Professor Radu SERBAN, PhD
Department of Mathematics
Assistant Daniela TODOSE, PhD Candidate
Department of Statistics and Econometrics
The Bucharest Academy of Economic Studies
Lecturer Radu R. SERBAN, PhD
"Spiru Haret" University

ALGORITHM WITH RANDOM CHARACTER FOR NONLINEAR OPTIMIZATION

Abstract.*In this paper an algorithm for solving optimization problems is proposed. The algorithm, which follows in principle the general scheme of search algorithm has a strong random character. Therefore, the search directions are vectors having a density of repartition with uniextremal character (in this case, the multidimensional normal law) the algorithm steps are given and the numerical results are presented for nine test problems known in the literature.*

Key words: *Nonlinear optimization problem; random search techniques; density of repartition.*
JEL classification: C 61

1. NONLINEAR OPTIMIZATION PROBLEM

The nonlinear optimization problem:

$$\min f(X) \text{ with the restrictions} \tag{1}$$
$$g_j(X) \leq 0, \; j = 1,,......,m$$

where f *şi* g_j are defined continuous functions $\Re^n$.

We will consider

$$D = \{X \in \Re^n \,/\, g_j(X) \leq 0, j = 1,.....,m\} \tag{2}$$

the set of admissible solutions of problem (1)

The algorithm presented is an algorithm with random character. The general scheme of the algorithm is inspired from a combination of two algorithms of multidimensional optimization, that is Baniciuc algorithm and Hook and Revees algorithm.

As a main description, like in case of any other search algorithm, we start from a certain iteration $k(k = 0,1.2,...)$ from an admissible point $X_k \in D$. We will note $X_k = (x_{k_1}, x_{k_2}, ..., x_{k_n})$. In Baniciuc algorithm, as known, the search for a new point (better as regards the optimum) is done on the direction of the axes coordinated from $\Re^n$.

In the algorithm presented in what follows, we propose for the search direction $d \in \Re^n$ to be a random **vector, uniformly distributed on the unity n-dimension**

$$d = (d_1, d_2, ..., d_n)$$

If starting from X_k through exploration on direction d_1 (with a given step) a success point is obtained, that is a point the value of function f is less than the value of the starting point, the exploration continues from this point on direction d_2 and so on.

In case the exploration on direction d_1 is unsuccessful, we shall return at the starting point X_k and a search is performed on the direction $-d_1$. If on this direction we will not obtain a success point, we will return again at the starting point X_k and we will continue the exploration on d_2 direction.

In case, after the exploration on all the random directions $d_i = (i = 1,....,n)$, no success point is obtained, this suggests that the optimum can be in the proximity of the starting point X_k. In this case, the search continues in a smaller step.

If at the end of the exploration on the *2n* directions $d \in \Re^n$ and $-d \in \Re^n$ a success point is obtained X_{k+1} (namely $f(X_{k+1} < f(X_k))$ there are several possibilities to continue, until, a stop criteria is obtained.

One of these methods, following the scheme of Baniciuc algorithm, is to repeat a new iteration, as per the process described, starting from the new point X_{k+1}.

Further on, considering that the direction $\bar{d} = X_{k+1} - X_k$ is a "success direction", as it is defined by Hooke and Jeeves algorithm, we propose the following continuation.

On the success direction $\bar{d} = X_{k+1} - X_k$, starting obviously from the success point X_{k+1} a new exploration is done. We propose that the search direction to be, as until now, distributed uniformly on the sphere of unity and dimension. Another alternative is to explore from X_{k+1} on the coordinates of $\Re^n$.

When a stop criterion is fulfilled and the last point obtained is considered the optimum point.

2. THE STEPS OF THE ALGORITHM

We know:

$X_0 \in D$ initial starting point;

$F = f(X_0)$ the value of the efficiency function in X_0;

$\varepsilon > 0$ required accuracy;

N the maximum number of generations at an iteration;

$p_0 = (p_{o_1}, p_{o_2}, ..., p_{0_n})$ the initial vector of the explorations steps;

$P = (P_1, P_2, ..., P_n)$ the vector of the exploration steps on the vector "success directions";

PP>0 represents the penalty coefficient.

