 + \sqrt{5}$ it follows that $s(4) > 3.32$, while*

$$x_{t_j} - x_{t_j - 1} > \overline{\overline{\varepsilon}}\ , \quad 1 \leq j \leq p\ .$$

2. *For $Q = 3, r > 2 + \sqrt{5}$ are used in the IAPT while*

$$\overline{\varepsilon} < x_{t_j} - x_{t_j - 1}, \quad 1 \leq j \leq p\ ,$$

and

$$x_{t_j} - x_{t_j - 1} \leq \overline{\overline{\varepsilon}}\ ,$$

at least for one $j, 1 \leq j \leq p$, it then follows that $s(3) > 2.66$.

3. *For $Q = 2, r > 1 + \sqrt{2}$, it follows that $s(2) = 0$, while*

$$x_{t_j} - x_{t_j - 1} > \overline{\varepsilon}\ , \quad 1 \leq j \leq p\ .$$

In all the assertions $\overline{\varepsilon}$ and $\overline{\overline{\varepsilon}}$ are calculated in the following way:

$$\overline{\varepsilon} = \frac{4\delta r}{M\,(r^2 - 2r - 1)}, \quad \overline{\overline{\varepsilon}} = \frac{4\delta r}{M\,(r^2 - 4r - 1)}.$$

The introduced theorems show how to use parallel computers without redundancy. If the user has additional information about the structure of the problem,

then it is possible to ensure high levels of speed up. More difficult is the problem, higher is the number of processors which can be used without generation of the redundant trials. For example, if the problem has 3 local minimizers close to the global one, then it is possible to use up to 8 parallel processors with the speed up close to the number of applied processors.

The described algorithm can be generalized for solving multidimensional problems with multiextremal constraints by using the fractal approach and the index scheme for both cases of synchronous and asynchronous computations (see [7]).

Test Example in Five Dimensions

Consider the problem of minimizing the function

$$\varphi(w) = \sin(xz) - (yv + zu)\cos(xy)$$

over the domain of search

$$-3 \le x, y, z \le 3, \quad -10 \le u, v \le 10 \ ,$$

subject to the functional constraints:

$$g_1(w) = -(x + y + z + u + v) \le 0 \ ,$$

$$g_2(w) = (y/3)^2 + (u/10)^2 - 1.4 \le 0 \ ,$$

$$g_3(w) = 3 - (x+1)^2 - (y+2)^2 - (z-2)^2 - (v+5)^2 \le 0 \ ,$$

$$g_4(w) = 4x^2 \sin x + y^2 \cos(y+u) +$$
$$+ z^2[\sin(z+v) + \sin(10(z-u)/3)] - 4 \le 0 \ ,$$

$$g_5(w) = x^2 + y^2[\sin((x+u)/3 + 6.6) + \sin((y+v)/2 + 0.9)]^2 -$$
$$-17\cos^2(z+x+1) + 16 \le 0 \ ;$$

$$w = (x, y, z, u, v) \in S$$

where $S \subset R^5$ is the described above domain of search.

The problem was solved by employing a uniform grid technique with the mesh width equal to 0.01 in each coordinate, which required (with account of some symmetry allowing to reduce the domain of search) about 10^{12} trials. The obtained best admissible node

$$w' = (-0.06, 2.20, 2.40, 9.28, 9.63)$$

characterized by the value $\varphi(w') = -43.22$ after it was locally refined with accuracy 0.0001 in each coordinate, yielded the value $\varphi^* = -43.2601$ at the point

$$w^* = (-0.0521, 2.2041, 2.3911, 9.2747, 9.6389).$$

FIGURE 10. xz-section (top) and uv-section (bottom) of the domain of search for the 5-dimensional test problem passing through the point w^* (marked by the dark dot). Bottom picture also presents the two-dimensional section of the admissible set

To illustrate the complexity of this problem Figure 10 presents two two-dimensional sections of the domain of search S passing through the point w^*; the image of this point is marked on both sections by the dark dots. Top picture corresponds to the xz-section, i.e. it gives the image of the square containing all the points

$$w = (x, 2.204, z, 9.275, 9.639) \in S \ ,$$

$-3 \leq x, z \leq 3$. Lines on this picture furnished with integers $4, 5$ plotted in the vicinities of these lines present the loci of the points w satisfying the respective equations $g_i(w) = 0$; the above numbers $i = 4, 5$ are plotted inside the admissible area for the corresponding constraints $g_i(w) \leq 0$. Level lines of the objective function φ are also plotted on the picture.

