On the Efficiency of a Global Non-differentiable Optimization Algorithm Based on the Method of Optimal Set Partitioning

ELENA KISELEVA and TATYANA STEPANCHUK
*Department of Computational Mathematics and Mathematical Cybernetics, Dnepropetrovsk
National University, Naukova st. 13, Dnepropetrovsk, 49050, Ukraine (e-mail:
ekiseleva@hotmail.com, tstepanchuk@yahoo.com)*

Abstract. The examined algorithm for global optimization of the multiextremal non-differentiable function is based on the following idea: the problem of determination of the global minimum point of the function $f(x)$ on the set Ω ($f(x)$ has a finite number of local minima in this domain) is reduced to the problem of finding all local minima and their attraction spheres with a consequent choice of the global minimum point among them. This reduction is made by application of **the optimal set partitioning method**. The proposed algorithm is evaluated on a set of well-known one-dimensional, two-dimensional and three-dimensional test functions. Recommendations for choosing the algorithm parameters are given.

Key words: global minimum, non-differentiable optimization, optimal set partitioning

1. Introduction

Because of the variety of practical global optimization problems, it is expedient to apply specific methods of global search that most completely take into account characteristics of a particular class of problems. This fact follows from the numerous publications, for example Batishev (1975), Dem'yanov and Vasil'ev (1981), Strongin (1978), Suharev (1981, 1989), Ziglyavskiy (1985), Ziglyavskiy and Zilinskas (1991), Zilinskas (1986), Zilinskas and Shaltyanis (1989), reflecting the modern state of theory and methodology of global optimization.

Due to such variety, there is no uniform standard classification of global methods. Each of known classifications has its advantages and its drawbacks. However, in each classification that claims its own completeness, the approaches such as "multistart" Ziglyavskiy and Zilinskas (1991) or such as covering methods Suharev (1989) are discussed. These approaches are reduced to the estimation of the attraction spheres of local minima and to the choice of initial points for consecutive or parallel local descent to the local minimum points with the subsequent choice of the global minimum point among them. Let us examine these approaches explicitly.

Let us formulate the global optimization problem following Strongin (1978).

Let $f(x)$ be a real, multiextremal, continuous function defined on the domain Ω in an n-dimensional Euclidean space E_n, where the number of local minima is finite and does not exceed N.

Let us consider the problem of searching for the point $x^* \in \Omega$ (assumed to exist) such that

$$f(x^*) = \min_{x \in \Omega} f(x). \tag{1}$$

If such a point x^* exists in domain Ω and in a certain vicinity $U(x^*)$

$$f(x^*) \leqslant f(x), x \in \Omega \cap U(x^*),$$

then function $f(x)$ is called *unimodal* in $U(x^*)$.

If there are several points $\tau_i^*, 1 \leqslant i \leqslant N$, in the domain of definition Ω and each of them has its own vicinity $U(\tau_i^*)$ such that

$$f(\tau_i^*) \leqslant f(x), \quad x \in \Omega \cap U(\tau_i^*), \tag{2}$$

then function $f(x)$ is called *multiextremal*.

The points from (2) are called *local minimum points*, and the point τ^* is a *global minimum*.

Let $f(x)$ be the unimodal function on a certain subset $\Omega_i, 1 \leqslant i \leqslant N$, of the domain Ω (the local minimum point of the function $f(x)$ on the subset Ω_i is denoted by τ_i^*), and let

$$\bigcup_{i=1}^{N} \Omega_i = \Omega.$$

Then, by applying the known local descent methods, for an arbitrary initial point $\tau_i^0 \in \Omega_i$, we can obtain the corresponding local minimum point $\tau_i^* \in \Omega_i$. It is said that the subset Ω_i is an *attraction sphere of the local minimum* τ_i^*. In other words, the *local minimum attraction sphere* is a sphere in which the steepest descent starting from any point of this sphere leads to this local minimum.

Thus, the global optimization problem (1) is solved once the domain of definition Ω is partitioned into attraction spheres Ω_i of the local minima $\tau_i^*, 1 \leqslant i \leqslant N$.

However, as many authors indicate, for example Batishev (1975), Strongin (1978), Ziglyavskiy and Zilinskas (1991), Zilinskas (1986), the evaluation for the partitioning the domain of definition Ω into attraction spheres $\Omega_i, 1 \leqslant i \leqslant N$ is a complex problem.

Besides, in the general case, obtaining the evaluation for the number N of attraction spheres is a hard problem. Thus, the development of the objective function models and methods that include the number of local minima with characteristics of their attraction spheres is still an issue, Ziglyavskiy and Zilinskas (1991).

In this paper we examine the global optimization algorithm based on applying the optimal set partitioning method from Kiseleva (1989), Kiseleva and Shor (1994) that overcomes some of the described difficulties and problems.

More than 120 publications by E.M. Kiseleva and her research laboratory are devoted to the presentation the optimal set partitioning method and examination the efficiency of the algorithms that realized this method as well as its different modifications and generalizations for various classes of problems. All of these problems are reduced to the problem of optimal set partitioning.

Some of these publications are given in references Kiseleva (1985, 1989, 1991, 1992), Kiseleva and Shor (1991, 1994), Kiseleva and Mironenko (1996). In the works of Kiseleva and her co-authors the possibility of application of the optimal set partitioning method is shown for: (a) solving the oneproduct or multiproduct infinite-dimensional transport problems and allocation problems as well as the generalized Neyman-Pearson problem of testing the statistical hypotheses; (b) the problems of getting the Dirihle-Voronov diagrams; (c) the services sector territorial planning problems; (d) the problems of determination of the nodes in the optimal quadrature formulae for numerical integration, Suharev (1989); (e) the problems of optimal covering the set by spheres of equal radius, Suharev (1989); (f) solving the problems of a choice of the group decision in sociology etc.

The present work is the continuation of the E.M. Kiseleva, N.Z. Shor and their students works cycle on examination of the possibilities of the optimal set partitioning methods application and also N.Z. Shor's r-algorithm for various classes of theoretical and practical optimization problems.

This method not only allows to find all local minimum attraction spheres but also it simultaneously yields the coordinates of all local minima and the global minimum among them. Besides, in the case when the true number N_{true} of the local minima is not known in advance, it is enough to set only an expected number of local minima N_{exp}. The true number of local minima N_{true} is adjusted by the algorithm.

At the same time, the class of the minimized functions considered by the method is broad. It is a class of non-differentiable functions that allow application of the local search non-smooth optimization method – N.Z. Shor's r-algorithm, Shor (1985), i.e., a class of almost differentiable functions.

Thus, the examined algorithm from Kiseleva (1985), Kiseleva and Mironenko (1996) for global optimization of the multiextremal, non-differentiable function having at most N local minima is based on the following idea. We transform the problem of searching for the global minimum of the function $f(x)$ on the set Ω into the problem of determination of the optimal partitioning $\Omega_1^*, \ldots, \Omega_N^*$ of the admissible domain Ω into N local minima $\tau_1^*, \ldots, \tau_N^*$ attraction spheres and the coordinates of these sphere centers, simultaneously. These center coordinates correspond to the local minimum points of the original function $f(x)$ on Ω. As the optimality of the domain Ω partitioning into local minimum attraction spheres we

take the minimum of the error, which is admitted by referring the point x of the one local minimum attraction sphere to the attraction sphere of another local minimum.

One of the versions of the local search method of the generalized gradient descent with the space expansion in the direction of the difference of two sequential generalized gradient vectors (so-called N.Z. Shor's r-algorithm) Shor (1985) is used for solving the problem of finite-dimensional non-differentiable optimization, which consists of searching for the optimal local minimum coordinates. The proposed algorithm is evaluated on a set of well-known test problems for one-dimensional, two-dimensional and three-dimensional cases. Recommendations for choosing the algorithm parameters are given.

Let us consider the continuous problem of optimal partitioning of the set Ω from an n-dimensional Euclidean space E_n into $\{\Omega_1, \ldots, \Omega_N\}$ subsets with determination of the coordinates of these subset centers, Kiseleva (1989), Kiseleva and Shor (1994).

Let Ω be a bounded Lebesgue-measurable set in E_n. It is required to find *such a partition* of the set Ω into N Lebesgue-measurable subsets $\Omega_1^*, \ldots, \Omega_N^*$ (some of them may be empty) and *such coordinates* $\tau_1^*, \ldots, \tau_N^*$, (unknown in advance) of $\Omega_1^*, \ldots, \Omega_N^*$ subset centers, that represent the solution of the following optimization problem.

PROBLEM A. (see Figure 1).

$$\min_{\{\Omega_1, \ldots, \Omega_N\},\{\tau_1, \ldots, \tau_N\}} \quad \sum_{i=1}^{N} \int_{\Omega_i} c(x, \tau_i)\, \mathrm{d}x$$

$$\text{s.t.} \quad \bigcup_{i=1}^{N} \Omega_i = \Omega,$$

$$\mathrm{mes}(\Omega_i \bigcap \Omega_j) = 0, \ i \neq j, \ i, j = 1, 2, \ldots, N,$$

where $\mathrm{mes}(\cdot)$ – is the Lebesgue measure.

Here and henceforth all integrals are understood to be Lebesgue integrals. We shall assume that the set of boundary points Ω_i, $1 \leqslant i \leqslant N$, is of measure zero.

Functions $c(x, \tau_i)$ are real bounded measurable on x with any fixed $\tau_i = (\tau_i^{(1)}, \ldots, \tau_i^{(n)})$ from Ω, functions defined on $\Omega \times \Omega$.

2. The Reduction of the Global Optimization Problem to the Problem of Optimal Set Partitioning

Let us consider the following problem. It is required to find a point $x^* = (x^{*(1)}, \ldots, x^{*(n)})$, such that the function $f(x)$ achieves its minimal value on $\Omega \subset E_n$, i.e.

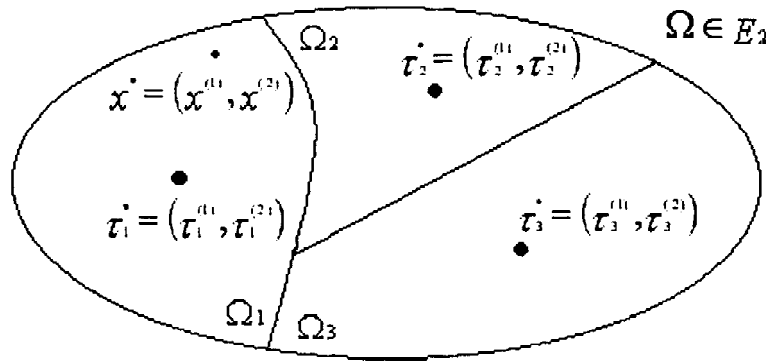$$f(x^*) = \min_{x \in \Omega} f(x), \tag{3}$$

Figure 1. The optimal partitioning of the set into three subsets.

where $f(x)$ is non-differentiable function on $\Omega \subset E_n$.