Let us assume that at an iteration $k = (k = 0,1,2...)$ X_k point is known and the corresponding value of the efficiency function:

$$F = f(X_k) + PP\sum_{j=1}^{m} g_j(X_k)(1 + signg_j(X_K)) \tag{3}$$

We will point out here that the value of F function calculated with formula (3) is "penalized", in the sense of the optimum if point $X_k \notin D$.

Then, the description of algorithm steps is the following:

Step 1. $m = 1$ (the contour of the number of generations by directions)

Step 2. The search directions is calculated

$d^m = (d_1^m, d_2^m, ..., d_n^m)$

Which is a random vector distributed uniformly on the sphere of *n*-dimensional unity $|\,|$.

Step 3. We calculate

$x_{k+1}, i = x_{ki} + p_{ki} d_{ki}^m$, $i = 1,2,...,n$

and though we obtain the point

$$XX_{k+1} = (x_{k+1,1}, x_{k+1,2}, ..., x_{k+1,n})$$

so

$FF = f(XX_{K+1})$, calculated with formula (3).

Step 4. If $FF < F$ then $X_{k+1} = XX_{k+1}$ and jump at Step 11.

If $FF \geq F$, jump at Step 2.

Step 5. We come back at the initial point X_k and we determine the new point

$XX_{k+1} = (x_{k+1}, x_{k+1,2}, ..., x_{k+1,n})$ (with the same steps) on the directions $-d_{ki}^{m}$, that is

$$x_{k+1}, i = x_{ki} - p_{ki} d_{ki}^{m},\ i = 1,2,...,n\,.$$

We will note the same

$$FF = f(XX_{K+1})\,.$$

Step 6.

If $FF < F$ then $X_{k+1} = XX_{k+1}$ and jump at step 11.

If $FF \geq F$, jump at step 7.

Step 7.

The exploration on d^{m} direction was unsuccessful, we come back at the initial point X_k, that is

$$X_{k+1} = X_K$$

And jump at Step 8.

Step 8.

If $m < N$, then $m = m + 1$ and jump at Step 2.

If $m = N$, jump at Step 9.

Step 9.

We will consider

$NP = \left(\sum_{i=1}^{n} p_{ki}^{2} \right)^{\frac{1}{2}}$ the norm of the vector of the exploration steps at the iteration *k*.

If $NP < \varepsilon$, jump at step 10.

If $NP \geq \varepsilon$ the exploration was not successful and we will diminish the steps for a search in the proximity of point X_k.

For example, $P_k := \frac{1}{2} P_k$ and jump at step 1.

Step 10.

The exploration on the success direction $\bar{d} = X_{k+1} - X_k$,

$d = (\overline{d}_1, \overline{d}_2, ..., \overline{d}_n)$ with one step $P = (P_1, P_2, ...P_n)$ where, for example,

$$P_i = \alpha p_{ki},\ i = 1,2,...,n$$

and the multiplication factor is $\alpha \geq 2$.

$$x_{0i} = x_{k+1}, i + P_i \overline{d}_i,\ i = 1,2,...,n$$

We will note:

$X_k = (x_{01}, x_{02}, ..., x_{0n})$ jump at step 1.

Step 11.

We will consider $ER = \left[\sum_{i=1}^{n} \left(x_{k+1,i} - x_{ki} \right)^2 \right]^{\frac{1}{2}}$

If $ER < \varepsilon$, jump at step 12.
If $ER \geq \varepsilon$, jump at step 1.

Step 12.

Optimum point

$$X^* = X_{k+1}$$

And the value of the efficiency function

$F = f\left(X^*\right)$

Stop.

Finally, we will make some remarks regarding the proposed algorithm.

In general, these types of algorithms, with stochastic character are cited as being extremely robust in the exploitation.

Moreover, given the big number of random generations, in many cases the algorithm presented has the chance of exploring and obtaining the global optimum of problem (1).

We have to mention, in the end, that the proposed algorithm can be successfully used for determining the initial admissible point $X_0 \in D$.

3. NUMERICAL RESULTS

There have been used nine test problems known in the literature as the problem of Lawler and Belle.

