APPROXIMATION OF COMPLEX, MULTIPARAMETER, ESSENTIALLY NONLINEAR, DYNAMIC RELATIONSHIPS BASED ON GENETIC ALGORITHMS

A.O. Glukhov,
Vice professor, PhD,
Polotsk State University, PSU
Polotsk, Belarus,
alexey.glukhov@gmail.com

D.O. Glukhov,
Vice rector, PhD,
Polotsk State University, PSU
Polotsk, Belarus,
d.gluhov@psu.by

V.V. Trofimov,
Chef of Informatics Department, professor,
St. Petersburg State Economics University,
St. Petersburg, Russia,
tww@mail.ru

L.A. Trofimova
professor, Doctor of economics science,
St. Petersburg State Economics University,
St. Petersburg, Russia,
L_Trofimova@bk.ru

***Abstract* - The work is focused on usage of genetic algorithms to get more precise approximation of complex and essentially nonlinear dynamic relationships. The algorithms precision was measured based on the model of stock market index prediction.**

***Ключевые слова*: генетические алгоритмы; сложные, многопараметрические, существенно нелинейные, динамические зависимости; индекс фондовых бирж.**

Taking a genetic approach as a basement an evolutionary algorithm has been developed based on the biological model of evolution of species.

The *creature* is an object that carries a chromosome (code) representing one of possible solutions of a given problem (alternative to [1, 2]). Each chromosome can be translated into a function that takes number of parameters of internal and external environments as arguments [3, 4]. The approach like this is widely used in building solution trees, modeling, solving systems of equation, image/voice recognizing area, etc. [5].

The *Population* is a collection of subpopulations sorted by quality of approximation functions of their chromosomes.

The *Subpopulation* is a set of chromosomes that code function of the same type to overcome local stoppers (local optimums). The *Subpopulation* is a set of chromosomes that code function of the same type to overcome local stoppers (local optimums).

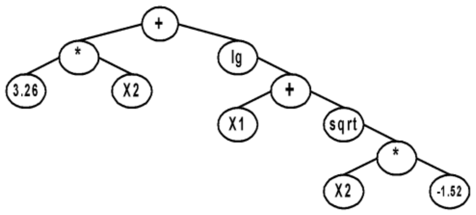
The size of the population is not constant and can varies while removing and adding chromosomes but it does not descend below a predetermined value (minimum number of the chromosomes of the best approximation quality within the population, the chromosomes of that group are never deleted).

The *Chromosome* is an element of the proposed algorithm. It is not constructed as a sequence of genes (as it is in traditional approach) but tree. The nodes of a tree are elements of three major types - operations, variables and constants. Each chromosome codes a function which is a target object of algorithm.

A tree is a convenient form to represent a function. It makes quite easy to perform most popular genetic heuristics like Grow, Crossover and Mutation. Let’s have a look at the example of function representation as a tree [6, 7]. The function example is given below:

$$f(X1, X2) = 3.26 \bullet X2 + \lg(X1 + \sqrt{X2 \bullet (-1.52)})$$

The tree for that function is shown in the pic.1



Pic 1. Example of the tree representation of the function.

The junction nodes in the tree are operations, the leaf nodes are constants and variables.

The algorithm checks the depth of a tree - the maximum number of nodes that can appear on any branch. The depth of a tree is limited by the maximum allowable depth parameter of the algorithm. The depth can be simply calculated by use of a recursive procedure that visits every node of a tree and counts its depth.

The *Operations*. The algorithm solves approximation problems by finding the most suitable approximation function for given values of multiple input and output parameters. A function consists of some operations, variables and constants. Operations are taken from a predefined list. In our example the following operations are used: addition, subtraction, multiplication, division, exhibitor, logarithm, sine, square root. The algorithm picks up operations from that list in order to combine them in an approximation function. The list of allowable operations can be easily expanded.

Quality of approximation (Q) is a criterion driving the selection of creatures within a genetic algorithm. The special fitness-function is used to evaluate Q based on the “distance” between results of approximation done by the creature function (which is coded by chromosome) and real input and output values. We define them formally as follows:

$$X_{11}\ X_{21}\ X_{31}\ X_{41}\ \dots\ X_{M1}\ Y_1$$
$$X_{12}\ X_{22}\ X_{32}\ X_{42}\ \dots\ X_{M2}\ Y_2$$
$$.\quad .\quad .$$
$$X_{1N}\ X_{2N}\ X_{3N}\ X_{4N}\ \dots\ X_{MN}\ Y_N,$$

where X_{ij} ($1 \le i \le M$, $1 \le j \le N$) – the values of input variables (function arguments); N – the number of examples (measurements); M – the number of input variables; Y_j – the result value from the example j.