The bottom picture exposing the uv-section of S is drawn in a somewhat different way – it also presents the uv-section of the *admissible set* for the problem under consideration but all the lines outside the admissible set are deleted. Note that $-10 \leq u, v \leq 10$ though both squares on the pictures are drawn as if they are of the same size.

Next, the problem was transformed to the standard cubic domain of search D and solved by the above described Index Method employing the Peano curve $y(x)$ approximated by the centers of the 11th partition subcubes.

The search process required in total 59697 trials and in each of these trials the auxiliary zero constraint was tested. The functions $g_1 \div g_6$ were computed respectively $k_1 = 22915$, $k_2 = 22307$, $k_3 = 20948$, $k_4 = 20802$, $k_5 = 17472$, $k_6 = 3877$ times (mind that $g_6 = \varphi$). Settings for all the parametrs of the algorithm are given in [7].

The obtained estimate

$$w'' = (-0.068, 1.962, 2.431, 9.833, 9.833)$$

characterized by the value $\varphi(w'') = -42.992$ after it was locally refined with the accuracy 0.0001 in each coordinate yielded the value $\varphi^{**} = -43.2985$ at the point

$$w^{**} = (-0.0679, 1.9434, 2.4512, 9.9013, 9.9008) \ .$$

which is better (in the objective function value) than the estimate obtained by the uniform grid technique.*

*This experiment was run on Pentium-3 computers granted to the University of Nizhni Novgorod by Intel Technologies Inc. All the programming for the experiments was done by Prof. V.Gergel.

References

[1] R. Horst and P.M. Pardalos (Eds.), *Handbook on Global Optimization*, Kluwer Academic Publishers, Dordrecht, 1995.

[2] C.T. Kelly, *Iterative Methods for Optimization*, SIAM, Philadelphia, 1999.

[3] J. Pintér, *Global Optimization in Action (Continuous and Lipschitz Optimization: Algorithms, Implementations and Applications)*, Kluwer Academic Publishers, Dordrecht, 1996.

[4] Yu.G. Evtushenko, *Numerical Optimization Techniques*, Translation Series in Mathematics and Engineering, Optimization Software Inc., New York, 1985.

[5] A.V. Fiacco and G.P. McCormic, *Nonlinear Programming: Sequential Constrained Minimization Techniques*, SIAM, Philadelphia, 1990.

[6] D.P. Bertsekas, *Constrained Optimization and Lagrange Multiplier Methods*, Athena Scientific, Belmont, 1996.

[7] R.G. Strongin and Ya.D. Sergeyev, *Global Optimization with Non-Convex Constraints: Sequential and Parallel Algorithms*, Kluwer Academic Publishers, Dordrecht, 2000.

[8] H.Sagan, *Space-Filling Curves*, Springer-Verlag, New York, 1994.

[9] Bertsekas, D.P., and Tsitsiklis, J.N., *Parallel and distributed computation: numerical methods*, Prentice-Hall,Englewood Cliffs., 1989.

[10] Migdalas, A., Pardalos, P.M., and Storøy, S. (Eds.), *Parallel Computing in Optimization*, Kluwer Academic Publishers, Dordrecht, 1997.

[11] Roosta, S.H., *Parallel Processing and Parallel Algorithms: Theory and Computation*, Springer Verlag, New York, 2000.