As the non-differentiable functions $f(x)$, according to Dem'yanov and Vasil'ev (1981), Mihalevich, Trubin and Shor (1986), Shor (1985), Shor (1998), we take the class of almost differentiable functions on $\Omega \subset E_n$.

The almost differentiable function in Mihalevich, Trubin and Shor (1986), Shor (1985), Shor (1998) is understood to be a continuous function $f(x)$ defined on E_n such that it is Lipschitz-locally, see Dem'yanov and Vasil'ev (1981), and continuously-differentiable on the set, where the gradient vector exists.

Let us assume that function $f(x)$ is unimodal on a certain subset Ω_i, $1 \leqslant i \leqslant N$, of the domain Ω (the point of the local minimum of the function $f(x)$ on Ω is denoted by τ_i^*), and let

$$\bigcup_{i=1}^{N} \Omega_i = \Omega,$$

$$\Omega_i \cap \Omega_j = 0, i \neq j, i, j = 1, 2, \dots, N. \tag{4}$$

We reduce the problem (3) of searching for the global minimum of the function $f(x)$ on Ω to the optimal set partitioning **Problem A** from section 1.

For the reduction of the problem (3) to the *Problem A* it is necessary to fulfill the following conditions:

(a) the optimal partitioning $\Omega_1^*, \dots, \Omega_N^*$ of the set Ω of the *problem A* must be the partitioning of the admissible domain of definition Ω of the function $f(x)$ into its local minima attraction spheres;

(b) the optimal coordinates of the centers $\tau_1^*, \dots, \tau_N^*$ of the subsets $\Omega_1^*, \dots, \Omega_N^*$, accordingly, must coincide with the local minima coordinates of the function $f(x)$ on Ω.

For simultaneous fulfillment of both of the described above demands the object in **Problem A** must be the minimization of the functional of the total losses of incorrect partitioning of the set Ω into attraction spheres $\Omega_1^*, \dots, \Omega_N^*$ and incorrect
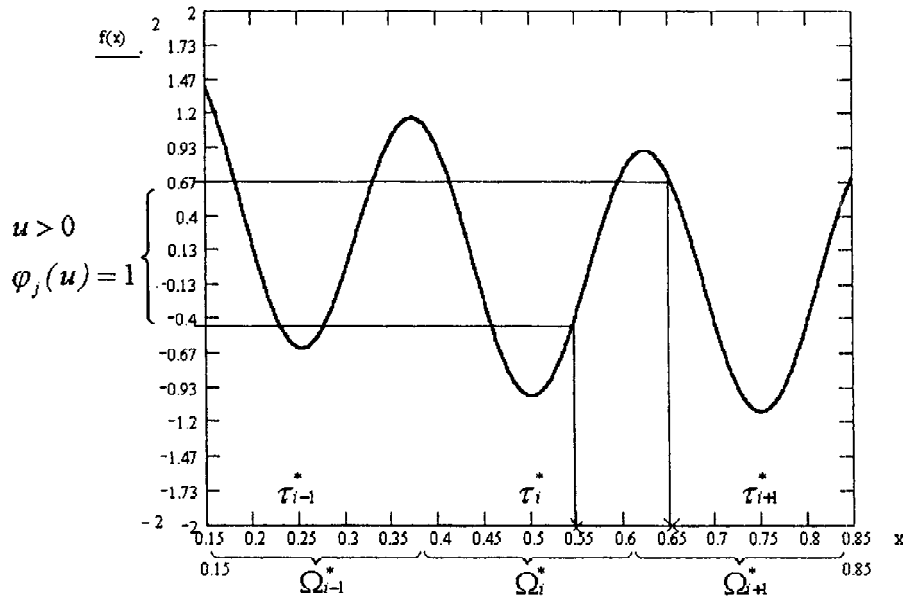
Figure 2. Plot of the minimized function $f_9(x)$.

determination of the local minima $\tau_1^*, \ldots, \tau_N^*$ at each of these spheres, that is the functional

$$\sum_{i=1}^{N} \int_{\Omega_i} c(x, \tau_i) \, dx,$$

where the functions $c(x, \tau_i)$, $i = 1, 2, \ldots, N$, must penalize the referral of a point x to the incorrect attraction sphere.

Let us define the penalty function $c(x, \tau_i)$ as follows:

$$c(x, \tau_i) = f(\tau_i) + S \sum_{j=1}^{k} \varphi_j(u), \tag{5}$$

where

$$\varphi_j(u) = \begin{cases} 0, & u \leqslant 0 \\ 1, & u > 0 \end{cases}, u = f\left(x + \frac{\tau_i - x}{k} j\right) - f\left(x + \frac{\tau_i - x}{k}(j - 1)\right),$$

S is a sufficiently large positive number; k is a natural number, which indexes the segment $[x, \tau_i]$.

As seen from Figure 2 if the point x belongs to the ith local minimum attraction sphere then $\varphi_j(u) = 0$. If the point x belongs to $(i + 1)$st local minimum attraction sphere, but it was referred to the ith local minimum attraction sphere, then $u > 0$ and $\varphi_j(u) = 1$. When the point x is at the larger distance from the point τ_i under an incorrect reference then the penalty value $\sum_{j=1}^{k} \varphi_j(u)$ is greater.

The partitioning of the set Ω into Ω_{i-1}^0, Ω_i^0, Ω_{i+1}^0 attraction spheres determined by the initial approximation of local minimum coordinates $\tau_{i-1}^0 = 0.2$, $\tau_i^0 = 0.6$, $\tau_{i+1}^0 = 0.65$ of the function plotted in Figure 2, and the graphs of the penalty functions $c(x, \tau_{i-1}^0)$, $c(x, \tau_i^0)$, and $c(x, \tau_{i+1}^0)$ are presented in Figure 3b.

Here the typical feature is the unimodality of each of the functions $c(x, \tau_{i-1}^0)$, $c(x, \tau_i^0)$ and $c(x, \tau_{i+1}^0)$ at the points of the corresponding attraction spheres Ω_{i-1}^0, Ω_i^0, Ω_{i+1}^0.

The optimal partitioning of a set Ω into Ω_{i-1}^*, Ω_i^*, Ω_{i+1}^* attraction spheres of the local minima τ_{i-1}^*, τ_i^*, τ_{i+1}^* of the function from Figure 2 is presented in Figure 3a.

It can be seen that the values of function $c(x, \tau_i^*)$ at the points of the local minimum $\tau_i^* = 0.5$ attraction sphere coincides with the value of the function $f(x)$ at point $x = 0.5$. Outside this sphere the value of the function $c(x, \tau_i^*)$ is abruptly increased (The rate of increase is determined by the parameter S from (5)). The same conclusion can be made for the functions $c(x, \tau_{i-1}^*)$, $c(x, \tau_{i+1}^*)$.

Thus, we transformed the problem (3) into one of searching for the local minimum coordinates and their attraction spheres which is reduced to the problem of optimal partitioning of the set Ω into subsets $\Omega_1, \ldots, \Omega_N$ with the determination of the coordinates $\tau_1, \ldots, \tau_N$ of these subset centers:

$$\min_{\{\Omega_1, \ldots, \Omega_N\}, \{\tau_1, \ldots, \tau_N\}} \sum_{i=1}^N \int_{\Omega_i} c(x, \tau_i) \, dx \tag{6}$$

$$\text{s.t.} \quad \bigcup_{i=1}^N \Omega_i = \Omega, \tag{7}$$

$$\text{mes}(\Omega_i \cap \Omega_j) = 0, i \neq j, i, j = 1, 2, \ldots, N, \tag{8}$$

where the penalty functions $c(x, \tau_i)$ have the form (5).

3. On the Method of Optimal Partitioning of Set Ω from E_n into $\{\Omega_1, \ldots, \Omega_N\}$ Subsets with Determination of the Coordinates of the Subset Centers

Let us describe the method for solving the optimal set partitioning **Problem A** from section 1 of this paper.

For each subset Ω_i, we introduce the characteristic function

$$\lambda_i(x) = \begin{cases} 1, & x \in \Omega_i, \\ 0, & x \in \Omega \backslash \Omega_i, i = 1, 2, \ldots, N. \end{cases}$$
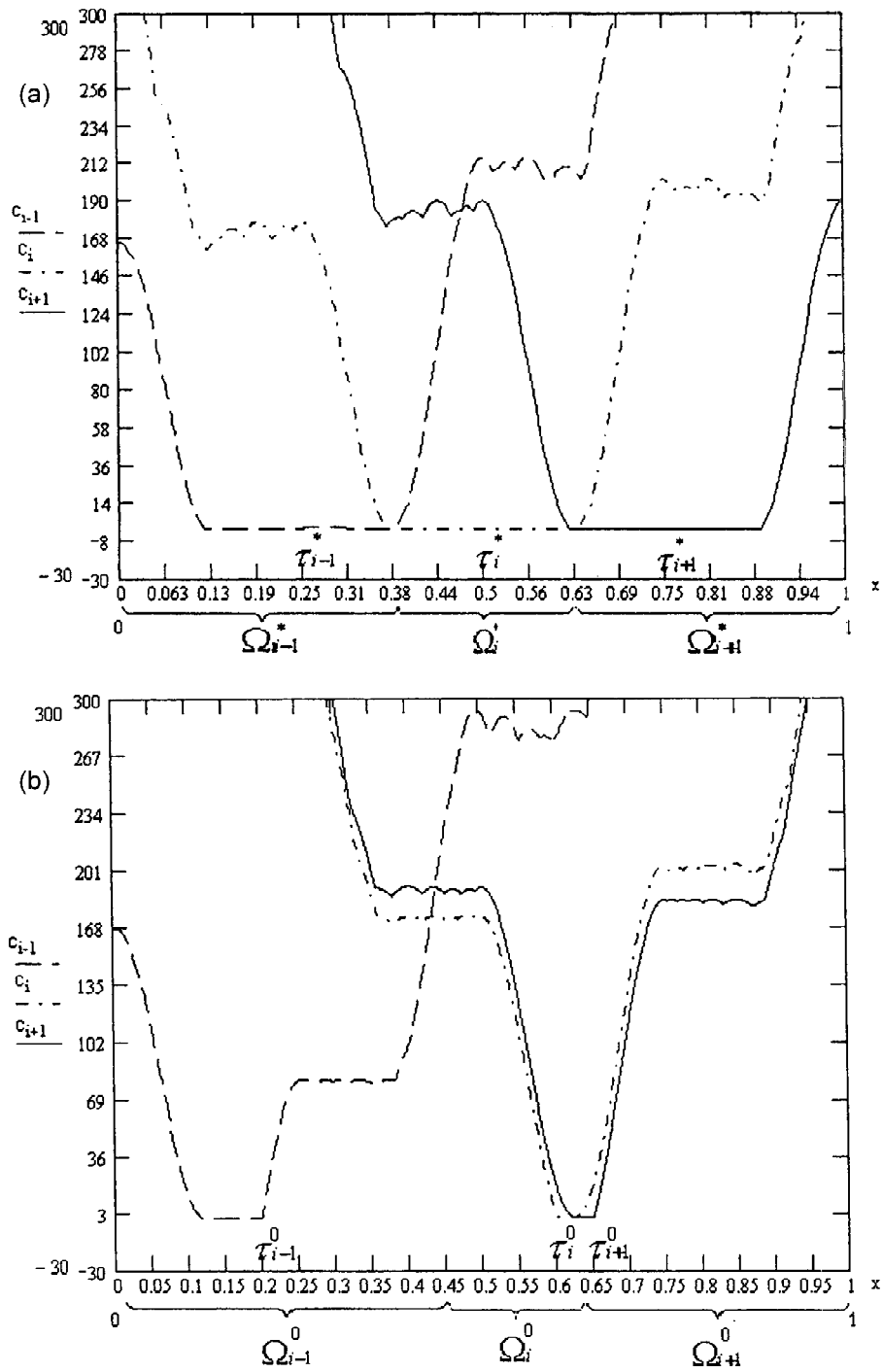
Figure 3. (a) Plot of the penalty functions of minimizing the function $f_9(x)$ at the optimal solution: $N_{\text{true}} = 3$. (b) Plot of the penalty functions of minimizing the $f_9(x)$ at the initial solution: $N_{\text{exp.}} = 3$.