The “distance” Q is calculated by use of the following formula:

$$Q = \frac{\sum_{i=1}^{N} (F(X_{1i},...,X_{Mi}) - Y_i)^2}{N} = \frac{\sum_{i=1}^{N} \Delta F_i^2}{N}, \quad (1)$$

where Y_j – the result value from the example j; N – the number of examples; M – the number of input variables; $F(X_{1j},...,X_{Mj})$ –the result of approximation.

In order to accelerate the search process, the algorithm uses a shift operator that corrects functions by shifting their results along the Y axis on the Const value. $F = f + Const$, where f is an initial result of approximation, F is a corrected function. The Const value is calculated as the average deviation of the initial calculated values of the approximation function from the example values with the sign "minus".

$$Const = -\frac{\sum_{i=1}^{N} (f(X_{1i},...,X_{Mi}) - Y_i)}{N} = -\frac{\sum_{i=1}^{N} \Delta f_i}{N},$$

where $X_{1i},\ldots, X_{Mi}$ – the values of input variables; N – number of examples; M – number of input variables; Y_i - result value from the example i.

Main heuristics. In the evolutionary algorithm the heuristic is an action resulted in improvement of populations of creatures in terms of their quality (quality of approximation in this particular case). Heuristics can be applied to a single creature as well as to the whole population. The set of heuristics is the important part of the evolutionary algorithm since they are responsible for creation of new creatures (chromosomes). They renew the population in generation by generation manner. A well-chosen set of heuristics affect the efficiency of the algorithm. We also can choose a percentage of usage of each heuristic in the set and that is additional possibility to tune the algorithm efficiency.

The set of heuristics of the algorithm is listed below: 1) Random creation; 2) Mutation of junction nodes; 3) Mutation of constants; 4) Chromosome growing; 5) Chromosome recombination; 6) Function shifting; 7) Expansion of population.

The heuristics 1, 2, 3, 4, 5, 6 work with single creatures modifying or building its chromosome whereas heuristic 7 deals with the whole population.

The heuristics working with single creatures can be divided into classes in some ways:

1)by the scale of chromosome changes: local heuristics (mutation of nodes) and global heuristics (random creation, global mutation of constants);

2)by the type of affected nodes: only constants (mutation of constants), nodes of any types (random generation, mutation of nodes, growing, recombination).

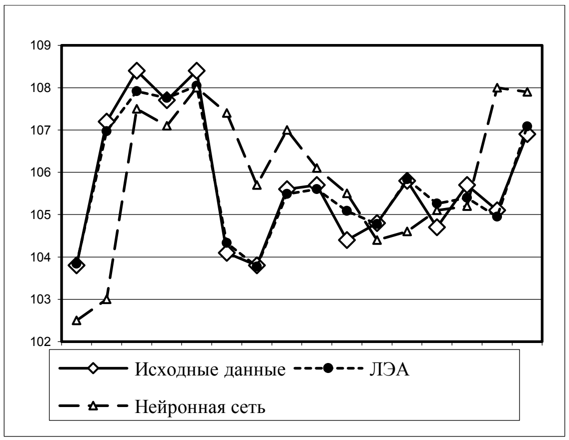
The algorithm uses a random generator to choose the place (node) where to apply a heuristic. However, it takes into account some necessary limitations.

Using mostly local heuristics we called the algorithm LEA (Local Evolutionary Algorithm).

The algorithm has been probated in making a forecast for the stock exchange indices.

Having statistics of IBM's stock price, we tried to make a forecast for the nearest future. The statistics covered a few weeks of price change. So the approximation aimed to get the best fit of the price for the last week (desired forecast) based on input values collected for all previous weeks (known points).

