Multidimensional Optimization Algorithms Numerical Results

Radu SERBAN, Mircea IULIAN, Radu R. SERBAN
Academy of Economic Studies, Bucharest

This paper presents some multidimensional optimization algorithms. By using the "penalty function" method, these algorithms are used to solving an entire class of economic optimization problems. Comparative numerical results of certain new multidimensional optimization algorithms for solving some test problems known on literature are shown.
Keywords: *optimization algorithm, multidimensional optimization, penalty function.*

Introduction

Let us consider a nonlinear optimization model under general form:

$$\begin{cases} \min f(x), & \text{with constraints} \\ g_j(x) \le 0, & j = 1, 2, \ldots, m \end{cases} \quad (1)$$

where $f, g_j : \Re^n \to \Re$, $j = 1, 2, \ldots, m$ are continuous real functions of n real variables. If the constraints are equalities and functions f and g_j, $j = 1, 2, \ldots, m$ are derivable, then we can use the well – known method of Lagrange multipliers.

For more general cases, the solving of an optimization problem with constraints as inequalities, as it is the case of problem (1), can be reduced to solving a sequence of optimization problems of an unconstrained function.

An interesting method to solve problem (1), subjecting the efficiency function f and constraints g_j , $j = 1, 2, \ldots, m$ to certain conditions, is the SUMT method. The methods cover many applications, known also under the name of penalty methods. The first results are due to Fiacco and McCormick [2].

The idea of the method consists in transforming the constrained optimization problem (1) into an unconstrained optimization problem of a function F ("aggregate" function). Function F is constructed so that the efficiency function f to be penalized (in the optimum sense) by the fact that any point $x \in \Re^n$ does not satisfy the constraints g_j , $j = 1, 2, \ldots m$. Denoting by M the feasible solutions domain of the constrained problem (1),

$$M = \{x \in \Re^n / g_j(x) \le 0, j = 1, 2, \ldots, m\}$$

The optimization problem (1) can be written: (PR) $\min\{f(x) / x \in M\}$, called the "constrained problem" . The problem (PR) is replaced with a sequence of minimization problems of an unconstrained function (penalty function), depending on a parameter: $(\text{PFR})_t$ $\quad F(x,t) = f(x) + \frac{1}{t} p(x)$. Let: $F(x,t) = \alpha(t)$ (2) be the solution to $(\text{PFR})_t$ problem (obviously, an unconstrained optimization problem, called also multidimensional optimization problem).

The penalty function "p" is chosen so that the solution to problem (2) to converge to solution of problem (1) (PR), when parameter t tends to zero, namely: $\lim_{t \to 0} \alpha(t) = F(x^*, t)$

In literature, [2],[5], several forms of the penalty function p are known. The most used ones are: $p(x) = \sum_{j=1}^{m} [\max(g_j(x), 0)]^2$ (3)

or $p(x) = \sum_{j=1}^{m} [\ln(-g_j(x))]$ (4)

By using, for instance (3), the unconstrained multidimensional optimization problem, $(\text{PFR})_t$ becomes: $F(x,t) = f(x) + \frac{1}{t} \sum_{j=1}^{m} [\max(g_j(x), 0)]^2$ (5). According to those mentioned above, we may state that the multidimensional optimization (optimization of a real function of n real variables) is an efficient tool to solving an entire class of nonlinear economic optimization problems.

[1],[3],[4],[5] are some of the most known algorithms of the class of multidimensional optimization algorithms which do not employ the information given by gradient:

- *Cyclic optimization algorithm on coordi-*

nate axes (OCA);
- Hooke and Jeeves' algorithm (HJ);
- Nelder and Mead's algorithm, called also simplex algorithm for multidimensional optimization (NM);
- Rosenbrock's algorithm (RB).

Two variants of the cyclic optimization algorithm on coordinate axes (OCAV1 and OCAV2) are presented in [6]. Two variants of the Hooke and Jeeves' algorithm (HJV1 and HJV2) and a variant of the Rosenbrock's algorithm (RBV1) are presented in [5] and [6].

Comparative numerical results

Further on, we shall present some numerical results for the mentioned algorithms. They have been tested on two test problems known in literature [1], problems which set serious "traps" to numerical solving, namely:

Problem 1 (Rosenbrock's test function) $\min f(x_1,x_2)=100\left(x_2-x_1^2\right)^2+(1-x_1)^2$. The function has the minimum point $x^*=(1,1)$ cu $f\left(x^*\right)=0$ along parabola $x_2=x_1^2$. The initial point is: $x^0=(-1.2,1)$ cu $f\left(x^0\right)=24.2$.

