A Multidimensional Bisection Method for Unconstrained Minimization Problem

E.Y. Morozova

Applied Mathematics Department
Herzen State Pedagogical University of Russia
48, Moika Emb., St.-Petersburg, 191186, Russia

melena65@mail.ru

Abstract

An extension of a new multidimensional bisection method for minimizing function over simplex is proposed for solving nonlinear unconstrained minimization problem. The method does not require a differentiability of function, and is guaranteed to converge to the minimizer for the class of strictly unimodal functions. The computational results demonstrating an effectiveness of algorithm for minimizing nonsmooth functions are presented.

Keywords: Convex set, *n*-dimensional simplex, strictly unimodal function, direct search methods, nonlinear unconstrained optimization.

1 Introduction

The problem considered here is an unconstrained minimization problem, which has the general form:

$$f(x) \to \min, \ x \in R^n, \qquad \text{(P)}$$

where $f: R^n \to R$ is a bounded below continuous strictly unimodal function.

We use the following definition of strict unimodality.

Definition. Let D be a bounded closed convex set in R^n. Function $f: D \to R$ is *strictly unimodal over set D* iff for any segment $\Delta \subset D$ $\# Arg\min\{f(x) | x \in \Delta\} = 1$, where «$\# A$» is the cardinality of set A.

The multidimensional bisection method (Baushev and Morozova, 2007) allows to solve constrained minimization problem when the feasible region is n-dimensional simplex. This method generalizes a one-dimensional bisection method for the case n>1 using a recursive procedure. This paper will present an extension of the multidimensional bisection method for solving problem (P). This method does not require a differentiability of function f, and is guaranteed to converge to the minimizer for the class of strictly unimodal functions.

It is known a class of methods that do not explicitly use derivatives - direct search methods for unconstrained optimization. Recent researches have shown the global convergence of pattern search algorithms (a class of direct search methods) for the case when function f is continuously differentiable (Aude & Dennis, 2003, Torczon, 1997). The advantage of the multidimensional bisection method presenting in this paper is that it convergence does not require an assumption about differentiability of function f and method allows to find the minimizer of nonsmooth functions.

In point 2 we describe the multidimensional bisection method (MBM) for minimizing function over simplex. In point 3, the details of the extension of MBM for solving the unconstrained minimization problem are presented. In point 4 some numerical results illustrate the robustness of the method.

2 The Multidimensional Bisection Method

The problem considered is

$$f(x) \to \min, \quad x \in S, \qquad (1)$$

where S - *a n-dimensional* simplex in R^n, and f – a continuous function.

1. Case *n*=1.

The one-dimensional bisection algorithm solves the problem

$$f(x) \to \min, \quad x \in [a,b], \qquad (2)$$

where f is a strictly unimodal function over segment $[a,b]$.

Let $bis(f,a,b,\varepsilon)$ denote the recursive one-dimensional bisection procedure. The inputs for this procedure are: the procedure for calculation values of f, the segment $[a,b]$ and the accuracy ε. The outputs are the estimations x_m for the minimizer x^* and f_m for the value of the minimum of the function f over the segment $[a,b]$.

The iteration of the recursive procedure includes the following steps.

Step 0. If $b-a \ge \varepsilon$, go to step 1, otherwise stop.

Step 1.

$$c = \frac{a+b}{2} \ a' = \frac{a+c}{2}, \ b' = \frac{b+c}{2}, f(c), \ f(a'), \ f(b').$$

Step 2.

If $f(a') \le f(c) \le f(b')$, set $b = b'$.

If $f(a') \ge f(c) \ge f(b')$, set $a = a'$.

If $f(c) \le \min\{f(a'), f(b')\}$, set $a = a'$, $b = b'$.

Step 3. Execute $bis(f, a, b, \varepsilon)$ with new inputs.

2. Case *n*>1.

Let S be a n-dimensional simplex. Let fix the vertex V^0 and denote by $V^1,...,V^n$ the opposite vertices. Set for each $t \in [0,1]$

$$S_t = conv\{V^0 + t(V^1 - V^0),...,V^0 + t(V^n - V^0)\}. \qquad (3)$$

The set S_t is the $n-1$-dimensional simplex for $0 < t \le 1$.

