1. INTRODUCTION

Optimizational approach to solving queries of complex systems synthesis is in itself a big reserve for elevating quality of planning, management and projecting. The choice of optimizational aim areas of changing parameters is the task of particular economic and technical branches of science. What concerns the optimization mechanism is the subject of mathematical programming.

The success of linear programming in increasing effectiveness of economic modeling and optimization of planning is well known. It is less obvious in technique, management and projecting as for added accuracy of above mentioned it is important to take into consideration nonlinear effects. Creation of "simplex method" and appearance of powerful PCs made linear programming an important tool for solving different problems but at the same time showed its weakness.\(^1\) Most of the queries cannot be adequately solved by linear programming model because they include nonlinear goal functions or constraints. All the above enumerated attracted special attention of mathematicians to the progress in nonlinear programming research.

It is important to mention that many nonlinear optimizational queries that exist in economy and technique are described as including convex or concave functional and convex areas of possible solutions parameters (for more information see Casten theorem). In literature on the subject, attention is mainly attracted to convex programming. The reason is the possibility to produce universal method for solving basic form equations as well as others with deviations from the basic form.\(^2\) One cannot guarantee the same for other nonlinear equations with wider range of parameters. Let me leave aside the prior and talk more about the later, as it is the subject of my research.

2. DESCRIPTION

Before I turn to the description of my research, let me first speculate a little on main definitions of nonlinear equations. These are tasks where two qualifications are relaxed: dividing and adding, which means that goal function and constraints might be nonlinear and variables can take values from some multitude, including discrete multitude. To the above mentioned problems of mathematical programming, refer the following:

- Tasks of integer programming with linear and nonlinear goal functions and constraint functions;
- Combined models of integer programming problems with linear and nonlinear goal functions and constraint functions;
- Problems of discrete programming, in which the value of variables is chosen out of given value multitude of rational numbers, not necessarily integer, also with linear and nonlinear goal function and constraint function;
- Problems of nonlinear programming with convex and concave functions;
- Problems of nonlinear programming with multiple-optima functions on non-convex or/and convex or/and non-linked zone of variables;
- Solving of equations and systems of equations, with functions that I identified above.

A large number of books are devoted to the problem of solving above-mentioned queries. It is enough to tell that G. Vagner's "Basic operational research", published in 1973, gives references to more than 162 publications, "Nonlinear programming", by Bazara and Shetti, published in 1982 sources 600 books. Literature on the subject is not of course bound to what was brought up here. The reason for such an attention lies in the provisions of practice. The absence of effective and quick method for finding global optimum in such problems leads to artificial simplification of mathematical model in practice. As a consequence system characteristics in economy, finance, technique, chemistry, physics, architecture, etc deteriorate.

To my knowledge, there is such an opinion among specialists that to create a universal method for

---

\(^1\) Mokhtar S. Bazaraa, C.M. Shetty, "Nonlinear programming, Theory and Algorithms", John Wiley and Sons, New York, 1979, p.8

solving nonlinear problems, like Simplex -method that is used for linear queries, is impossible and the only way out are specialized methods for each particular type of programming problem. I hope now I can prove this information not up-to-date. Application of specialized algorithms requires exact correspondence of real model and problems that can be solved with the help of this particular algorithm. In case of any deviations you have to change it usually by simplifying mathematical model to the requirements of algorithm. It is important to mention also that initially the reality was already simplified in the model (in order to make it equal to the needs of an algorithm). As a result you answer the question that was not asked by the practice. In order to meet the requirements of imperfect mathematical method we replace the model of real natural conditions by the picture that either you or algorithm found suitable. Such fractional results are inapplicable for problems of integer programming because this data is used for forming planned decisions in complicated situations. Such problems with nonlinear functions are important for non-proportional fluctuation of expenses, determination of productivity value, quality assurance, in technical and economic addendum, in the sphere of nonlinear physical laws and others.

To begin with the description of my research, I want to insist on the fact that I created a unique and general method for solving non-linear queries of any level of complication. Queries that this method is solving are used in calculation of optimization problems and solution of nonlinear equations, for such areas as Modeling and Simulation, Operations Management, Machine Vision, Robotics & Automation. Using my method I successfully solved test queries given by Wolf, H.H.Rosenbrock and J.D. Powell, as well as different sets of non-linear equations. Using it, I am capable of solving quite complicated non-linear optimization problems of the classic form:

\[\begin{align*}
\text{min } & F(X) \\
\text{subject to: } & Wi(X) \leq 0, \ i = 1, \ldots, m_1, \\
& Gi(X) = 0, \ i = 1, \ldots, m_2, \\
\end{align*}\]