List of Figures

University of Nizhni Novgorod, 23, Gagarin pr., Nizhni Novgorod, Russia
E-mail address: strongin@unn.ac.ru

ISI-CNR, c/o DEIS, University of Calabria, Rende (CS), Italy
E-mail address: yaro@si.deis.unical.it

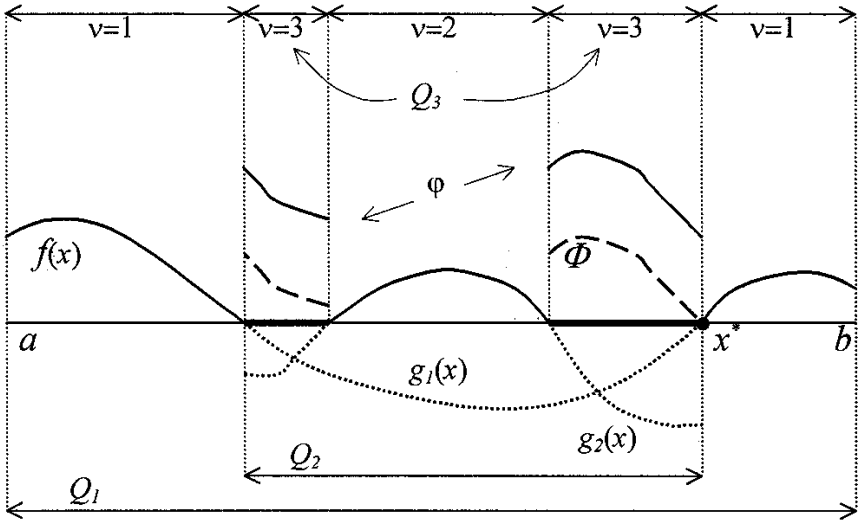