We can rewrite the *Problem A* as follows:

Problem B.

$$\min_{(\lambda(\cdot),\tau)\in\Gamma\times\Omega^N} \int_\Omega \sum_{i=1}^N c(x,\tau_i)\lambda_i(x)\,\mathrm{d}x,$$

where

$$\Gamma = \left\{ \begin{array}{c} \lambda(x) = (\lambda_1(x),\dots,\lambda_N(x)) : \lambda_i(x) = 0 \vee 1 \\ \text{almost everywhere (a.e.) for } x\in\Omega, i=1,2,\dots,N, \\ \sum_{i=1}^N \lambda_i(x) = 1 \text{ a.e. for } x\in\Omega, \end{array} \right\},$$

$$\tau = (\tau_1,\dots,\tau_N) \in \underbrace{\Omega\times\cdots\times\Omega}_{N} = \Omega^N.$$

(We recall that $(\tau_i = (\tau_i^{(1)},\dots,\tau_i^{(n)})$ – the points of the subset Ω_i, $1\leqslant i\leqslant N$).

Denote

$$I(\lambda(\cdot),\tau) = \sum_{i=1}^N \int_\Omega c(x,\tau_i)\lambda_i(x)\,\mathrm{d}x.$$

Evidently,

$$I(\lambda^*(\cdot),\tau^*) = \inf_{(\lambda,\tau)\in\Gamma\times\Omega^N} I(\lambda(\cdot),\tau)) =$$
$$= \inf_{\tau\in\Omega^N} (\inf_{\lambda(\cdot)\in\Gamma} I(\lambda(\cdot),\tau)). \tag{9}$$

At first let us find the optimal solution of the inner problem from (9) with any fixed $\tau\in\Omega^N$.

It follows from Kiseleva (1989) that there is at least one simplex point among the set of points in which the functional $I(\lambda,\tau)$, that is linear over $\lambda(\cdot)$, attains its minimal value for the $\lambda(\cdot)$ on Γ with any fixed $\tau\in\Omega^N$ and

$$\Gamma_1 = \left\{ \begin{array}{c} \lambda(x) = (\lambda_1(x),\dots,\lambda_i(x),\dots,\lambda_N(x)) : \\ 0\leqslant\lambda_i(x)\leqslant 1, i=1,\dots,N, \sum_{i=1}^N \lambda_i(x) = 1 \end{array} \right\}.$$

The extreme points of simplex Γ_1 are the characteristic functions of some subsets Ω_i, that form a partitioning of a set Ω with any fixed $\tau\in\Omega^N$.

Thus, we can show by analogy with Kiseleva (1989), that the optimal solution of the inner problem from (9) is attained with any fixed $\tau\in\Omega^N$ for the vector-valued function $\lambda^*(x) = (\lambda_1^*(x),\dots,\lambda_N^*(x)) : 0\leqslant\lambda_i(x)\leqslant 1, i=1,\dots,N$, the ith component of which has the form

$$\lambda_i^*(x) = \begin{cases} 1, & \text{if } c(x,\tau_i^*) = \min_{k=1,\dots,N} c(x,\tau_k^*), \text{ then } x\in\Omega_i^*, \\ 0, & \text{then } x\notin\Omega_i^*. \end{cases} \tag{10}$$

Let us pass to the specific definition of the outer problem from (9). Denote

$$G(\tau) = \min_{\lambda(\cdot) \in \Gamma} I(\lambda(\cdot), \tau), \tau \in \Omega^N. \tag{11}$$

It is proved in works Kiseleva (1989), Kiseleva and Shor (1994) that for vector $\tau^* = (\tau_1^*, \ldots, \tau_N^*)$ from formulae (10) it is necessary to choose the optimal solution of the outer problem from (9) that transforms it into the following form:

$$G(\tau) = \inf_{\tau \in \Omega^N} \int_{\Omega} \min_{i=1,2,\ldots,N} c(x, \tau_i) \, dx. \tag{12}$$

So, to sum up, notice that the analytical expression for the first component $\lambda^*(\cdot)$ of the optimal solution $(\lambda^*(\cdot), \tau^*)$ of the *Problem B* is obtained in the form (10) with any fixed τ as an inner problem solution of infinite-dimensional optimization from (9).

It should be observed that if $c(x, \tau_i)$ is a Euclidean metric, then the optimal solution of the inner problem from (9), that determined for any fixed $\tau \in \Omega^N$, of the vector-valued function $\lambda^*(x) = (\lambda_1^*(x), \ldots, \lambda_N^*(x))$ by the formulae (10), turns out to be the well-known Dirihle-Voronogo partitioning, Suharev (1989).

For finding the second component $\tau^* = (\tau_1^*, \ldots, \tau_N^*)$ of the optimal solution of the *Problem B* it is necessary to solve the finite-dimensional problem (12). To solve this problem we apply one of the versions of the method of the generalized gradient descent with the space expansion in the direction of the difference of two sequential generalized gradient vectors (so-called N.Z. Shor's r-algorithm). This algorithm is used, in general, for the determination of the local minima of the non-differentiable multiextremal objective function $G(\tau)$ from (12).

We can find the detailed description of the different r-algorithm versions in works by Mihalevich et al. (1986), and Shor (1985, 1998).

The essence of the methods of generalized gradient descent with the space expansion is based on construction of sequential linear operator approximations that change the space metrics, and the choice of the descent direction corresponding to the antigradient from a transformed space with a new metrics.

The r-algorithm iterated formula has the form Shor (1985):

$$\tau^{k+1} = \tau^k - h_k B_{k+1}^\tau [B_{k+1}^\tau]^T g_G(\tau^k), k = 0, 1, 2 \ldots, \tag{13}$$

Here B_{k+1}^τ is the operator from the transformed space to the base space E_N (such that $B_0^\tau = I_N$ is the identity matrix), h_k is the step multiplier, which can be determined from the minimum condition of function G on the direction $-B_{k+1}B_{k+1}^T g_G(\tau^k)$, where $g_G(\tau^k)$ is the generalized gradient vector of the function $G(\tau)$ at the point τ^k.

In the present paper N.Z. Shor's r-algorithm is utilized in the H-form (Mihalevich et al. 1986) (H_k is the symmetric matrix such that $H_k = B_k B_k^T$), so the iterated formulae (13) take the form:

$$\tau^{k+1} = \tau^k - h_k \frac{H_{k+1} g_G(\tau^k)}{\sqrt{(H_{k+1} g_G(\tau^k), g_G(\tau^k))}}, k = 0, 1, 2 \ldots, \tag{14}$$

where

$$H_{k+1} = H_k + \left(\frac{1}{\alpha_k^2} - 1\right) \frac{H_k \Delta_k \Delta_k^T H_k}{(H_k \Delta_k, \Delta_k)},$$

$$\Delta_k = g_G(\tau^k) - g_G(\tau^{k-1}).$$

The coefficient of the space expansion $\alpha_k = 3$. Here we apply the adaptive control method for the step multiplier h_k described in Mihalevich et al. (1986) and Shor (1985).

4. The Algorithm for Solving the Global Optimization Problem that is Reduced to the Problem of Optimal Set Partitioning

Let us describe the algorithm for solving the global optimization problem (3) that is reduced to the problem of optimal set partitioning (6)–(8) based on the method from Section 3 of this paper.

Taking into account that x, τ_i from (6)–(8) are the points of n-dimensional Euclidean space, i.e., $x = (x^{(1)}, \ldots, x^{(n)})$, $\tau_i = (\tau_i^{(1)}, \ldots, \tau_i^{(n)})$ we can rewrite the function $c(x, \tau_i)$ in a form that is convenient for further examinations as follows:

$$c(x^{(1)}, \ldots, x^{(n)}, \tau_i^{(1)}, \ldots, \tau_i^{(n)}) = f(\tau_i^{(1)}, \ldots, \tau_i^{(n)}) +$$

$$+ S \sum_{j=1}^{k} \max \left\{ \begin{array}{l} 0, f\left(x^{(1)} + \frac{\tau_i^{(1)} - x^{(1)}}{k} j, \ldots, x^{(n)} + \frac{\tau_i^{(n)} - x^{(n)}}{k} j\right) - \\ -f\left(x^{(1)} + \frac{\tau_i^{(1)} - x^{(1)}}{k}(j-1), \ldots, x^{(n)} + \frac{\tau_i^{(n)} - x^{(n)}}{k}(j-1)\right) \end{array} \right\}$$

$$(15)$$

To solve the problem (6)–(8), (15) we have to concretize the generalized gradient vector $g_G^\tau(\tau)$ from (13) of the function $G(\tau)$ from (12) at the point

$$\tau = (\tau_1^{(1)}, \ldots, \tau_1^{(n)}; \ldots; \tau_i^{(1)}, \ldots, \tau_i^{(p)}, \ldots, \tau_i^{(n)}; \ldots; \tau_N^{(1)}, \ldots, \tau_N^{(n)}).$$

Note that the generalized gradient vector of the convex function coincides with a subgradient vector. Otherwise the almost gradient vector is taken as a generalized gradient vector (see Mihalevich et al., 1986; Shor, 1985, 1998).

The almost gradient vector coincides with the gradient vector at the point when the function $f(x)$ is differentiable. The almost gradient vector coincides with the gradient vector at one of the adjoining to this point pieces in the case when the function $f(x)$ is a piecewise smooth (Mihalevich et al. 1986).

Thus, in most of the practical cases, there is no difficulty in the computation of the generalized gradient vector.