Where X is an n-vector, and the functions F, Wi and Gi should not have gaps and can be linear, nonlinear, multi-extreme, convex and non-convex. My universal method of finding optimal solution for the above- enumerated programming queries consist of two well-known and simple algorithms: method of proportionally deformed polyhedron and method of gradient descending. It is well known that these two methods usually produce only local, separate solutions. But it is not true for my technique. In the algorithm these simple methods let me always acquire global solutions. Obviously there is no magic in it. The results obtained by me (proved theoretically) are the effects of extending the theory of convex functions. My algorithm was checked during two years of investigations and illustrated by solving of several hundred, better to say thousand of, tests and real nonlinear equations (not all of them were included in this work as examples) and also applied theoretically. My algorithm meets all the requirements that I imposed before creating it. I was very thorough while testing it. My testing included the following steps:

- I solved all test equations that I could find in the literature on the subject.
- All real problems from the literature were solved too.
- Successfully tested it on several hundred self-created equations with number of variables starting from 1 up to 500
- Solved equations with implanted linear and nonlinear constraints, including discrete constraints on variables
- Solved several sets of equations for clients via e-mail

Results obtained were always satisfactory which means that algorithm was able to find global optimum for multi-extreme function, as I was only calculating that type of function. Program that embodied my method and that can also be used, as subprogram is in FORTRAN-77, MS DOS. Resolution time depends on the type of a problem or number, and complexity of functions. For Pentium 200, I did not find an equation for which resolution time was longer than an hour. In my practice programming, a query with 100 variables was solved within 15 minutes. Algorithm stops its search when either global optimum is found, or a particular number of iterations is achieved.

### 3. COMPARISON

It is worth mentioning that from the beginning of realization of existence of both local and global optimum or optimums, we find information on algorithms for their location. It is hard for me to make any connections or comparison of my method with others. In accordance with the information of Northwestern University and the Argonne National Laboratory of in USA, such a method of mine does not exist. Scientists consider it unrealistic to expect to find one general nonlinear optimization code that is going to work for every kind of nonlinear model. Instead one should try to select a code that fits the problem one is solving.\(^3\)

I will try here to bring about possible comparison points. It is feasible to relate solution time of my algorithm and others. I was comparing my algorithm to stochastic search code by Georgian professor Chichenadze. Test task # 14 was solved within 14 seconds while it took Chichenadze 10 hours.  

Modified method of nonlinear and stochastic optimization systems based on ideas of Professors L. Ingber and J. Powel allows to make solution more precise and finds global optimum with set before probability but it does not guarantee theoretically global solution. For this algorithm there will emerge a counterexample that cannot be calculated by it. This cannot be said about my algorithm which finds global optimum every time. From the calculation point of view it does not have common error - the rounding default of the results accuracy. Error is equal to the error of optimum calculation in optimal point, which is determined by the chosen software and type of the computer. However I should add that any universal algorithm (and mine is not an exception) will not be as good for solving some particular problems as algorithm specifically created for them. To my knowledge there is no other algorithm with such broad universality as mine.

4. EXAMPLES

Computing examples (numbered N_1 to N_14) using the proposed algorithm are presented below to show the efficacy of the method.

**N_1. Equation with multi-extreme goal function.**

Minimize:

\[
EE_1 = \exp[\sin(10^{\times}X_4)\times\sin(123+X_4^{0.73}+3\times X_4^{2}+X_4^{1.55})]
\]

where:

\[
EE_2 = \left|\left(\left[X_1^{0.9}+X_2^{0.8}\right]\left[1.1+\cos(X_3^{2.3}+X_4^{1.3})\right]\right) -3\right| < 0
\]

\[
EE_3 = \left|\left(\left[X_1^{0.9}+X_2^{0.8}\right]\left[1.1+\cos(X_3^{2.3}+X_4^{1.3})\right]\right)\right| = 0
\]

Solution:

\[
EE_1 = 0.3682, \; X_1 = 1.9484, \; X_2 = 1.5357, \; X_3 = 2.6601, \; X_4 = 1.445, \; EE_2 = -0.3259, \; EE_3 = 0.00
\]

**N_2. Equation with multi-extreme goal function.**

Minimize:

\[
EE_2 = 3(\sin(4(3+X_4)))^2+(\sin(4+X_4))^{12}
\]

where:

\[
EE_3 = -X_1.
\]

Solution:

\[
EE_2 = 0.0061823, \; X_1 = 7.995.
\]

Control:

\[
EE_2 = 0.00150968, \; X_1 = 8.00
\]

**N_3. Equation with multi-extreme goal function.**

Minimize:

\[
EE_1 = \sin(X_4+\sin(X_1)+4) / (10+\exp(\sin(33X_4))))
\]

where:

\[
\begin{align*}
EE_1 &= -0.3259, \\
EE_2 &= 0.00, \; X_1 = 1.445, \; X_2 = 1.5357, \; X_3 = 2.6601.
\end{align*}
\]