Problem 2 (Witte and Holst's test function) $\min f(x_1,x_2)=100\left(x_2-x_1^3\right)^2+(1-x_1)^2$. The function has the minimum point $x^*=(1,1)$ cu $f\left(x^*\right)=0$. The initial point is $x^0=(-1.2,\ 1)$ with $f\left(x^0\right)=749.037$.

Problem 3 (Beale's test function) $\min f(x_1,x_2)=\sum_{i=1}^{3}\left[c_i-x_1(1-x_2^i)\right]$, where: $c_1=1.5$; $c_2=2.25$; $c_3=2.625$

The initial point $x^0=(0\,,0)$ and the optimum point $x^*=(3\,,0.5)$.

Accuracy $\varepsilon_1=0,1$ and also $\varepsilon_2=0,01$ were taken into consideration for all problems.

The indicator NF represents the evaluations number of function f.

i) The results obtained for Rosenbrock's function are the following:

Cyclic optimization algorithm on axes:

Accuracy	Optimization on axes (OCA)	Variant 1 (OCAV1)	Variant 2 (OCAV2)
$\varepsilon_1=0.1$	$\begin{cases}x_1=0.981\\x_2=0.903\end{cases}$ *NF* = 1183	$\begin{cases}x_1=0.973\\x_2=0948\end{cases}$ *NF* = 1415	$\begin{cases}x_1=0.964\\x_2=0.988\end{cases}$ *NF* = 1502
$\varepsilon_2=0.01$	*NF* > 5 000	$\begin{cases}x_1=0.990\\x_2=0.992\end{cases}$ *NF* = 3725	$\begin{cases}x_1=0.994\\x_2=0.996\end{cases}$ *NF* = 4253

It was obtained for Hooke and Jeeves' algorithm and suggested variants:

Accuracy	Hooke and Jeeves	HJV1	HJV2
$\varepsilon_1=0.1$	$\begin{cases}x_1=0.989\\x_2=0.901\end{cases}$ *NF* = 114	$\begin{cases}x_1=0.977\\x_2=0.899\end{cases}$ *NF* = 111	$\begin{cases}x_1=0.979\\x_2=0.908\end{cases}$ *NF* = 124
$\varepsilon_2=0.01$	$\begin{cases}x_1=1.042\\x_2=1.099\end{cases}$ *NF* =139	$\begin{cases}x_1=0.999\\x_2=1.009\end{cases}$ *NF* = 137	$\begin{cases}x_1=0.991\\x_2=1.018\end{cases}$ *NF* = 151

It was obtained for Nelder and Mead algorithm and suggested variant:

Accuracy	**Nelder and Mead (NM)**	**VNM**
$\varepsilon_1 = 0.1$	$\begin{cases} x_1 = 0.988 \\ x_2 = 0.902 \end{cases}$ $NF = 131$	$\begin{cases} x_1 = 0.991 \\ x_2 = 0.911 \end{cases}$ $NF = 119$
$\varepsilon_2 = 0.01$	$\begin{cases} x_1 = 1.012 \\ x_2 = 1.009 \end{cases}$ $NF = 219$	$\begin{cases} x_1 = 9.998 \\ x_2 = 1.010 \end{cases}$ $NF = 200$

It was obtained for Rosenbrock's algorithm and suggested variant:

Accuracy	**Rosenbrock (RB)**	**RBV1**
$\varepsilon_1 = 0.1$	$x_1 = 0.975$ $x_2 = 0.913$ $NF = 349$	$x_1 = 0.983$ $x_2 = 0.951$ $NF = 299$
$\varepsilon_2 = 0.01$	$x_1 = 0.987$ $x_2 = 0.989$ $NF = 395$	$x_1 = 0.998$ $x_2 = 0.993$ $NF = 355$

ii) The results obtained for Witte and Holst's problem (problem 2)
Cyclic optimization algorithm on axes