Let us define the *i*th component of the generalized gradient vector $g_G^\tau(\tau) = (g_G^{\tau_1}(\tau), \dots, g_G^{\tau_i}(\tau), \dots, g_G^{\tau_N}(\tau))$ of the function

$$G(\tau_1^{(1)}, ..., \tau_1^{(n)}; \dots; \tau_i^{(1)}, ..., \tau_i^{(p)}, ..., \tau_i^{(n)}; \dots; \tau_N^{(1)}, ..., \tau_N^{(n)}) =$$

$$= \int_\Omega \min_{i=1,2,...,N} c(x^{(1)}, ..., x^{(n)}; \dots; \tau_i^{(1)}, ..., \tau_i^{(n)}) \, dx^{(1)}, ..., dx^{(n)} \qquad (16)$$

at the point $\tau = (\tau_1^{(1)}, ..., \tau_N^{(n)})$ as follows:

$$g_G^{\tau_i}(\tau) = g_G^{\tau_i}(\tau_1^{(1)}, ..., \tau_N^{(n)}) =$$

$$= \int_\Omega g_c^{\tau_i}(x^{(1)}, ..., x^{(n)}; \dots; \tau_1^{(1)}, ..., \tau_N^{(n)}) \lambda_i(x^{(1)}, ..., x^{(n)}) \, dx^{(1)}, ..., dx^{(n)},$$

$$(17)$$

where $g_c^{\tau_i}(x, \tau)$ is the *i*th component of the *n*-dimensional generalized gradient vector $g_c^\tau(x, \tau)$ of the function $c(x, \tau_i)$ at the point $\tau = (\tau_1, ..., \tau_i, ..., \tau_N)$ (here $\tau_i = (\tau_i^{(1)}, ..., \tau_i^{(n)})$) with any fixed x has the form:

$$g_c^{\tau_i}(x^{(1)}, ..., x^{(n)}; \tau_1^{(1)}, ..., \tau_1^{(n)}; \tau_i^{(1)}, ..., \tau_i^{(p)}, ..., \tau_i^{(n)}; \tau_N^{(1)}, ..., \tau_N^{(n)}) =$$

$$= \begin{pmatrix} g_c^{\tau_i^{(1)}}(x^{(1)}, ..., x^{(n)}; \tau_1^{(1)}, ..., \tau_N^{(n)}) \\ \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \\ \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \\ \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \\ g_c^{\tau_i^{(p)}}(x^{(1)}, ..., x^{(n)}; \tau_1^{(1)}, ..., \tau_N^{(n)}) \\ \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \\ \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \\ \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \\ g_c^{\tau_i^{(n)}}(x^{(1)}, ..., x^{(n)}; \tau_1^{(1)}, ..., \tau_N^{(n)}) \end{pmatrix}$$

The substantiation of Equation (17) can be found in Kiseleva (1989).

In turn, the *p*th component $g_c^{\tau_i^{(p)}}(x, \tau)$ of the generalized gradient vector of the function $c(x, \tau_i)$ at the point $\tau = (\tau_1^{(1)}, ..., \tau_N^{(n)})$ with any fixed $x = (x^{(1)}, ..., x^{(p)}, ..., x^{(n)})$ is defined for all $i = 1, ..., N$, and $p = 1, \dots, n$, by formulae:

$$g_c^{\tau_i^{(p)}}(x^{(1)}, ..., x^{(n)}; \dots; \tau_1^{(1)}, ..., \tau_1^{(n)}; \dots; \tau_i^{(1)}, ..., \tau_i^{(p)}, ..., \tau_i^{(n)}; \dots; \tau_N^{(1)}, ...,$$

$$\tau_N^{(n)}) = g_f^{\tau_i^{(p)}}(\tau_1^{(1)}, ..., \tau_i^{(p)}, ..., \tau_N^{(n)}) + \qquad (18)$$

$$+ S \begin{cases} 0, \text{ if } f(u^{(1)}, ..., u^{(n)}) - f(v^{(1)}, ..., v^{(n)}) \leqslant 0 \\ \sum_{j=1}^{k} \left(\begin{array}{c} g_f^{u^{(p)}}(u^{(1)}, ..., u^{(p)}, ..., u^{(n)}) \dfrac{j}{k} - \\ -g_f^{v^{(p)}}(v^{(1)}, ..., v^{(p)}, ..., v^{(n)}) \dfrac{j-1}{k} \end{array} \right), \\ \text{if } f(u^{(1)}, ..., u^{(n)}) - f(v^{(1)}, ..., v^{(n)}) > 0, \end{cases}$$

where

$$u = (u^{(1)}, ..., u^{(p)}, ..., u^{(n)}) =$$
$$\left(x^{(1)} + \frac{\tau_i^{(1)} - x^{(1)}}{k}j, ..., x^{(p)} + \frac{\tau_i^{(p)} - x^{(p)}}{k}j, ..., x^{(n)} + \frac{\tau_i^{(n)} - x^{(n)}}{k}j \right);$$

$$v = (v^{(1)}, ..., v^{(p)}, ..., v^{(n)}) =$$
$$\left(x^{(1)} + \frac{\tau_i^{(1)} - x^{(1)}}{k}(j-1), ..., x^{(p)} + \frac{\tau_i^{(p)} - x^{(p)}}{k}(j-1), ..., \right.$$
$$\left. x^{(n)} + \frac{\tau_i^{(n)} - x^{(n)}}{k}(j-1) \right).$$

For simplicity, let us describe the general scheme of the algorithm for one-dimensional global optimization. The multi-dimensional case is substantially the same. It differs basically just by replacement the one-dimensional integrals with the multi-dimensional and by the specific definition of the formulae (5), (15) that compute the penalty function $c(x, \tau_i)$.

5. Algorithm for Searching for the Local Minima of the Function $f(x)$ on [a,b] and their Attraction Spheres

Step 0 (Initialization). Specify the values for parameters N and M; we cover the segment $[a, b]$ with a one-dimensional mesh $x_i = a + (i - 1)H_x, i = 1, ..., M$, where H_x is a step of the mesh, $H_x = b - a/M - 1$; we specify the initial approximation $\tau^0 = (\tau_1^0, ..., \tau_N^0)$; we use the formulae (10) with $\tau = \tau^0$ to compute the vector-valued function $\lambda^0(x) = (\lambda_1^0(x), ..., \lambda_N^0(x))$ at the nodes of the mesh, that characterize the initial partitioning of the segment $[a, b]$ into attraction spheres $\Omega_1^0, ..., \Omega_N^0$ of the local minima $\tau_1^0, ..., \tau_N^0$, (see Figure 3a); we use the formulae (17),(18) with $\lambda(x) = \lambda^0(x)$, $\tau = \tau^0$, $n = 1$, $p = 1$ to compute the values of the generalized gradient vector $g_G^\tau(\tau) = (g_G^{\tau_1}(\tau), ..., g_G^{\tau_N}(\tau))$; we choose the initial sample step $h_0 > 0$ for N.Z. Shor's r-algorithm; set $k = 0$.
Step 1. We compute $\lambda^k(x)$ at the nodes of the mesh using formulae (10) with $\tau = \tau^k$.
Step 2. We compute $g_G^{\tau_i}(\tau), i = 1, ... , N$ at the nodes of the mesh using formulae (17),(18) with $\lambda(x) = \lambda^k(x)$, $\tau = \tau^k$.
Step 3. For minimizing the function $G(\tau)$ from (16) on $\tau = (\tau_1, ... , \tau_N)$ from Ω^N we perform the $(k + 1)$-st step of N.Z. Shor's r-algorithm in H-form (Mihalevich et al., 1986), whose short scheme has the form:

$$\tau^{k+1} = P_{[a,b]} \left(\tau^k - h_k \frac{H_{k+1}g_G(\tau^k)}{\sqrt{(H_{k+1}g_G(\tau^k), g_G(\tau^k))}} \right),$$

where $P_{[a,b]}$ is the operator of projection on $[a, b]$.
Step 4. If the condition

$$\left\| \tau^k - \tau^{k+1} \right\| \leqslant \varepsilon, \varepsilon > 0 \tag{19}$$

is not satisfied, then we set $k = k+1$ and pass to **step 1** of the algorithm. Otherwise we pass to **step 5**.
Step 5. We set $\tau^* = \tau^p, \lambda^*(x) = \lambda^p(x)$, where p is the number of the iterations for which the condition (19) has been satisfied. Thus we obtain the vector of local minimum coordinates $\tau^* = (\tau_1^*, ..., \tau_N^*)$ and vector-valued function $\lambda^*(x) = (\lambda_1^*(x), ..., \lambda_N^*(x))$ that characterize the partitioning of the segment [a,b] into attraction spheres $\Omega_1^*, ..., \Omega_N^*$ of these local minima.
Step 6. We compute the value of the original function at the local minimum points $\tau_1^*, ..., \tau_N^*$ and choose the point of the global minimum $x^* : f(x^*) = \min\{f(\tau_1^*), ..., f(\tau_N^*)\}$ among them.
    This completes the description of algorithm.

6. The Efficiency Estimate and the Properties Revelation of the Described Algorithm

To estimate the efficiency of the algorithm and to show its properties the algorithm is evaluated on a set of test problems in one-, two- and three-dimensional cases taken from Batishev (1981), Himmelblau (1975), Zilinskas (1989), Zilinskas and Shaltianis (1989), Strongin (1978), Suharev (1981), Floudas (2000).