Set $x_t \equiv \arg\min\{f(x) \mid x \in S_t\}$. Each simplex S_t for $0 < t < 1$ part the initial simplex S in two sets: $conv\{V^0, S_t\}$ and $conv\{S_1, S_t\}$. Fig. 1 illustrate an example of the partition of the simplex S for the case $n = 3$.

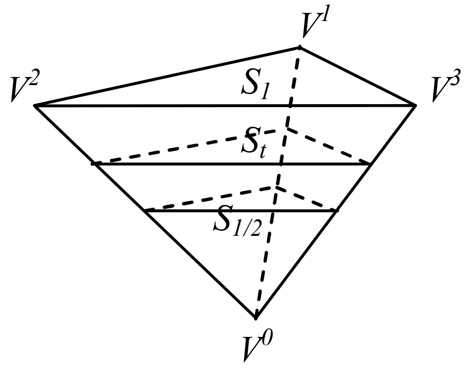

Figure 1: The partition of the simplex S

Let $bissimpl(f, SS_1, SS_2, d, \varepsilon)$ denote the recursive procedure in case $n > 1$. The inputs for this procedure are: the procedure for calculation values of *f*, boundary simplices SS_1 and SS_2, the current dimension *d* and the accuracy *ε*. The outputs are the estimations x_m for the minimizer x^* and f_m for the value of the minimum of the function f over the set $conv\{SS_1, SS_2\}$. Originally *d* is equal *n*, then this parameter varies depending on the dimension of the simplex where the point of a minimum is searched. Actually this parameter at first decreases to value 1, and then increases to value *d=n.* Three circles of such calculations we consider as the iteration with number *k*. Denote by f^k the estimation of a minimum of the function f and by x^k the estimation of the point x^*. The parameter *d* and the outputs must be declared as global variables and its initial values must be defined before starting procedure $bissimpl(f, SS_1, SS_2, d, \varepsilon)$. More concretely the preliminary step includes the following destinations:

$SS_1 = S_0 = V^0$, $SS_2 = S_1 = conv\{V^1,\ldots V^n\}$ according to (7), *d=n*;

x^0, x^1, f^0, f^1 we define in a such way that the condition

$$\max\left\{\left|f^k - f^{k-1}\right|, \left\|x^k - x^{k-1}\right\|\right\} < \varepsilon \qquad (4)$$

be failed.

Step 1.

If (4) is hold, stop. Otherwise set $\sigma_1 = SS_1$, $\sigma_2 = SS_2$ and go to step 2.

Step 2.

If *d*=1, execute $bis(f, a, b, \varepsilon)$ with a= SS_1, b= SS_2 Otherwise, go to step 3.

Step 3.

Two cases are possible.

1) SS_1 and SS_2 are *d*-dimensional simplices. Let $V^0_{SS_1}, V^1_{SS_1},...,V^d_{SS_1}$ and $V^0_{SS_2}, V^1_{SS_2},...,V^d_{SS_2}$ be vertices of simplices SS_1 and SS_2 accordingly. Then we define $S_{\frac{1}{2}}$ $S_{\frac{1}{4}}$ $S_{\frac{3}{4}}$ by

$$\begin{aligned} S_t = conv\Big\{&V^0_{SS_1} + t\left(V^0_{SS_2} - V^0_{SS_1}\right), V^1_{SS_1} + t\left(V^1_{SS_2} - V^1_{SS_1}\right), \\ &..., V^d_{SS_1} + t\left(V^d_{SS_2} - V^d_{SS_1}\right)\Big\}. \end{aligned} \qquad (5)$$

2) One of the sets SS_1, SS_2 is a vertex, another is *d*-dimensional simplex. In this case $S_{\frac{1}{2}}$ $S_{\frac{1}{4}}$ $S_{\frac{3}{4}}$ are defined by (3). Set $d = d - 1$.