Figure 1

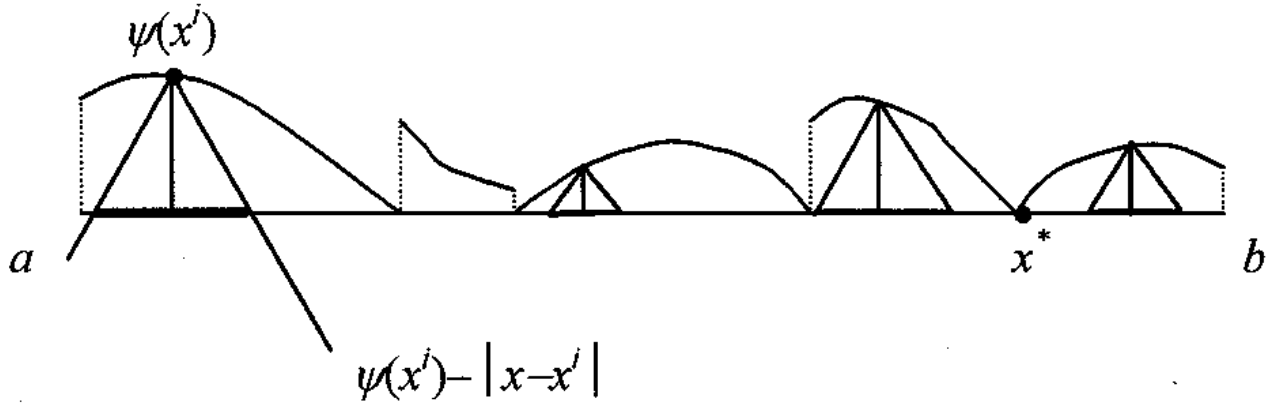

Figure 2

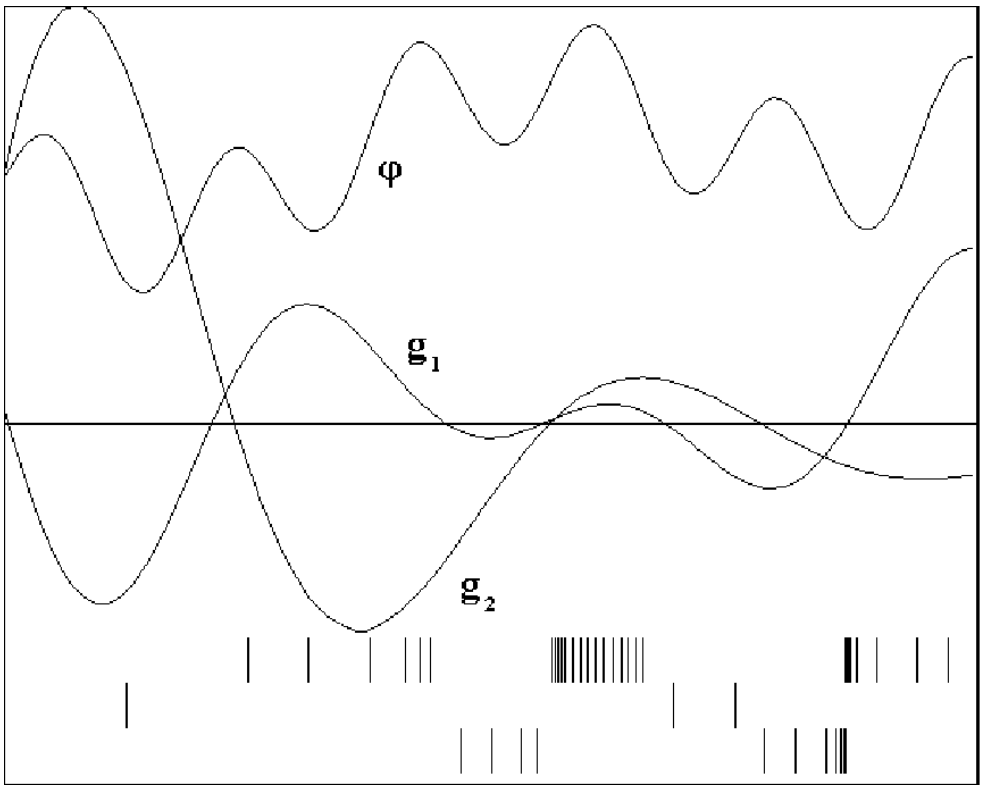

Figure 3

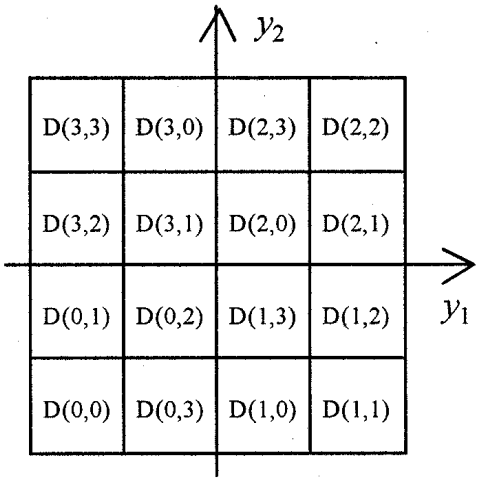
y_2
D(3,3)
D(3,0)
D(2,3)
D(2,2)
D(3,2)
D(3,1)
D(2,0)
D(2,1)
D(0,1)
D(0,2)
D(1,3)
D(1,2)
D(0,0)
D(0,3)
D(1,0)
D(1,1)
y_1