One-dimensional case:
(1) Himmelblau (1975).

$$f_1(x) = (x^2 - 1)^2, \quad -2 \leqslant x \leqslant 2.$$

(2) Himmelblau (1975).

$$f_2(x) = x \cdot \sin x, \quad -7.85 \leqslant x \leqslant 7.85.$$

(3) Authors choice.

$$f_3(x) = |x| + |x - 1| - 1, \quad -2 \leqslant x \leqslant 2.$$

(4) Strongin (1978), Zilinskas (1986), Zilinskas and Shaltyanis (1991).

$$f_4(x) = \sin x + \sin(10x/3) + \ln x - 0.84x + 3, \quad 3 \leqslant x \leqslant 7.5.$$

(5) Strongin (1978).

$$f_5(x) = 2 - \cos x - \cos(2x), \quad -1.5 \leqslant x \leqslant 4.5.$$

(6) Suharev (1981).

$$f_6(x) = \sin(1/x), \quad 0 \leqslant x \leqslant 1.$$

(7) Himmelblau (1975).

$$f_7(x) = (1 - x)^2 \cdot (x + 1)^4 \cdot (x - 2)^3 \cdot x, \quad -2 \leqslant x \leqslant 2.$$

(8) Zilinskas (1986).

$$f_8(x) = \sin x + \sin(2x/3), \quad 3.1 \leqslant x \leqslant 20.4.$$

(9) Zilinskas (1986).

$$f_9(x) = 2(x - 0.75)^2 + \sin(8\pi x - \pi/2) - 0.125,$$
$$0 \leqslant x \leqslant 1, \quad -2 \leqslant \text{x} \leqslant 2.$$

(10) Suharev (1981), Zilinskas (1986).

$$f_{10}(x) = -\sum_{i=1}^{5} i \sin((i + 1)x + i), \quad -10 \leqslant x \leqslant 10.$$

(11) Zilinskas (1986), Zilinskas and Shaltyanis (1989).

$$f_{11}(x) = (x + \sin x) \cdot \exp(-x^2), \quad -10 \leqslant x \leqslant 10.$$

(12) Zilinskas (1986).

$$f_{12}(x) = \sin(x), \quad 0 \leqslant x \leqslant 50, \quad 0 \leqslant x \leqslant 100, \quad 0 \leqslant x \leqslant 20.$$

Two-dimensional case:
(13) Suharev (1981).

$$f_{13}(x, y) = x^4 + 4x^3 + 4x^2 + y^2, \quad -3 \leqslant x \leqslant 1, \quad -2 \leqslant y \leqslant 2.$$

(14) Himmelblau (1975).

$$f_{14}(x, y) = (x^2 + y - 11)^2 + (x + y^2 - 7)^2, \quad -2 \leqslant x \leqslant 4, \quad -2 \leqslant y \leqslant 4.$$

(15) Batyshev (1975), Himmelblau (1975), Suharev (1981).

$$f_{15}(x, y) = -\left(1 + 8x - 7x^2 + \frac{7}{3}x^3 - \frac{1}{4}x^4\right) \cdot y^2 \cdot e^{-y},$$
$$0 \leqslant x \leqslant 4.2, 0 \leqslant y \leqslant 6.4.$$

(16) Zilinskas (1986).

$$f_{16}(x, y) = a(y - bx^2 + cx - d)^2 + l(1 - f) \cdot \cos x + l,$$
$$-5 \leqslant x \leqslant 10, \quad 0 \leqslant y \leqslant 15,$$
$$a = 1 b = \frac{5.1}{4 \cdot 3.14159^2} c = \frac{5}{3.14159} d = 6 l = 10 f = \frac{1}{8 \cdot 3.14159}.$$

(17) Floudas (2000).

$$f_{17}(x, y) = \cos x \sin y - \frac{x}{y^2 + 1}, \quad -1 \leqslant x \leqslant 2, \quad -1 \leqslant y \leqslant 1.$$

Three-dimensional case:
(18) Himmelblau (1975).

$$f_{18}(x, y, z) = x^2 + y^2 + z^2, \quad x, y, z \in [-0.5; 0.5].$$

(19) Zilinskas (1986) ($n = 3, m = 4$).

$$f_{19}(x, y, z) = - e^{-3(x-0.3689)^2 - 10(y-0.117)^2 - 30(z-0.2673)^2} -$$
$$- 1.2 e^{-0.1(x-0.4699)^2 - 10(y-0.4387)^2 - 35(z-0.747)^2} -$$
$$- 3 e^{-3(x-0.1091)^2 - 10(y-0.8732)^2 - 30(z-0.5547)^2} -$$
$$- 3.2 e^{-0.1(x-0.03815)^2 - 10(y-0.5743)^2 - 35(z-0.8828)^2},$$
$$x, y, z \in [0; 1]$$

The results of the numerical experiments with one-dimensional case are given in Table 1, with two-dimensional case are given in Table 2, and with three-dimensional case are given in Table 3.

As seen from Table 1, to implement the algorithm, the user has to set the following parameters.

1. N: a true number of local minima of the function $f(x)$ on $[a, b]$ in the case when the user knows it. In the case when the N_{true} is unknown then we specify $N_{\text{exp.}}$ – an expected (by user) number of local minima ($N_{\text{exp.}}$ – an initial estimation of a number of local minima);
2. M: the number of nodes in the mesh, covering the segment $[a, b]$;
3. K: a natural number, which characterizing the number of subdivisions of the segment $[x, \tau_i]$, for computing the value of $c(x, \tau_i)$;
4. $\tau^0 = (\tau_1^0, ..., \tau_N^0)$: an initial approximation for the local minima of the function $f(x)$ on $[a, b]$;
5. ε: convergence tolerance of N.Z. Shor's r-algorithm for the calculation of the local minimum coordinates (see algorithm, step 4).

Table 1. The algorithm testing results: one-dimensional case

No.	The function, search domain	N_{exp}	M	K	Initial approximation τ^0	Local minima τ^*	The value of the function at the local minima	N_{true}	Global minima	Tolerance ε	Iterations I	ho	S	CPU time (s)
1.	f1(x) [−2,2]	3	2	3	(−2; 0; 2)	(−1,00; 1,00)	(0,00; 0,00)	2	1,00	0,001	8	1	100	0,05
		3	10	10	(0; 0; 0)	(−1,00; 1,00)	(0,00; 0,00)		1,00	0,00001	26	1	100	1,05
2.	f2(x) [−7,85; 7,85]	3	3	3	(−7,8; 0; 7,8)	(−4,9; 0; 4,9)	(−4,8; 0; −4,8)	3	−4,9	0,01	10	1	10	0,17
		3	7	3	(1; 1; 1)	(−4,9; 4,9; 0)	(−4,8; −4,8; 0)		−4,9	0,01	22	1	100	0,65
3.	f3(x) [−2;2]	3	2	2	(−2; −2; −2)	0,00	0,00	1	0,00	0,001	8	1	100	0,06
4.	f4(x) [3; 7,5]	4	3	4	(3; 4; 6; 7,5)	(3,44; 5,19; 7,07)	(0,16; −1,60; −1,27)	3	5,19	0,001	20	1	100	0,09
5.	f5(x) [−1,5; 4,5]	3	9	10	(−1,5; 2; 4,5)	(0,00; 3,14)	(0,00; 2,00)	2	0,00	0,001	16	1	100	0,12
6.	f6(X) [0,05; 1]	4	35	20	(0,05; 0,07; 0,12; 0,24)	(0,058; 0,091; 0,213)	(−0,999; −0,999; −0,999)	3	0,058	0,001	8	0,01	10	4,00
		4	35	20	(0,05; 0,07; 0,12; 0,24)	(0,058; 0,091; 0,213)	(−1; −0,9999; −0,9999)		0,0578	0,00001	20	0,01	10	6,00
		4	20	20	(0,05; 0,07; 0,12; 0,24)	(0,058; 0,091; 0,213)	(−0,998; −0,999; −0,999)		0,058	0,001	9	0,01	10	5,00
		4	20	12	(0,05; 0,07; 0,12; 0,24)	(0,058; 0,091; 0,213)	(−0,998; −0,999; −0,999)		0,058	0,001	9	0,01	10	5,00
7.	f7(x) [−2;2]	3	10	4	(−0,8; 0; 0)	(−1,00; 1,51; 0,41)	(0,00; −1,83; −2,27;)	3	0,41	0,001	18	1	100	0,22
8.	f8(x) [3; 20]	4	3	4	(3; 7; 11; 16)	(5,36; 10,45; 17,04)	(−1,22; −0,22; −1,91)	3	17,04	0,001	17	0,1	10	0,10
9a.	f9(x) [0; 1]	3	5	5	(0,2; 0,6; 0,65)	(0,25; 0,5; 0,75)	(−0,62; −1,00; −1,12)	3	0,75	0,001	14	0,1	10	0,04
		4	5	6	(0,2; 0,44; 0,6; 0,8)	(0,25; 0,50; 0,75)	(−0,63; −1,00; −1,12)		0,75	0,001	14	0,1	10	0,05
		4	5	30	(0,4; 0,4; 0,4; 0,4)	(0,25; 0,50; 0,75)	(−0,63; −1,00; −1,12)		0,75	0,001	18	0,1	10000	0,05
		5	5	6	(0,2; 0,3; 0,4; 0,6; 0,8)	(0,25; 0,50; 0,75)	(−0,63; −1,00; −1,12)		0,75	0,001	13	0,05	5	0,07
		2	4	5	(0,2; 0,8)	(0,24; 0,75)	(−0,60; −1,12)		0,75	0,001	10	0,05	5	0,04
9b.	[−2;2]	17	40	50	(−1,8; −1,6; −1,6; −1,4; −1,4; −0,7; −0,7; 0,4; 0,4; 0,4; 0,9; 0,9; 0,9; 1,4; 1,4; 1,4; 1,8)	(−1,73; −1,48; −1,23; −0,99; −0,74; 0,50; 0,25; 1,00; 0,75; 1,50; 1,75)	(11,3; 8,94; 6,83; 4,97; 3,34; −1,009; −0,63; −0,999; −1,12; −0,005; 0,86)	17	0,75	0,001	52	0,05	1000	27,07
9c.			30	20	(−2; −1,75; −1,5; −1,25; −1; 0,75; −0,5; −0,25; 0; 0,25; 0,5; 0,75; 1,0; 1,25; 1,5; 1,75; 2)	(−1,98; −1,73; −1,48; −1,23; −0,99; −0,74; −0,5; −0,25; 0; 0,25; 0,5; 0,75; 0,999; 1,25; 1,5; 1,74; 1,99)	(13,91; 11,3; 8,94; 6,83; 4,97; 3,34; 1,98; 0,86; −0,007; −0,63; −1,002; −1,12; −1,00; −0,63; −0,005; 0,86; 1,98)		0,75	0,001	32	0,05	5	25,10
10.	f10(x) [−10; 10]	20	30	20	(−10; −9; −8; −7; −6; −5; −4; −3; −2; −1; 1; 2; 3; 4; 5; 6; 7; 8; 9; 10)	(−10,004; −9,11; −8,02; 5,38; −4,93; −3,53; −2,918; −0,86; 0,55; 1,83; 2,99; 3,94; 6,53; 7,66; 8,56; 9,95)	(−2,64; −3,36; −9,47; 1,72; −0,35; −0,53; −1,99; 4,75; −3,54; 0,54; 2,96; 3,64; 3,04; −0,65; 0,46; −2,27)	18	−8,020	0,0000001	1	0,001	5	59,30
11.	f11(x) [−2;2]	3	4	5	(−2; 0;2)	(−0,68; 4,699)	(−0,82; 0)	1	0,68	0,001	12	0,1	5	0,10
12.	f12(x) [0; 50]	10	20	25	(2; 8; 12; 16; 20; 26; 32; 38; 44; 50)	(4,7; 10,99; 17,28; 23,56; 29,85; 36,13; 42,41; 48,70)	(−0,99; −1,00; −0,99; −1,00; −0,99; 0,99 −1,00; −1,00)	8	10,99	0,001	45	1	100	14,00