Step 4.

For each of simplices $S_{\frac{1}{2}}$ $S_{\frac{1}{4}}$ $S_{\frac{3}{4}}$ the following actions must be done:

1) Fix a vertex V^0 in the simplex S_t and let $V^1,...,V^n$ be an opposite vertices.

2) Execute $bissimpl(f, SS_1, SS_2, d, \varepsilon)$ with new values $SS_1 = V^0$ and $SS_2 = conv\{V^1,...,V^n\}$.

Step 5.

Let x^1_m x^2_m, x^3_m and f^1_m f^2_m, f^3_m be results of the previous step (for $S_{\frac{1}{2}}$ $S_{\frac{1}{4}}$ $S_{\frac{3}{4}}$ accordingly).

If $f^2_m \le f^1_m \le f^3_m$, set $SS_1 = \sigma_1$ $SS_2 = S_{\frac{3}{4}}$.

If $f^2_m \ge f^1_m \ge f^3_m$, set $SS_1 = S_{\frac{1}{4}}$ $SS_2 = \sigma_2$.

If $f^1_m \le \min\{f^2_m, f^3_m\}$, set $SS_1 = S_{\frac{1}{4}}$, $SS_2 = S_{\frac{3}{4}}$.

Set $d = d + 1$.

Step 6.

Execute $bissimpl(f, SS_1, SS_2, d, \varepsilon)$ with current inputs.

The following theorem presents the convergence result.

Theorem. Let $x(\varepsilon)$ be the final estimation of the minimizer x^* for the function *f* where f is a continuous strictly unimodal function over *n-dimensional* simplex *S* then $\lim\limits_{\varepsilon \to 0} x(\varepsilon) = x^*$.

3 The main algorithm

Consider problem (P). The algorithm consecutively solves the following constrained optimization problems:

$$f(x) \to \min, \; x \in S^k, \qquad (6)$$

where S^k is *a n-dimensional* simplex in R^n, k is a number of iteration, k=0, 1, 2,…. Each of problems (6) is solved by the multidimensional bisection method described above. Let $x^k = \{\arg\min f(x) \mid x \in S^k\}$. The algorithm constructs simplexes S^k using two basic operations of reflection and shift, so that $x^{k-1} \in \mathrm{int}S^k$, and generates a sequence of points $\{x^k\}$ with decreasing vales of f:

$$f(x^k) \le f(x^{k-1}), \quad k \in N.$$

This iterative process stops when point $x^k \in \mathrm{int}S^k$. A more formal description of the algorithm is as follows:

Step 1.
Choose $S^0 = conv\{v^{0,i} \,|\, i = 0,...,n\}$ - an arbitrary initial simplex. Set k=0.
Step 2.
Call the procedure *bissimpl* which solves problem (1).
Step 3.
Finding barycentric coordinates

$$\lambda_i \ge 0, \; i = 0,...,n, \; \sum_{i=0}^{n} \lambda_i = 1 \qquad (7)$$

of the point $x^k \in S^k = conv\{v^{k,i} \,|\, i = 0,...,n\}$.
Step 4.
If $x^k \in \mathrm{Int}S^k$, then set $x^* = x^k$, stop; otherwise go to step 5.
Step 5.
We have $x^k \in \partial S^k$. Construction of the simplex S^{k+1} by the procedure *reflect*$(v^0,...,v^n, x^k, \lambda_0,...,\lambda_n, \theta)$ (we shall describe *reflect* below).
Step 6.
Set k=k+1. Go to step 2.