Figure 4

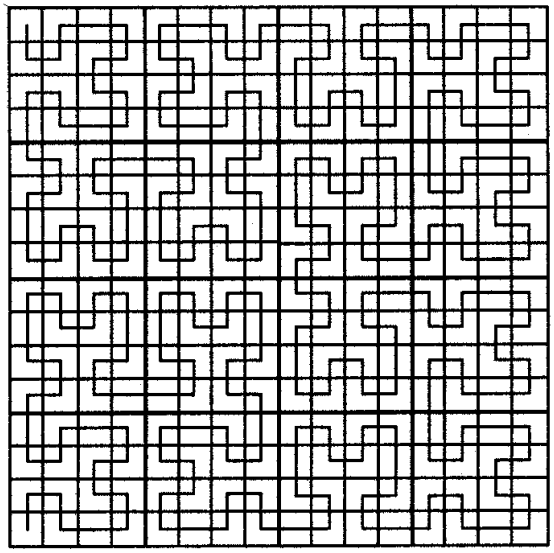

Figure 5

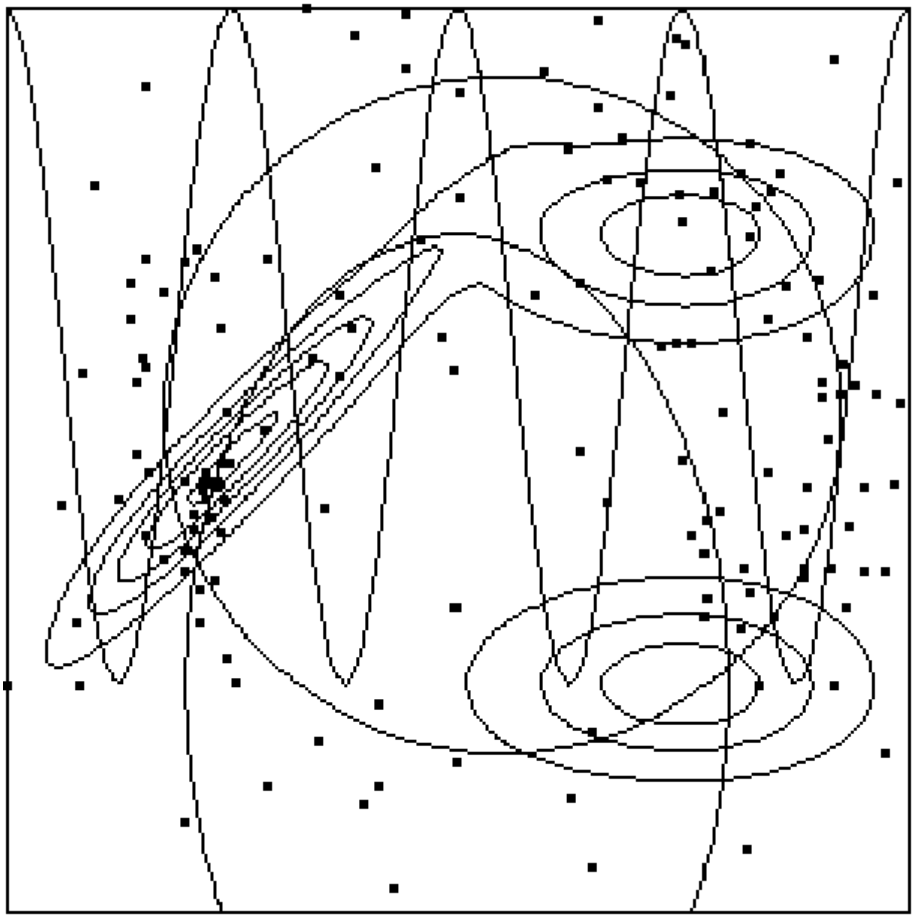

Figure 6

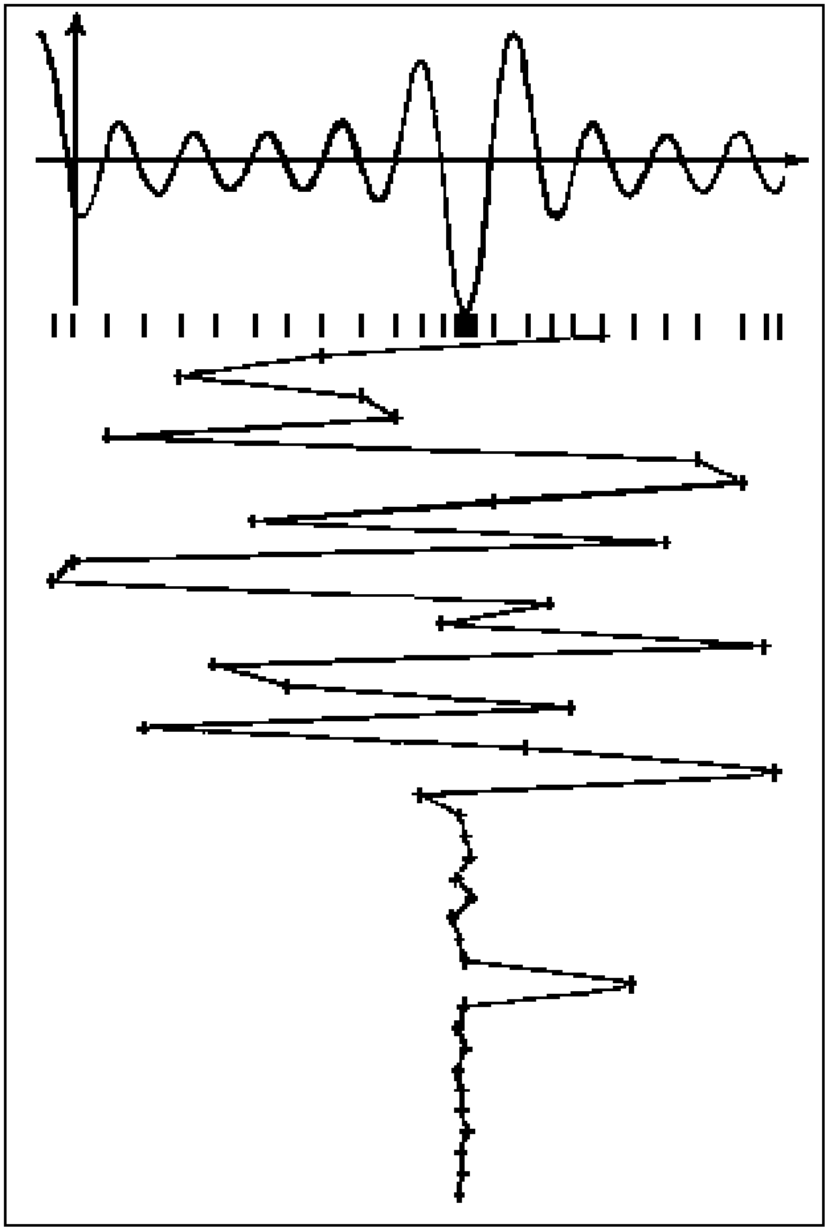