Table 2. The algorithm testing results: two-dimensional case

No.	The function, search domain	Nexp.	M, M1	K	Initial approximation τ^0	Local minima τ^*	The value of the function at the local minima	N_{true}	Global minima	Tolerance ε	Iterations I	The notes ho	The notes S	CPU time (s)
13.	f13(x,y) [−3;1] × [−2;2]	3	10	10	(−1;−1),(−1;−1),−1;−1)	(0.00; 0.00),(−2.00;0.00)	(0.00; 0.00)	2	(0.00;0.00)	0.001	32	1	100	10.06
			5	12	(−1;1),(−1;1),−1;1)	(−2.00;0.00),(0.00;0.00)	(0.00; 0.00)		(0.00; 0.00)	0.001	34	1	100	4.07
			5	15	(1;3),(1;3),(1;3)	(−2.00;0.00),(0.00;0.00)	(0.00; 0.00)		(0.00;0.00)	0.001	27	1	100	10.21
			5	15	(−3;2),(−3;2),(1;−2)	(−2.00;0.00),(0.00;0.00)	(0.00; 0.00)		(0.00; 0.00)	0.001	26	1	10	5.40
			5	12	(−3;2),(−3;2),(1;−2)	(−2.00;0.00),(0.00;0.00)	(0.00; 0.00)		(0.00; 0.00)	0.001	26	1	10	5.30
14.	f14(x,y) [−2;4] × [−2;4]	5	10	15	(4;−2),(4;−2),(4;−2),(4;−2)	(3.00; 2.07),(−2.00; 3.09),(3.58; −1.85)	(0.00; 15.58; 0.00)	3	(3.58; −1.85)	0.00001	46	1	10	27.35
			10	15	(−1;1),(−1;1),(−1;1),(−1;1),(−1;1)	(3.00; 2.03),(−2.00; 3.08),(3.58; −1.85)	(0.00; 15.61; 0.00)		(3.58; −1.85)	0.00001	61	1	100	35.10
15.	f15(x,y) [0: 4.2] × [0: 6.4]	4	15	10	(4.2; 6.4),(4.2; 6.4),(4.2; 6.4),(4.2; 6.4)	(1.09; 1.68),(3.83; 1.68)	(−2.14, −3.29)	2	(3.83; 1.68)	0.001	40	1	100	37.06
			10	15	(2.5; 5.5),(2.5; 5.5),(2.5; 5.5),(2.5; 5.5)	(3.83; 1.70),(1.11; 1.74)	(−3.30, −2.16)		(3.83; 1.70)	0.00001	69	1	100	75.10
			20	25	(3.0; 5.0),(3.0; 5.0),(1.0; 5.0),(1.0; 5.0)	(3.83; 1.75),(1.12; 1.78)	(−3.32, −2.17)		(3.83; 1.75)	0.001	64	1	1000	63.07
16.	f16(x,y) [−5; 10] × [0;15]	4	10	10	(10;0),(0;0),(−5,15),(0,15)	(9.42; 2.47),(3.14; 2.27),(−3.14; 12.28)	(0.40; 0.40; 0.40)	3	(9.42; 2.47)	0.001	43	1	100	50.20
			10	15	(10; 0),(10; 0),(−5,15),(−5,15)	(9.42; 2.47),(3.14; 2.27),(−3.14; 12.28)	(0.40; 0.40; 0.40)		(9.42; 2.47)	0.001	40	1	100	65.08
17.	f17(x,y) [−1; 2] × [−1; 1]	4	15	25	(0.2; 0),(0.4; 0.4),(0.9; 0.6),(1.4; 0.8)	(0.639; −0.9999),(1.389; 0.999),(1.9999; 0.1054)	(−0.9949; −0.543; −2.02181)	3	(1.9999; 0.1054)	0.00001	62	1	10000	141.15
			15	15	(0.2; 0),(0.4; 0.4),(0.9; 0.6),(1.4; 0.8)	(−0.999; 0.999),(1.9999; 0.1023)	(0.95466; −2.02176)		(1.9999; 0.1023)	0.00001	43	1	10000	63.00

Table 3. The algorithm testing results: three–dimensional case

No.	The function, search domain	Nexp.	M, M1	K	Initial approximation τ^0	Local minima τ^*	The value of the function at the local minima	N_{true}	Global minima	Tolerance ε	Iterations I	The notes ho	The notes S	CPU time (s)
18.	f18(x,y,z) [−0,5; 0,5] x [−0,5; 0,5] x [−0,5; 0,5]	3	10	15	(−0,5; −0,5; −0,5); (−0,5; 0; 0,5); (0,5; 0,5; 0,5)	(−0,00038; −0,00003; 0,00004)	0,0000002	1	(−0,00038; 0,00003;	0,001	22	1	100	
		3	5	7	(0,3: 0,3; 0,3); (0,3; 0,3; 0,3); (0,3;0,3;0,3)	(0,00039; 0,00027; −0,00029)	0,0000003		(0,00039; 0,00027; −0,00029)	0,001	30	1	100	
			5	7	(0,5;0,5;0,5); (0,5;0,5;0,5); (0,5; 0,5; 0,5)	(−0,0000679; −0,0000993; 0,0004)	0,0000002		(−0,0000679; −0,0000993 0,0004)	0,001	21	1	100	
		4	5	7	(0,5; 0,5; 0,5); (0,5; 0,5; 0,5); (0,5; 0,5; 0,5)	(−0,0000679; −0,0000993; 0,0004)	0,0000002	(−0,0000679; −0,0000993 0,0004)	0,001	21	1	100		
19.	f19(x,y,z) [0:1] x [0:1] x [0:1]	5	12	15	(0,5; 0,5; 0,5); (0,5; 0,5; 0,5); (0,5; 0,5; 0,5)	(0,106; 0,86; 0,56); (0,417; 0,5369; 0,828); (0,09; 0,56; 0,85); (0,368; 0,12; 0,267)	(−3,089; −3,74; −3,86; −1)	4	(0,09; 0,5576; 0,85;)	0,00001	269	1	100	

Notice that the parameter h_0, the initial sample step for N.Z. Shor's r-algorithm, as a rule, equals to 1. It is recommended to reduce h_0 for the segments, which have lengths not more than 1.

It should be noted that the user can regulate one more parameter, S, the penalty coefficient in the expression for $c(x, \tau_i)$. It is recommended to choose the value of S as a sufficiently large positive number, commensurable with the values of the function $f(x)$ and the other parameters. It is explained by the opportunity of the accidental r-algorithm stopping, because of overfilling, when S has a very large value. But when S has a small value, the exit of the point x from the local minimum attraction sphere is not penalized.

The parameters h_0 and S are not in the list of the fundamental algorithm parameters, as they may be given in the computer program automatically. Then there must be provided an automatic choice in a case of the accidental r-algorithm stopping.

Thus, to examine the properties of described algorithm we begin with minimizing the function $f_9(x)$ on $[0; 1]$. The graph of this function with $i - 1 = 1$, $i = 2$, $i + 1 = 3$ is presented in Figure 2. This function has three local minima $\tau_1^* = 0.25$, $\tau_2^* = 0.5$, $\tau_3^* = 0.75$.

The case corresponding to the 1-st experiment of the 9a series, given in Table 1 (the user specified $N_{\exp.}$ value coincides with $N_{\text{true}} = 3$) is presented in Figure 3b. Here are the initial approximations of the local minima $\tau_1^0 = 0.2$, $\tau_2^0 = 0.6$, $\tau_3^0 = 0.65$ and their attraction spheres Ω_1^0, Ω_2^0, Ω_3^0 correspondingly.

Figure 3a shows the optimal partitioning the set Ω into the attraction spheres $\Omega_1^*, \Omega_2^*, \Omega_3^*$ of the local minima $\tau_1^* = 0.25$, $\tau_2^* = 0.5$, $\tau_3^* = 0.75$, obtained by algorithm (with $M = 5$, $K = 5$).

Figure 4 corresponds to the second experiment of the 9a series. Here the user set value $N_{\exp.}$ equal to 4 that is greater than the true number $N_{\text{true}} = 3$. In this case the algorithm started from different initial local minimum points $\tau_1^0 = 0.2$, $\tau_2^0 = 0.4$, $\tau_3^0 = 0.6$, $\tau_4^0 = 0.8$ attained the optimal solution presented in Figure 3a.

In the third experiment of the 9a series with $N_{\exp.} = 4$, even though the algorithm was started from one initial point $\tau_1^0 = \tau_2^0 = \tau_3^0 = \tau_4^0 = 0.4$ it found three true local minima correctly ($N_{\text{true}} = 3$) as shown in Figure 3a .

In the fourth experiment of the 9a series with $N_{\exp.} = 5$, algorithm started from initial approximation $\tau_1^0 = 0.2$, $\tau_2^0 = 0.3$, $\tau_3^0 = 0.4$, $\tau_4^0 = 0.6$, $\tau_5^0 = 0.8$ found three true local minima also correctly (see Figure 3a).

In the 5-th experiment of the 9a series the $N_{\exp.}$ has the value less than N_{true}, i.e., $N_{\exp.} = 2$. The algorithm missed one local minimum $\tau_2^* = 0.5$ (see Figure 5) but the other 2 local minima were found correctly: $\tau_1^* = 0.25$, $\tau_3^* = 0.75$.

Thus, we can make a conclusion that to implement the algorithm it is not compulsory to know the true number of the local minima (N_{true}). It is enough to select $N_{\exp.} > N_{\text{true}}$. In this case, algorithm finds all N_{true} local minima. If, nevertheless, $N_{\exp.} < N_{\text{true}}$, then some local minima will be missed and the other ones will be

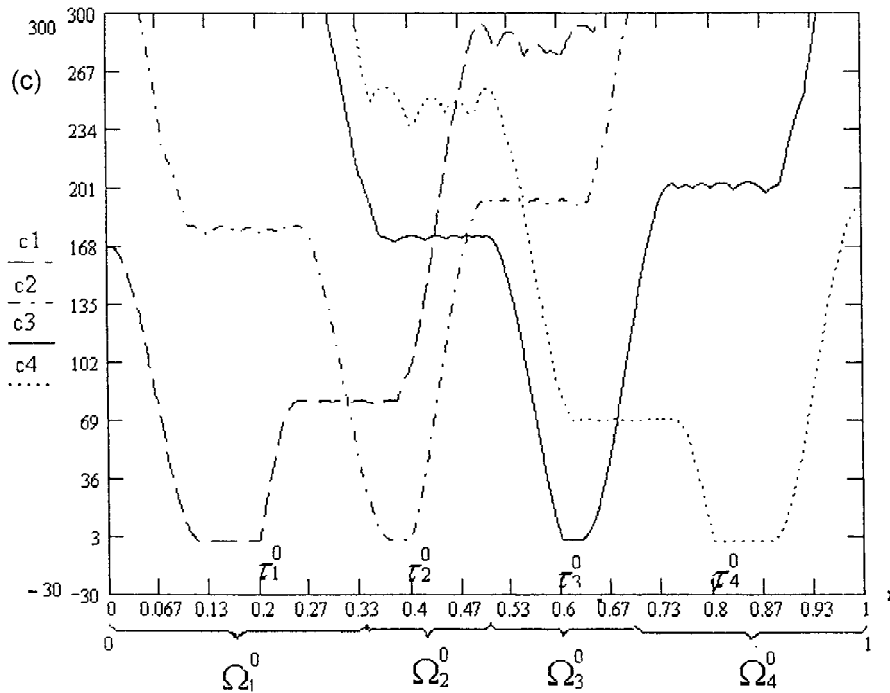*Figure 4.* Plot of the penalty functions of minimizing the $f_9(x)$ at the initial solution: $N_{\exp.} = 4$.

found correctly. In such a case the experiment should be repeated several times with increasing values of $N_{\exp.}$.