**N_4. Equation with multi-extreme goal function (function of Rozenbrok).**

Minimize:

\[
EE_2 = 100(X_2-X_1^2)^2+(1-X_1)^2
\]

where:

\[
\begin{align*}
EE_1 &= 0.00, \; X_1 = 0.999, \; X_2 = 0.999, \; X_3 = 0.999, \; X_4 = 0.999.
\end{align*}
\]

**N_5. Equation with multi-extreme goal function (function of Powell).**

Minimize:

\[
EE_2 = \sum_{i=1}^{100}(x_i^2+10\sin(3.1416*3/2/iX_i))
\]

where:

\[
\begin{align*}
EE_1 &= 99.99999999, \; X_1 = 2.999, \; X_2 = 2.999, \; X_3 = 2.999, \; X_4 = 2.999, \; X_5 = 2.999.
\end{align*}
\]

**N_6. Equation with multi-extreme goal function.**

Minimize:

\[
EE_2 = 0.00, \; X_1 = 0.00, \; X_2 = 0.00, \; X_3 = 0.00, \; X_4 = 0.00
\]

**N_7. Equation with multi-extreme goal function and non-convex and non-linked goal function.**

Minimize:

\[
EE_2 = \cos(3.22X_1)+1
\]

---

**N. 8. Equation with multi-extreme goal function and non-convex and non-linked goal function**

Minimize:

\[ E_E = 0.001((X_1-10)(X_1-1)(X_1-2)(X_1-3)^6 \right) \]
\[ + (X_1-4)(X_1-5)(X_1-6)(X_1-7)(X_1-8)(X_1-9)+ \]
\[ \cos(17X_1)(X_1-5)(X_1-6) \]
\[ \text{where} \ (<=0) \]
\[ E_E = (\sin(X_1+X_2)) \]

Solution:

\[ E_E = 4.29346, \ X_1 = 1.29246267, \ X_2 = 11.2795776, \]
\[ E_E = 0.0000321 \]

**N. 9. Equation with multi-extreme goal function.**

Minimize:

\[ E_E = (X_1-22)^2 + (\sin(13X_1))-3 \]
\[ \text{where} \ (<=0) \]
\[ E_E = X_1 \]

Solution:

\[ E_E = 0.00022543, \ X_1 = 21.98646728 \]

Control:

\[ E_E = 0.000354, \ X_1 = 21.99 \]

**N. 10. Equation with multi-extreme goal function.**

Minimize:

\[ E_E = 22\sin(0.001X_1)(X_1- \]
\[ 5)^2((1(X_1+4))(\sin(13X_1)+\cos(20X_1)) \]
\[ \text{where} \ (<=0) \]
\[ E_E = X_1 \]

Solution:

\[ E_E = 35.2194, \ X_1 = 8.6238524 \]

Control:

\[ E_E = 35.143360, \ X_1 = 8.62 \]

**N. 11. Equation with multi-extreme goal function.**

Minimize:

\[ E_E = 3(\sin(4X_1))^2 + (\sin(4+X_1)) \]
\[ +22\sin(0.01X_1)(X_1-5)^2((1(X_1+4))^6 + \]
\[ (\sin(13X_1)+\cos(20X_1)) + 100(X_5+X_6)^2+(1- \]
\[ X_3)^2+X_5+10X_6)^2+5(X_9+X_10)^2 \]
\[ \text{where} \]
\[ X_1, X_2, \ldots, X_10 > 0 \]

Solution:

\[ E_E = -3.07, X_1 = 1.3200E-001 \]
\[ X_2 = 22.20, X_3 = 16.035, X_4 = 8.6238, X_5 = 1.030, \]
\[ X_6 = 1.0622 \]
\[ X_7 = 4.8579E-006, X_8 = 4.65E-009, X_9 = 5.3838E-003, X_10 = 5.3836E-003 \]

**N. 12. Equation with multi-extreme goal function.**

\[ E_E = (3(\sin(4(3+X_1)))^2 + (\sin(4+X_1)) \]
\[ +5)^2((1(X_1+4))(\sin(13X_1)+\cos(20X_1)) \]
\[ \text{where} \]
\[ X_1 < 0 \]
\[ E_E = X_1 \]

Solution:

\[ E_E = 4.00, X_1 < 0.01 \]

**N. 13. Equation with multi-extreme goal function and non-convex and non-linked goal function.**

Minimize:

\[ E_E = (X_1+1E-3)-X_2 \]
\[ \text{where} \]
\[ X_1 = 1 \]
\[ E_E = (X_1+1E-3)-X_2 \]

Solution:

\[ E_E = 4.6579E-006, X_1 = 1.00 \]


Queries with multi-extreme goal function.

\[ X_1 = \abs((X_1+0.9+X_2^0.8)/(1.1+\cos(X_3^2.3+X_4^2.3))) \]
\[ \text{where} \]
\[ X_1 < 0 \]

Solution:

\[ E_E = 0.3679, X_1 = 1.00 \]