Figure 7

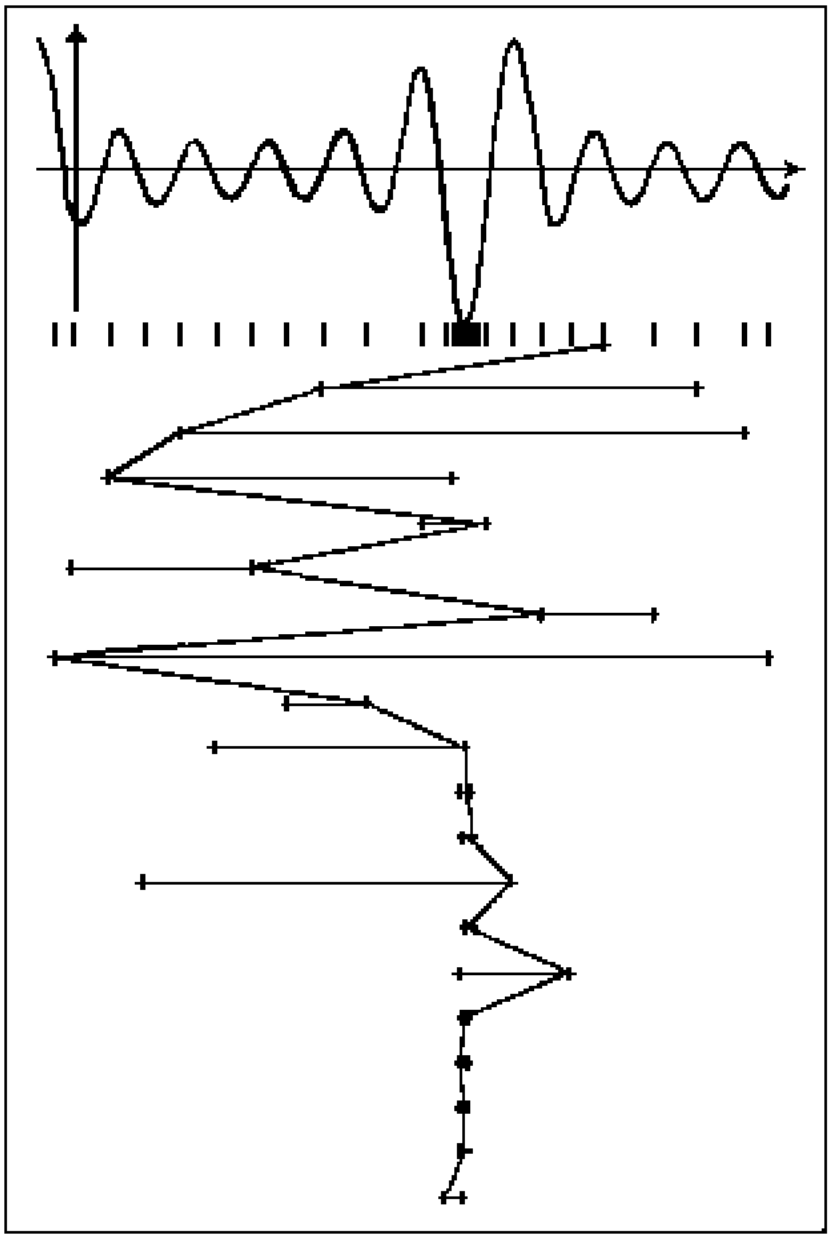

Figure 8

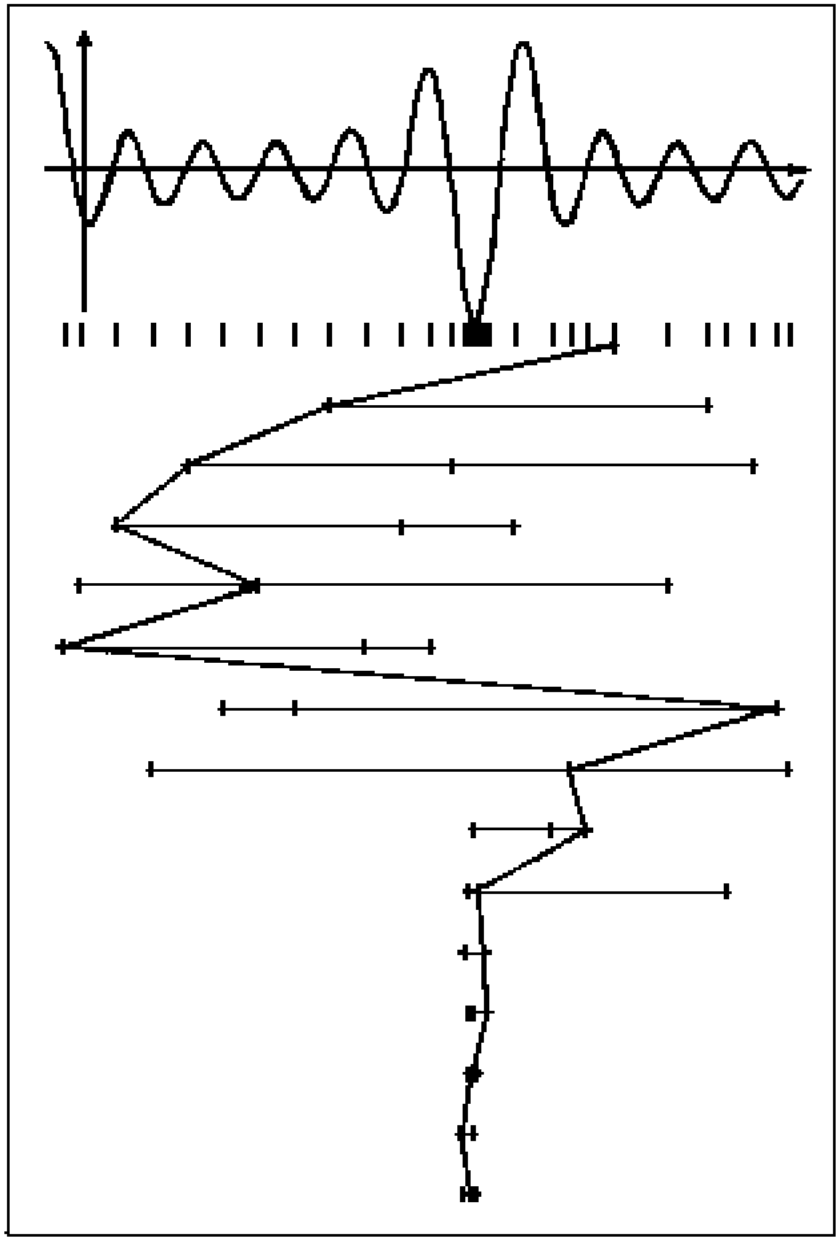