The same effect of implementation the algorithm is observed for the three-dimensional function $f_{18}(x)$ (see Table 3 with $N_{\exp.} = 3$ and $N_{\exp.} = 4$, the algorithm leads to the $N_{\text{true}} = 1$), as well as for lots of the two-dimensional functions from Table 2.

There is no difficulty with implementation the algorithm for searching for the eight local minima of the function $f_{12}(x)$ on [0;50] (see Table 1, experiment 12), and also the 17 local minima of the function $f_9(x)$ on [−2; 2] (see Table 1, experiment 9c; the graph of this function is presented in Figure 6).

We have performed a number of successful experiments with the function $f_{12}(x)$ that has 33 local minima on segment [0;200] and of the function $f_9(x)$ that has 33 local minima on segment [−4; 4].

Choice of the parameters M and K, generally speaking, depends on a type of the function $f(x)$. For more smooth functions, the M and K values may be chosen with less values, especially, if an initial approximation for τ is given successfully, for example, uniformly distributed on [a, b]: (see Table 1, $f_1(x)$, the 1-st experiment, here $M = 2$, $K = 3$ and Table 1, $f_9(x)$, 9a series: the 2-d experiment, here $M = 5$, $K = 6$).
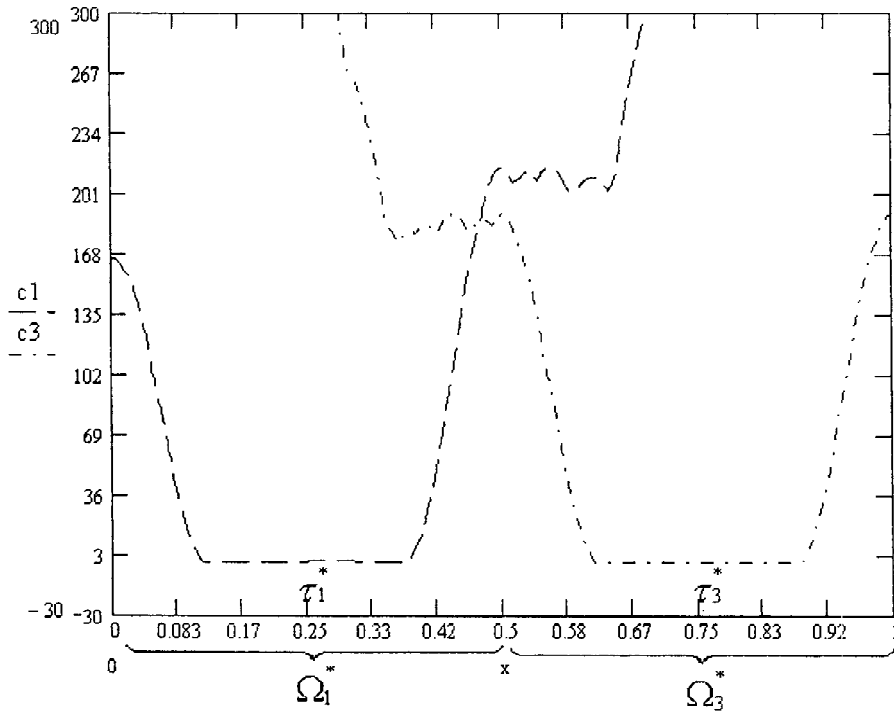
Figure 5. Plot of the penalty functions of minimizing the $f_9(x)$ at the optimal solution: $N_{\exp.} = 2$.

If the initial approximation of the local minima is bad, for example, all τ_i^0, $i = 1, ..., N$, set at the same point, then M and (or) K should be increased (see Table 1, $f_1(x)$, 2-d experiment, here M and K values are increased in comparison with the 1-st experiment, $M = 10$, $K = 10$). We can see the same for function $f_9(x)$ (see Table 1, the 9a series: experiment 3). Here the parameter K value is increased and equal 30 in comparison with the experiment 2 when $K = 6$.

Besides, if the initial approximation of the local minima is chosen poorly then some local minima can be missed. For example, the function $f_9(x)$ has 17 local minima on the admissible domain $[-2; 2]$ (see Figure 6). When we determine eight initial approximations for τ_i^0 instead of 17 (see Table 1, $f_9(x)$, experiment 9b) the algorithm found only 11 local minima and the rest are missed. If we specify an initial approximation τ_i^0 uniformly distributed (see Table 1, $f_9(x)$, experiment 9c), all local minima are found. Let us notice that in the first case with non-uniformly distributed initial approximation for the local minima parameters M and K values are increased and essentially increase the CPU time ($M = 40$, $K = 50$, CPU time $= 140.07$ s). Then in the second case $M = 30$, $K = 20$, CPU time $= 25.10$ s.

The above observations were also observed in the minimization of the function $f_6(x)$ (see Figure 7) that has three non-uniformly distributed local minima on segment $[a, b] = [0.05; 1]$. Naturally, the M and K values are increased: $M = 35$,
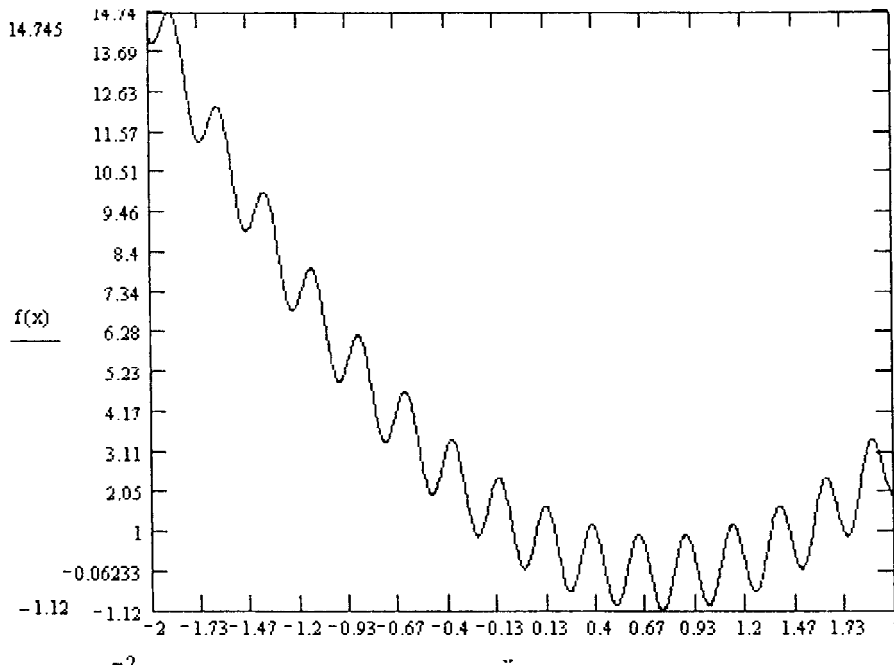
Figure 6. Plot of the minimized function $f_9(x)$ on $[-2; 2]$.

$K = 20$ (see Table 1, $f_6(x)$, experiment 1). Algorithm finds all local minima with CPU time = 4 s.

Note that the results of algorithm give us extra information about the locations of local maxima and saddle points of the examined function. So, for example, for function $f_9(x)$, see Figure 2, the points that are situated at the optimal bounds of the attraction spheres Ω_{i-1}^*, Ω_i^* and attraction spheres Ω_i^*, Ω_{i+1}^* are the local maximum points of this function. (The optimal set partitioning method from Section 3 of this paper has the possibilities to find the points which are situated at the optimal bounds between subsets Ω_i^* and Ω_j^*, $i, j = 1, ..., N$ (see Kiseleva, 1985).

At last, it follows from the numerous publications on the global optimization (for example, Ziglyavsliy and Zilinskas, 1991; Zilinskas, 1986) that for many well-known algorithms there are difficulties with the plane part of the hyper-surface that is determined by the minimized function (the so-called the Plateau problem). In the present paper algorithm is applied for minimization of the function $f_3(x)$ that has such a feature (see Figure 8). For algorithm there is no difficulty with the Plateau problem (see Table 1, $f_3(x)$, experiment 1, 2, 3).

By taking $N_{\exp.}$ equal to 3, 2, 1 we received $N_{\text{true}} = 1$ and one global minimum point from the optimal solution set $[0; 1]$.

So in 1-st experiment with $N_{\exp.} = 3$ even started from one initial point $\tau_1^0 = \tau_2^0 = \tau_3^0 = -2$ the algorithm finds 1 true local minimum ($N_{\text{true}} = 1$): $\tau_1^* = 0.0002019$ (here $M = 2$, $K = 2$).
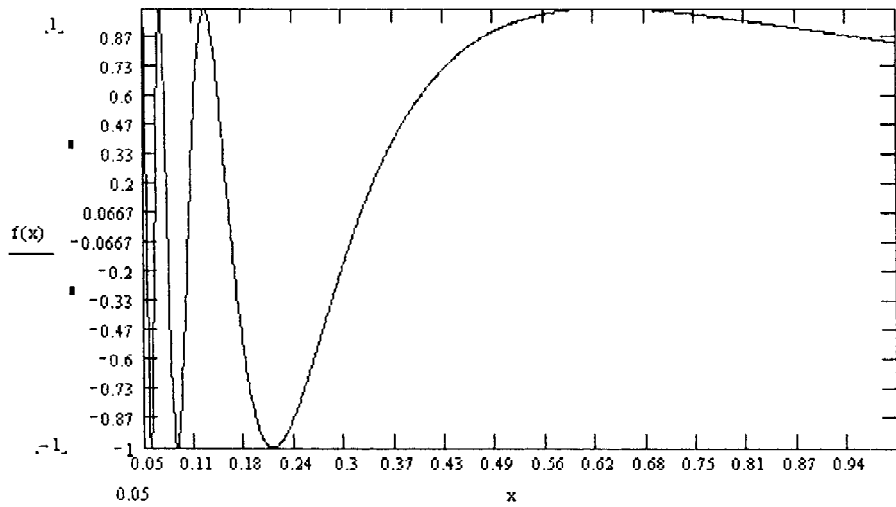
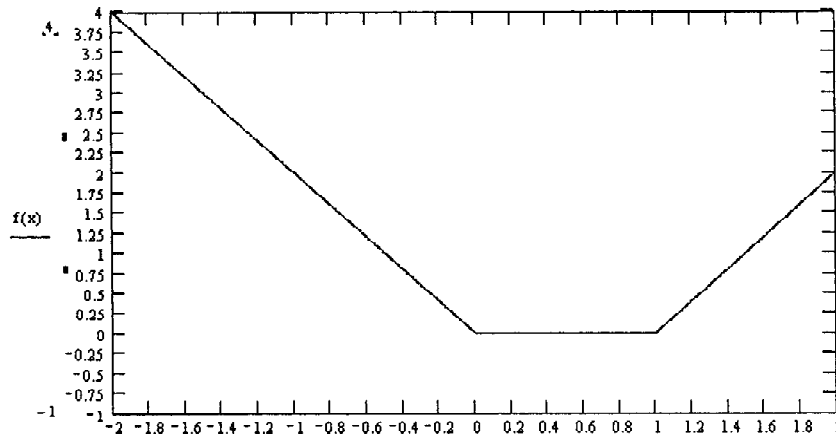*Figure 7.* Plot of the minimized function $f_6(x)$.