No.	Problem	Number of variables	Number of restrictions	Objective function	Type of restrictions	Value of the function in the optimum point $f(x^*)$
1	LB1	5	11	$f = \sum_{i=1}^{5} x_i^2$	Linear	7.44
No.	Problem	Number of variables	Number of restrictions	Objective function	Type of restrictions	Value of the function in the optimum point $f(x^*)$

2	LB2	7	14	$f = x_1x_7 + 3x_2x$ $+ x_3x_5 + 7x_4$	10 linear 2 squared 2 non convex	0.00
3	LB3	7	14	$f = \sum_{i=1}^{7} x_i$	10 linear 2 squared 2 non convex	7.33
4	LB4	7	14	$f = \sum_{i=1}^{7} \sum_{j=1}^{7} x_i x_j$ $-\sum_{i=1}^{7} x_i^2$	10 linear 2 squared 2 non convex	16.00
5	LB5	7	14	$f = \sum_{i=1}^{7} x_i^2$	10 linear 2 squared 2 non convex	13.68
6	LB7	8	15	$f = 3x_1x_1 + x_2 +$ $+ x_2x_5 + x_2x_7 +$ $+ 6x_3x_8 +$ $+ x_5x_7 + 11x_6$	10 linear 2 square 2 non convex	9.47
7	LB8	8	15	$f = \sum_{i=1}^{8} x_i^2$	11 linear 2 squared 2 non convex	39.20
8.	LB9	8	15	$f = \sum_{i=1}^{8} x_i$	11 linear 2 squared 2 non convex	12.65
9.	LB10	8	15	$f = x_1x_2 + x_3 +$ $+ x_1x_4x_5 +$ $+ x_2x_4x_6 +$ $+ x_2x_5x_7 +$ $+ x_6x_7x_8$	11 linear 2 square 2 non convex	0.00

The calculations have been performed with $\varepsilon = 0.1$ accuracy. The initial vector of the exploration steps p_0 had all components equal to one and the vector P of the exploration steps on the "success directions" had three times bigger components than the components of p_0 vector.

We will note:
$N = 3$ (the maximum number of generations at an iteration);
k – represents the number of iterations;

NF – represents the number of evaluation of efficiency functions.

The results obtained are presented in the table bellow:

No.	Problem	Number of k iterations	Number of NF evaluations	Value of function $f(x^*)$	Observations
1.	LB1	105	281	7.499	Global optimum
2.	LB2	138	462	0.080	Global optimum
3.	LB3	122	401	7.291	Global optimum
4.	LB4	159	571	16.081	Global optimum
5.	LB5	182	701	14.559	Global optimum
6.	LB7	179	680	9.507	Global optimum
7.	LB8	207	844	42.003	Global optimum
8.	LB9	188	785	12.699	Global optimum
9.	LB10	235	1027	0.098	Global optimum

From the tests performed we can observe the fact that the algorithm presented is very stable. As mentioned in the literature, in most of the cases the global optimum of the problem is obtained.

REFERENCES

[1] **Badescu, Andrei, Badescu, Adrian (2007),** ***Saddlepoint Approximations for Ruin Probabilities.*** Journal of Economic Computation and Economic Cybernetics Studies and Research, issue 3-4, ASE Publishing House, Bucharest;

[2] **Lawler, E.L., Belle, M.D. (1982),** ***A method for Solving Discrete Optimization Problems***, Operations Research 14, 1098-1112;

[3] **Mohan, C., Shanker, K. (1990),** ***Computational Algorithms Based on Random Search for Solving Global Optimization Problems***; International Journal of Computer Mathematics, Volume 33, Issue 1-2, pages 115 – 126;

[4] **Price, W.L. (1983),** ***Global Optimization by Controlled Random Search***. Journal Optimization Theory Applied. Volume 40, 333-348;

[5]**Ruxanda Gh., Botezatu, A. (2008),** ***Spurious Regression and Cointegration. Numerical Example: Romania's M2 Money Demand;*** Romanian Journal of Economic Forecasting, no. 3, Bucharest;

[6]**Şerban, R, (1999),** ***Optimizare cu aplicații în economie*** **(Optimization with Applications in Economy),** MATRIXROM Publishing House, Bucharest;

[7] **Wenzu Sun, Za-Xiang Zuan (2006),** ***Optimization Theory and Methods. Nonlinear Programming***. Springer.

[8] **Gurievici, I.B., Zaharenco, V.G., Pocitman, I.M. (1989),** ***Randomirovanii algoritm dlia resenia zadaci nelineinovo programirovania.*** Tehniceskaia Kibernetica, Kiev, no.5.