Accuracy	**Optimization on axes (OCA)**	**Variant 1 (OCAV1)**	**Variant 2 (OCAV2)**
$\varepsilon_1 = 0.1$	$\begin{cases} x_1 = 0.918 \\ x_2 = 0.902 \end{cases}$ $NF = 40$	$\begin{cases} x_1 = 0.939 \\ x_2 = 0.927 \end{cases}$ $NF = 58$	$\begin{cases} x_1 = 0.949 \\ x_2 = 0.988 \end{cases}$ $NF = 76$
$\varepsilon_2 = 0.01$	$\begin{cases} x_1 = 0.998 \\ x_2 = 0.991 \end{cases}$ $NF = 140$	$\begin{cases} x_1 = 0.991 \\ x_2 = 0.994 \end{cases}$ $NF = 183$	$\begin{cases} x_1 = 0.999 \\ x_2 = 1.009 \end{cases}$ $NF = 198$

Hooke and Jeeves' algorithm

Accuracy	**HJ**	**HJV1**	**HJV2**
$\varepsilon_1 = 0.1$	$\begin{cases} x_1 = 0.901 \\ x_2 = 0.972 \end{cases}$ $NF = 312$	$\begin{cases} x_1 = 0.912 \\ x_2 = 0.968 \end{cases}$ $NF = 310$	$\begin{cases} x_1 = 0.929 \\ x_2 = 0.970 \end{cases}$ $NF = 328$
$\varepsilon_2 = 0.01$	$\begin{cases} x_1 = 0.997 \\ x_2 = 1.007 \end{cases}$ $NF = 446$	$\begin{cases} x_1 = 0.995 \\ x_2 = 0.995 \end{cases}$ $NF = 426$	$\begin{cases} x_1 = 0.991 \\ x_2 = 1.009 \end{cases}$ $NF = 450$

Nelder – Mead's algorithm

Accurracy	**NM**	**VNM**
$\varepsilon_1 = 0.1$	$\begin{cases} x_1 = 1.099 \\ x_2 = 0.897 \end{cases}$ *NF* = 1545	$\begin{cases} x_1 = 1.094 \\ x_2 = 0.899 \end{cases}$ *NF* =1600
$\varepsilon_2 = 0.01$	$\begin{cases} x_1 = 1.008 \\ x_2 = 0.991 \end{cases}$ *NF* = 2176	$\begin{cases} x_1 = 1.000 \\ x_2 = 0.999 \end{cases}$ *NF* = 2202

Accurracy	**Rosenbrock (RB)**	**RBV1**
$\varepsilon_1 = 0.1$	$x_1 = 1.089$ $x_2 = 0.899$ *NF* = 196	$x_1 = 1.099$ $x_2 = 0.911$ *NF* =191
$\varepsilon_2 = 0.01$	$x_1 = 1.009$ $x_2 = 0.992$ *NF* = 2264	$x_1 = 1.007$ $x_2 = 0.998$ *NF* = 245

Further on, we present the results obtained using Rosenbrock's algorithm for Beale's test function.

Accurracy	**Rosenbrock (RB)**	**RBV1**
$\varepsilon_1 = 0.1$	$x_1 = 2.900$ $x_2 = 0.599$ *NF* = 60	$x_1 = 2.931$ $x_2 = 0.580$ *NF* = 60
$\varepsilon_2 = 0.01$	$x_1 = 2.989$ $x_2 = 0.509$ *NF* = 101	$x_1 = 2.993$ $x_2 = 0.506$ *NF* = 97

From an experimental point of view, we may state that the suggested variants are stable and generally use a smaller number of function evaluations to get the required accuracy.

References

[1] **Dumitru, V**., (1975) - *Programare neliniara*, Appendix P., in cooperation with Fl. Luban, S. Moga, R. Serban, Ed. Academiei R.S.R. , Bucharest;

[2] **Fiacco, A.V.**, **McCormick** (1968) - *Nonlinear Programming: Sequential Unconstrained Minimization Technique*, John Wiley, N.Y;

[3] **Karmanov, V.G.** (1975) - *Matematiscescoe programmirovanie*, Izd. Nauka, Moscova;

[4] **Panteleev, A.V.**, **Lemova** (2002) - T.A., *Metodi optimizatii v primerah i zadaciah*, Ed. Visaia Scola, Moskova;

[5] **Serban, R.** (1999) - *Optimizare cu aplicatii in economie*, Ed. MATRIXROM, Bucharest;

[6] **Serban, R.R.** (2005) - *Optimizarea deciziei firmei*, Doctor's degree thesis, ASE Bucharest.