Figure 8. Plot of the minimized function $f_3(x)$.

In 2-d experiment with uniformly distributed initial approximation for local minima $\tau_1^0 = -2$, $\tau_2^0 = -1$, $\tau_3^0 = -1.6$ algorithm also finds 1 true local minimum: $\tau_1^* = 0.0000291$ (here $N_{exp.} = 3$, $M = 3$, $k = 2$).

In the third experiment with $N_{exp.} = 2$, the algorithm started from the initial approximation $\tau_1^0 = -2$, $\tau_2^0 = 0.4$, finds 1 true local minimum: $\tau_1^* = 0.0004419$ (here $M = 3$, $K = 2$).

In the fourth experiment even when $N_{exp.} = 1$, algorithm finds 1 true local minimum correctly: $\tau_1^* = -0.0000391$, $\tau_1^0 = -2$ with $M = 3$, $K = 2$.

The graphs of the penalty functions at the initial approximation $\tau_1^0 = -1$, $\tau_2^0 = 1.6$ are shown in Figure 9.
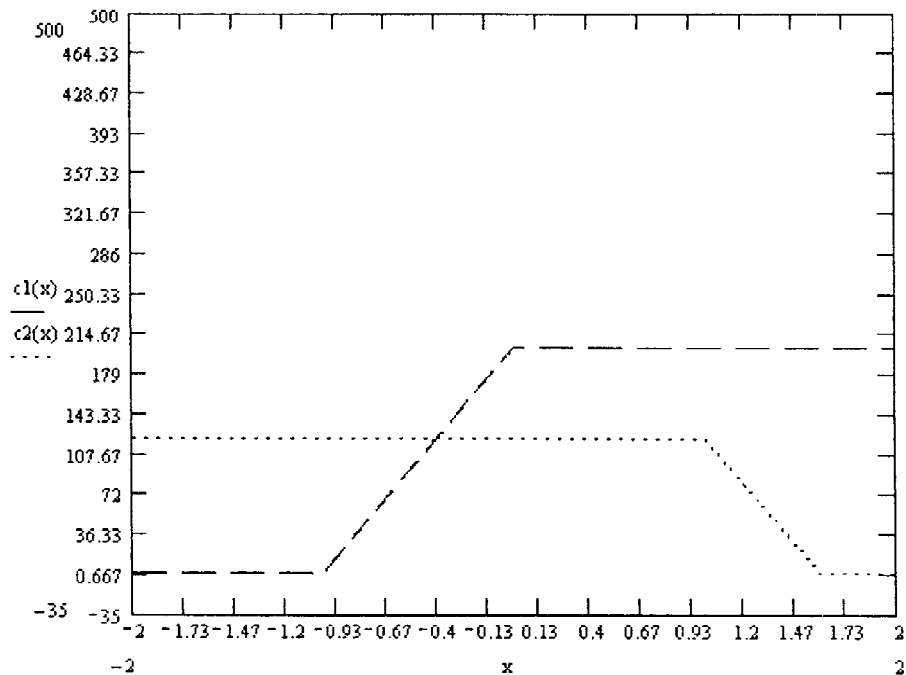
Figure 9. Plot of the penalty functions of minimizing the $f_3(x)$ at the initial solution.

The examined algorithm has been coded using Fortran 7.0. The Fortran code has been compiled with Fortran-IV Compiler. All computations for the mentioned set of the test functions have been carried out on a Pentium II PC.

7. Summary

Thus, in conclusion, let us list the main characteristics of the examined algorithm.

1. Under a suitable choice of the basic parameters N, M, K, algorithm for wide class of non-differentiable functions finds simultaneously all local minima, including the global one, their attraction spheres and also the true number of the local minima N_{true}.

2. The algorithm shows its worth in minimizing the function with a Plateau problem.

3. It is not necessary to know the true number of the local minima (N_{true}). It is enough to select $N_{\text{exp.}} > N_{\text{true}}$. In this case, the algorithm finds all N_{true} local minima. If, nevertheless, $N_{\text{exp.}} < N_{\text{true}}$, then some local minima will be missed and the other ones will be found correctly. In such a case the experiment should be repeated with increasing values of $N_{\text{exp.}}$ until the N_{true} has been determined.

4. It is recommended that the initial approximation for local minima τ_i^0, $i = 1, ..., N$, is distributed uniformly on $[a, b]$ or is chosen by means of the random

number generator, if there is no a priori information about the values of the local minima. If all the values for the initial approximation are at the same point, then the parameters M, K must be adjusted in order not to miss some local minima.

5. The recommendations for choosing the M, K parameters demand further correction. The user-algorithm interaction gives us the possibility of correction of these parameters during the process of solving the problem.

Acknowledgement

References

1. Batishev, D.I. (1975), *Poiskovie metodi optimalnogo proektirovania* (The searching methods of optimal projection), Moscow, Sovetskoe Radio (Soviet Radio) (in Russian).
2. Dem'yanov, V.F. and Vasil'ev, L.V. (1981), *Nedifferenciruemaya optimizatia* (Non-differentiable optimization), Moscow, Nauka (Science) (in Russian).
3. Floudas, C.A. (2000), *Deterministic Global Optimization. Theory, Methods and Applications*, Kluwer Academic Publishers, Dordrecht.
4. Himmelblau, D. (1975), *Applied Nonlinear Programming*, McGraw-Hill Book Company, New York.
5. Kiseleva, E.M. (1985), O primenenii metodov optimalnogo razbienia mnozestv k resheniu zadach globalnoi optimizatii (About the application the optimal set partitioning method to the global optimization problems), *Metodi reshenia nelineinih zadach i obrabotki dannih* (Methods for solving nonlinear problems and data processing), DSU:111-116 (in Russian).
6. Kiseleva, E.M. (1989), Reshenie odnoi zadachi optimalnogo razbienia s razmesheniem centrov tyazesti podmnozestv (Solution of the one optimal partitioning with allocation gravity centers of these subsets), *J. of Computational Math. and Math. Physics* 29(25), 709–722 (in Russian).
7. Kiseleva, E.M. (1991), Matematicheskie metodi i algoritmi reshenia neprerivnih zadach optimalnogo razbienia mnozestv i ih prilozenia (Mathematical methods and algorithms for solving the continuous problem and its applications), *Auto abstract, dissert. doct. of math. & physics sciences,* Kiev: 33 (in Russian).
8. Kiseleva, E.M. and Shor, N.Z. (1991), Reshenie neprerivnoi zadachi optimalnogo razbienia v usloviyzh nepolnoi informacii ob ishodnih dannih (Solution the continuous problem of the optimal partitioning in uncertain information about the original data condition), *J. of Computational Math. and Math. Physics* 31(6), 799–809 (in Russian).
9. Kiseleva, E.M. (1992), Solution of the generalized Neyman-Pearson problem using the methods for the optimal partitioning of sets, *J. of Computational Math. and Math. Physics* 31(1), 167–173.
10. Kiseleva, E.M. and Shor, N.Z. (1994), Issledovanie algoritma reshenia odnogo klassa neprerivnih zadach razbienia (Examination the algorithm for solving one class of continuous partitioning problems), *Kibernetika i sistemniy analiz* (Cybernetics and System Analyses), 1: 84–96 (in Russian).
11. Kiseleva E.M. and Mironenko, A.A. (1996), Primenenie metoda optimalnogo razbienia mnozestv k resheniu zadach dvumernoi optimizacii (Application the optimal set partitioning method to the problems of two-dimensional global optimization), *Matem. Modeli i vichislitel-*

nie metodi v prikladnih zadachah (Math. Models and Computational Methods in Application Problems), DSU: 70–73 (in Russian).

12. Mihalevich, B.C., Trubin, B.A. and Shor, N.Z. (1986), *Optimizasionnie zadachi proisvodstvenno-transportnogo planirovania* (Optimization problems of transport and manufacturing facilities planning), Moscow, Nauka (Science) (in Russian).

13. Strongin, R.G. (1978), *Chislennie methodi v mnogoextemalnih zadachah* (The numerical methods in the multiextremal problems), Moscow, Nauka (Science) (in Russian).

14. Suharev, A.G. (1981), Optimalnie i posledovatelno-optimalnie algoritmi v zadachah chislennogo analiza (Optimal and sequential algorithms in the numerical analyses problem), *Matem. Metodi v issledovanii operacii* (Mathematical methods in the operation research), Moscow: 54-68 (in Russian).

15. Suharev, A.G. (1981), Clobalni extremum i metodi ego otiskania (The global extremum and the methods of serching it), *Matem. Metodi v issledovanii operacii* (Mathematical Methods in the Operation Research), Moscow: 64–69 (in Russian).

16. Suharev, A.G. (1989), *Minimaksnie algoritmi v zadachah chislennogo analiza* (Minimax algorithms in the numerical analyses problems), ?oscow, Nauka (Science) (in Russian).

17. Ziglyavskiy, A.A. (1985), *Matematicheskaya teoria globalnogo sluchainogo poiska* (The Mathematical Theory of the Global Random Search), LSU (in Russian).

18. Ziglyavskiy, A.A. and Zilinskas, A.G. (1991), *Metodi poiska globalnogo exstremuma*, (Methods of the Global Search), Moscow, Nauka (Science) (in Russian).

19. Zilinskas, A.G. (1986), *Aksiomatika statisticheskih modeley, algoritmi, primenenie* (The Statistic Model Axiomatics, Algorithms Applications), Mokslas, Vilnus, Nauka (Science) (in Russian).

20. Zilinskas, A.G. and Shaltyanis, B.R. (1989), *Poisk optimuma* (The Optimum Search), Moscow, Nauka (Science) (in Russian).

21. Shor, N.Z. (1985), *Minimization Methods for Non-Differentiable Functions*, Springer, Berlin.

22. Shor, N.Z. (1998) *Nondifferentiable Optimization and Polynomial Problems. Nonconvex Optimization and its Applications.* Kluwer Academic Publishers, Dordrecht