Now consider step 5 in details and describe the procedure *reflect*$(v^0,...,v^n, x^k, \lambda_0,...,\lambda_n, \theta)$.
The inputs for this procedure are: the vertices $\{v^i \,|\, i = 0,...,n\}$ of the simplex S^k, the point $x^k = \{\arg\min f(x) \mid x \in S^k\}$, barycentric coordinates (7), small positive number θ. The outputs are the vertices $\{v^{k+1,i} \,|\, i = 0,...,n\}$ of new simplex S^{k+1}. Let

$$I_0(v) = \{i \,|\, \lambda_i(v) = 0\}, \; i = 0,...,n,$$

$$I_1(v) = \{j \,|\, \lambda_j(v) \ne 0\}, \; j = 0,...,n.$$

The procedure *reflect* includes two following operations (at iteration with number k):

1. *Reflection.* This move reflects the points $v^{k,i}$ for all $i \in I_0(v)$ through the point x^k:

$$\tilde{v}^{k+1,i} = v^{k,i} + 2\left(x^k - v^{k,i}\right) \text{ for all } i \in I_0(v).$$

2. *Shift.* Parallel displacement of the vertices $\tilde{v}^{k+1,i}, v^{k,j}$ for all $i \in I_1(v), j \in I_1(v)$ of the simplex $\tilde{S}^{k+1}$ along vector $\left(x^k - x^c\right)$, where x^c - centroid of $\tilde{S}^{k+1}$:

$$v^{k+1,i} = \tilde{v}^{k+1,i} + \theta\left(x^k - x^c\right) \text{ for all } i \in I_0(v),$$

$$v^{k+1,j} = v^{k,j} + \theta\left(x^k - x^c\right) \text{ for all } j \in I_1(v),$$

where θ - some small positive number.

Then $S^{k+1} = conv\left(v^{k+1,i} \,|\, i = 0,...,n\right)$ and we get $x^k \in \mathrm{Int}S^{k+1}$ (figure 2).

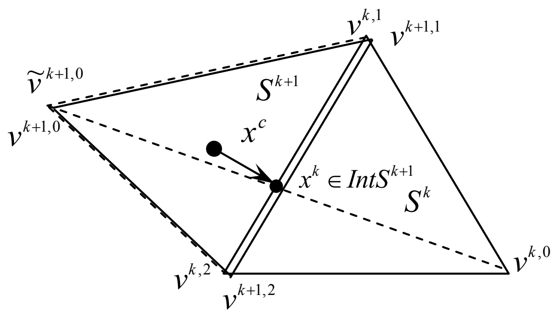

Figure 2: Construction of the simplex S^{k+1} in space R^2

Remark. Note that we get an ε-approximate solution to the original minimization problem (P), where ε is an input for procedure *bissimpl*.
The following lemma is needed to prove the convergence of our algorithm.

Lemma. Let f be continuous strictly unimodal function on the set D and let segment $\left[x^1, x^2\right] \subset D$. If $x^3 \in \mathrm{Int}\left[x^1, x^2\right]$ and $f\left(x^1\right) < f\left(x^3\right)$, then $f\left(x^3\right) \le f\left(x^2\right)$.

Proof. Assume that $f\left(x^3\right) > f\left(x^2\right)$. Then there is $x^* \in \mathrm{Int}\left[x^1, x^2\right] = \arg\max\left\{f(x) \,\middle|\, x \in \left[x^1, x^2\right]\right\}$ and there are positive numbers δ_1, δ_2 such that function f increases over segment $\left[x^* - \delta_1, x^*\right]$ and decreases over segment $\left[x^*, x^* + \delta_2\right]$. Then $f\left(x^* - \delta_1\right) = f\left(x^* + \delta_2\right)$ by virtue of continuity of the function f, i.e. the points $x^* - \delta_1$ and $x^* + \delta_2$ are minimizers of the function f over segment $\left[x^* - \delta_1, x^* + \delta_2\right]$ that contradicts to definition of strict unimodality. □
The following theorem presents the convergence result.
Theorem. If $f : R^n \to R$ is a bounded below continuous strictly unimodal function, S^0 is an arbitrary simplex, $\{x^k\}, k = 0, 1, 2, ...$ and $S^1, S^2, ..., S^k, ...$ are found by the

above described algorithm then there is number k^* such that $x^* = \{\arg\min f(x) \mid x \in R^n\} \in \operatorname{int} S^{k^*}$.