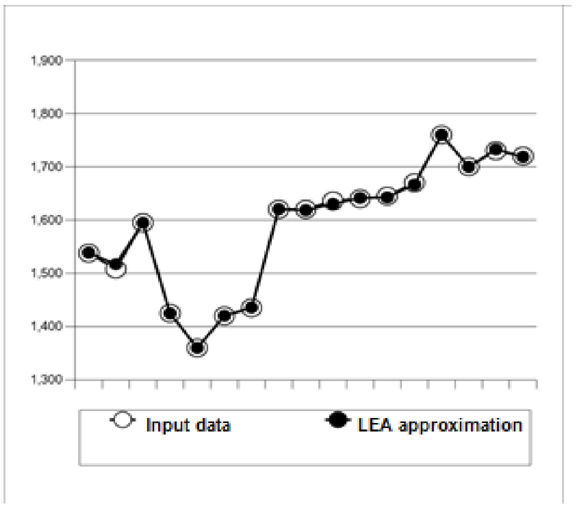
The forecast done that way assumes that the character of changing all influencing factors persists. The approximation quality comparison is illustrated in the picture below. It shows difference between neural network results and results of the proposed algorithm (LEA).



Pic 2. Comparison of the neural network approximation quality and the proposed algorithm (LEA) approximation quality.

As we see it the proposed algorithm gains in quality of approximation comparing to the neural network.

The next example is about the approximation of NASDAQ stock exchange index for six weeks long period (pic.3).



Pic.3. Approximation of NASDAQ stock exchange index

As we see it the LEA gives pretty good fitting to the real index values.

Conclusion

Different methods of planning and forecasting are often used in decision making and management areas. The most problems of those areas can be characterized as approximation of complex, multiparameter, essentially nonlinear dynamic relationships.

The adaptive and self-organizing algorithms is getting more common approaches in solving that sort of problems and neural networks and the proposed LEA algorithm are part of them.

The comparison we made shows that the LEA has advantages over the neural networks of different structures. The LEA is more flexible and gives better results whereas neural networks are not good in approximating the surfaces with large variations in function values.

Both algorithms have disadvantages like time of solution searching which cannot be predicted well and algorithms can stuck at the local extremes with no further notable improvements.

References

1.Трофимова Л.А., Трофимов В.В. Методы принятия управленческих решений: учебник и практикум для академического бакалавриата/Л.А., Трофимова, В.В. Трофимов. – М.: Издательство Юрайт, 2015.– 335 с.

1.Трофимов В.В., Трофимова Е.В. Конвергенция ИТ. Методологические аспекты эволюции ./В.В. Трофимов, Е.В. Трофимова. – Saarbrucken, Deutschland. LAP LAMBERT Academic Publishing, 2014. – 91 с.

3.Глухов А.О., Трофимов В.В. Использование рекурсивных эвристик при решении задач дискретной оптимизации большой размерности // Современные проблемы менеджмента: межвуз. сб. Выпуск 6. – СПб.: Изд-во СПбГУЭФ, 2003. – С. 99-105.

4.Глухов А.О., Трофимов В.В., Глухов Д.О. Локальный генетический алгоритм планирования процесса многопрофильного производства / Экономическая кибернетика: системный анализ в экономике и управлении: Сб. научных трудов. Выпуск №5 – СПб.: СПбГУЭФ, 2002 – С. 50-56.

5.Муттер В.М., Трофимов В.В., Иванова И.В., Калинушкина М.Ю. Математические основы цифровой техники – СПб.: Литера плюс, 1999, 351 с.

6.Trofimov V.V., Hluhov A.O. The Optimal Schedule for the Technological Process of Semiconductor Production. Operations Research 2002: International Conference on Operations Research Sept.2-sept.5, 2002, Klagenfurt, Austria, Klagenfurt University, 2002. – P.148.

7.Трофимов В.В., Глухов А.О. Метод решения задачи идентификации и моделирования самоорганизующихся систем на основе генетического подхода. Международная конференция по мягким вычислениям и измерениям, Санкт-Петербург, 25-27 июня 2001, Сборник докладов. СПб.: ЛЭТИ, 2001, том 1. – с.291-293.

8.Trofimov V.V., Hluhov A.O. Construction of the Optimal Schedule for the Technological process of semiconducting production. 46. International Scientific Colloquium. “Multimedia – The Challenge for Science, Technology and Business”. 24-27.09.2001 Technique University of Ilmenau, Germany 2001.