Figure 9

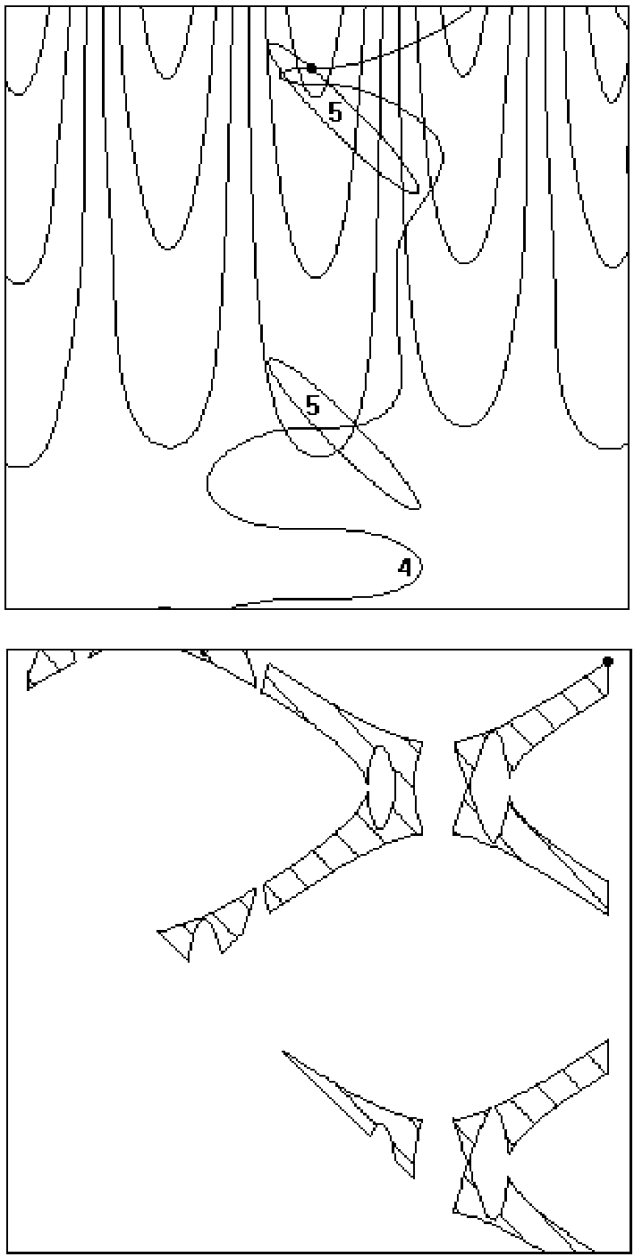

Figure 10