Sketch of proof. The algorithm of this paper generates the sequence $\{x^k\}$ and $x^k = \{\arg\min f(x) \mid x \in S^k\}$. If $x^k \in \operatorname{int} S^k$, then x^k solves problem (1) by virtue of strict unimodality of function *f*. If $x^k \in \partial S^k$, then $x^k \in \operatorname{int} S^{k+1}$ according to the rule of construction of simplex S^{k+1}. Let Γ^k be the nearest to the point x^* $n-1$–dimensional face of simplex S^k. We shall show that $x^k \in \Gamma^k$. Assume that $x^k \in \tilde{\Gamma}^k$. Consider segment $[x^k, x^*]$. Let y^k be the point of intersection of face Γ^k with segment $[x^k, x^*]$. Then $f(x^k) < f(y^k) > f(x^*)$, that contradicts to lemma. So, $x^k \in \Gamma^k$. Let $\rho_k = \rho(x^*, \Gamma^k) = \min\{\rho(x^*, y) | y \in \Gamma^k\}$. We shall show that $\liminf \rho_k = 0$. Let z^k be the point of emergence of ray $x^k + t(x^* - x^K)$ from simplex S^k and $r_k = \rho(z^k, x^*)$. Sequence r_k is monotonically decreasing to zero, $\lim r_k = 0$. So, $\liminf \rho_k = 0$. Thus, there is number k^* such that $x^* = \{\arg\min f(x) \mid x \in R^n\} \in \operatorname{int} S^{k^*}$. □

4 The numerical results

We implemented the multidimensional bisection method discussed above in MATLAB. The program was tested for different examples of minimization of nonsmooth functions. Some of numerical results we present in this section, some other examples can be found in (Morozova, 2006).

Example 1. Minimization of Dennis-Wood function.
Consider the following variant of Dennis-Wood function (Dennis & Wood, 1987):

$$f(x) = \frac{1}{2}\max\left\{\|x - c_1\|^2, \|x - c_2\|^2\right\}, \tag{8}$$

where $c_1 = (1, -1)$ $c_2 = -c_1$. This function is continuous and strictly convex, but its gradient is discontinuous everywhere on the line $x_1 = x_2$.

As shown in some works (Kolda and others, 2003, Torczon, 1991) such of direct search methods as compass search, multidirectional search algorithm can fail to converge to the minimizer of function (8).

We will illustrate the convergence of our algorithm to the minimizer of function (8).

The level sets of function (8) and the sequences of the simplexes S^k are shown in figure 3.

The sequence of the points x^k generated by our algorithm converges to the minimizer $x^5(0,0)$ as shown in figure 3. The regular simplex with centre $(1.5; -1)$ and the length of edge $l = 1$ was chosen as an initial simplex. The accuracy ε was chosen equal 10^{-6}. Figure 4 illustrates decreasing function values at the each of six iteration.

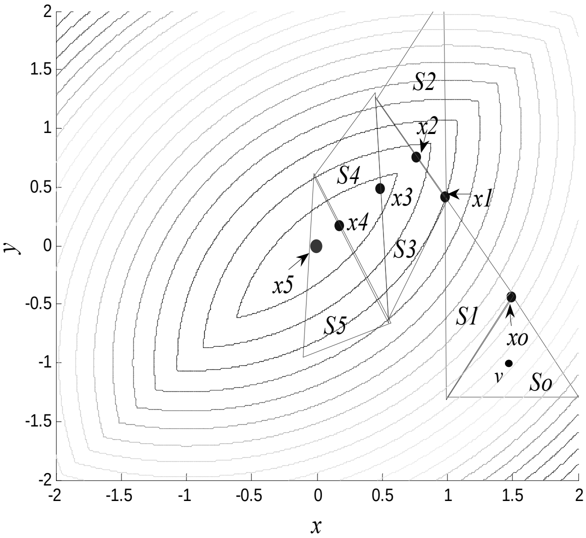

Figure 3: The level sets of the function (3), the sequences of the simplexes S^k and $\{x^k\}$

$x^*(0; 0)$, $f_{\min} = 1$

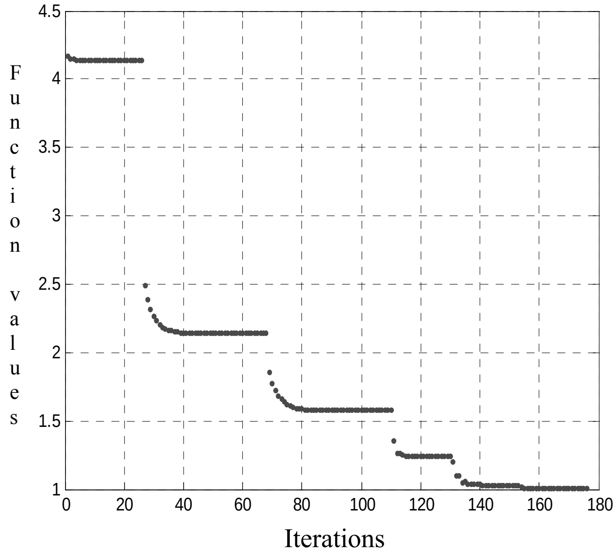

Figure 4: Decreasing function values at the each iteration

Table 1 shows the computational results for each of 6 iterations.

Iteration, k	Minimizer $x^k \subset S^k$	$f(x^k)$
0	$x^0(1.4910; -0.4382)$	4.1368
1	$x^1(0.9821; 0.4123)$	2.1370
2	$x^2(0.7575; 0.7575)$	1.5738
3	$x^3(0.4884; 0.4884)$	1.2385
4	$x^4(0.1717; 0.1717)$	1.0295
5	$x^5(0.0000; 0.0000)$	1.0000

Table 1: Iterative results

Example 2. Minimization of McKinnon function.

This example was chosen for comparing with the most popular direct search method – the Nelder-Mead algorithm (Nelder and Mead, 1965) which convergence is proved only for dimension 1 and some limited results for dimension 2 (Lagarias and others, 1998). At the same time there are examples of family of functions in R^2 (McKinnon, 1998) which demonstrate that the Nelder-Mead simplex algorithm can fail to converge to a stationary point of *f*. Consider the following function of this family:

$$f(x,y)=\begin{cases} 360x^2+y+y^2, & x\le 0\\ 6x^2+y+y^2, & x\ge 0\end{cases}. \qquad (9)$$

Function (9) is strictly convex and has up to three continuous derivatives. As shown in (McKinnon, 1998) if the initial simplex is $S^0=conv\{v^0,v^1,v^2\}$,

$$v^0=(0,0),\ v^1=(\lambda_1,\lambda_2),\ v^2=(1,1),$$

$$\lambda_1=\frac{1+\sqrt{33}}{8},\ \lambda_2=\frac{1-\sqrt{33}}{8}, \qquad (10)$$

then all vertices in the Nelder-Mead method converge to a nonminimizing point.

We illustrate that our algorithm applied to the function (9) converges to the minimizer. The initial simplex was equal $S^0=conv\{v^0,v^1,v^2\}$, where vertices v^0,v^1,v^2, and values λ_1,λ_2 where chosen according to (10). The accuracy ε was chosen equal 10^{-6}. As shown in figure 5, point $x^0(0.0542,-0.0381)\in\partial S^0$, $f(x^0)=-0.0190$. After constructing new simplex S^1 and performing the first iteration of our algorithm we have point $x^1(0,-0.5)$ with function value $f(x^1)=-0.25$. Point $x^1\in\operatorname{int}S^1$ is the minimizer of function (9).

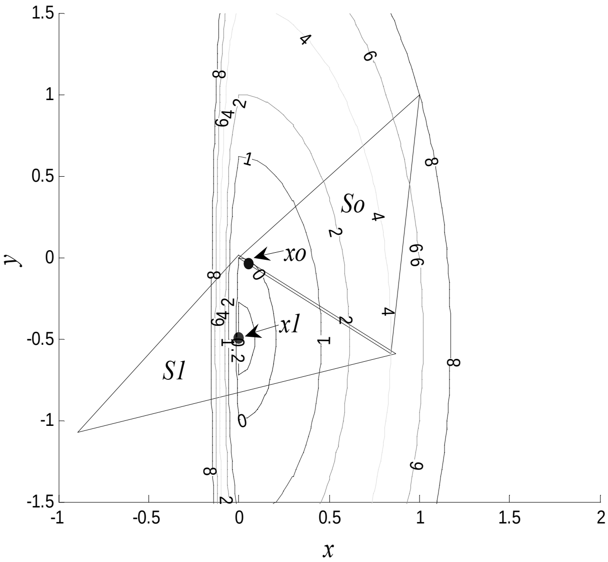

Figure 5: The level sets of the function (9), the sequences of the simplexes S^k and $\{x^k\}$

The level sets of function (9), the sequences of the simplexes S^k and the minimizers x^k are shown in figure 5. Figure 6 illustrates decreasing function values at the each of two iteration.

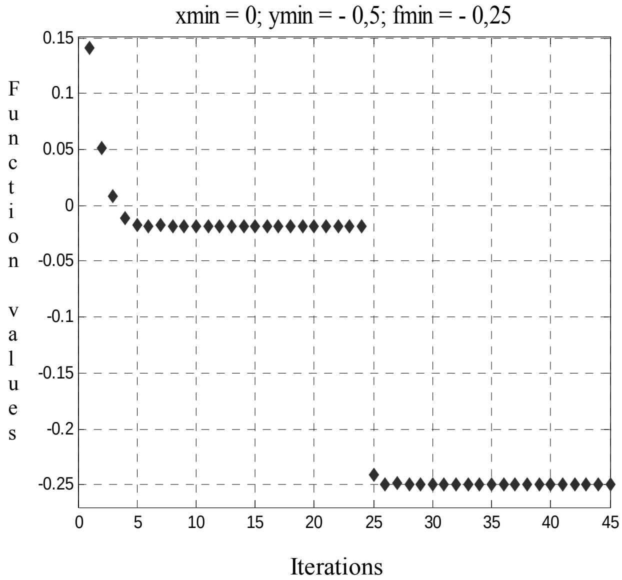

Figure 6: Decreasing function values at the each iteration

Example 3. Minimization of nonsmooth function.

The basic advantage of our algorithm is that it guarantees the convergence for a nonsmooth functions. Consider the following family of nonsmooth functions:

$$f(x,y)=\sum_{k=1}^{n}|x-a_k|^p+\sum_{k=1}^{n}|y-b_k|^p, \qquad (11)$$

where a_k and b_k are some pairwise different real numbers, *n* is an odd number, $0<p\le 1$. If we sort the numbers a_k and b_k in increasing order then the medium point minimizes function (11).

Figure 7 illustrates the convergence of our algorithm for this family of functions when $p=\frac{1}{2}, n=11$ and the numbers a_k and b_k were chosen from the uniform distribution on the segment $[0,1]$. Point $x^2\in\operatorname{int}S^2$ is the minimizer of function (11). Figure 8 illustrates decreasing function values at the each of three iterations.

Table 2 shows the computational results for each of 3 iterations.

Iteration, k	Minimizer $x^k\subset S^k$	$f(x^k)$
0	$x^0(17;\ 2)$	17.4847
1	$x^1(4.6858;\ 0.5711)$	8.2839
2	$x^2(0.6649;\ 0.5711)$	3.1503

Table 2: Iterative results

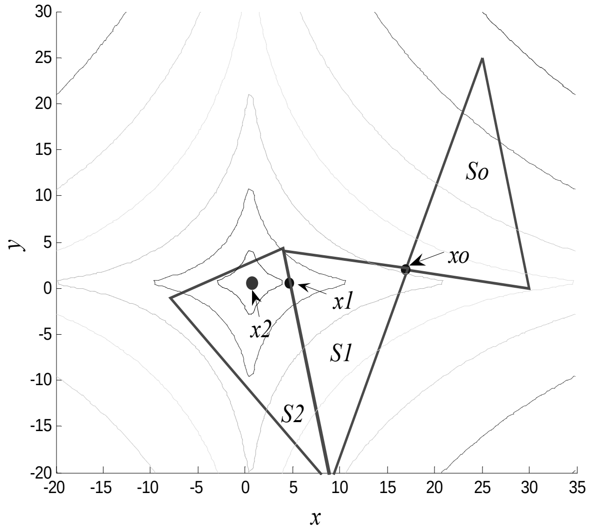

Figure 7: The level sets of the function (11), the sequences of the simplexes S^k and $\{x^k\}$

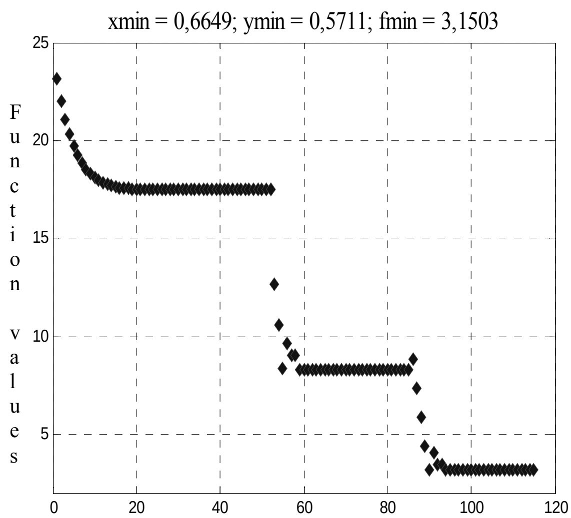

Figure 8: Decreasing function values at the each iteration

5. Conclusion

We have exposed our algorithm for the class of strictly unimodal functions only. However one can show that the algorithm can be applied for a wider class of functions, namely, we consider the class of functions Φ_S, where S - the n-dimensional simplex, defined as follows: $f \in \Phi_S$ iff for any segment $\Delta \subseteq S$ each local minimum of f over this segment is also a global minimum of the function f over this segment. The class Φ_S contains a subclass of strictly unimodal functions over set S. Function (11) considered in last example 3 is belong to the class Φ_S.

References

Baushev A.N., Morozova E.Y (2007): A multidimensional bisection method for minimizing function over simplex. *Lectures notes in engineering and computer science*, **2**:801-803.

Aude C. Dennis J.E. (2003): Analysis of Generalized Pattern Searches. *SIAM J. Optim,* **13**(3):889-903.

Torczon V. (1997): On the convergence of Pattern Search Algorithms. *SIAM J. Optim*, **7(**1):1-25.

Dennis, J. E., Woods, Jr. and Daniel, J. (1987): Optimization on microcomputers: The Nelder-Mead simplex algorithm, *in New Computing Environments: Microcomputers in Large-Scale Computing*, A. Wouk, ed., SIAM, Philadelphia, 116-122.

Morozova, E.Y. (2006): The direct search recursive algorithm for minimizing function of several variables. *The Review of applied and industrial mathematics*, **13**(5):783-796. (In Russian).

Kolda, T. J., Lewis, R. M., Torczon, V. (2003): Optimization by Direct Search: New Perspectives on Some Classical and Modern Methods. *SIAM Review*, **45**(3):385-482.

Torczon, V. (1991): On the convergence of the multidirectional search algorithm. *SIAM J. Optim.*, **1**:123-145.

Nelder J.A. and Mead R. (1965): A simplex method for function minimization. *Computer Journal*, **7**: 308-313.

Lagarias J.C., Reeds J.A., Wright M.H. and Wright P.E. (1998): Convergence properties of the Nelder Mead simplex algorithm in low dimensions. *SIAM J. Optim*, **9**: 112-147.

McKinnon, K.I.M. (1998): Convergence of the Nelder-Mead simplex method to a non-stationary point. *SIAM J. Optim*, **9**: 148-158.