# Application of a Multiplicative Model with Linear Partial Descriptions in Self-organization Methods

Anatoliy Povoroznyuk[1][0000-0003-2499-2350], Oksana Povoroznyuk[1][0000-0001-7524-5641], Inna Skarga-Bandurova[2][0000-0003-3458-8730]

[1]National Technical University "Kharkiv Polytechnic Institute", Kyrpychova street, 2, Kharkiv, 61002, Ukraine
`ai.povoroznjuk@gmail.com, povoks@i.ua`
2Oxford Brookes University,(School of Engineering, Computing and Mathematics) Wheatley Campus, Oxford, OX33 1HX, UK
`iskarga-bandurova@brookes.ac.uk`

**Abstract.** The problem of constructing regression models is that it is necessary to specify the structure of the model, in addition, in models of large dimensions, poor conditioning of the matrices is possible, which leads to an unstable solution. The paper considers methods of self-organization (methods of group accounting of arguments), which use an iterative procedure for the simultaneous synthesis of the structure of the model and the calculation of its coefficients. The advantages and disadvantages of the known methods of self-organization are analyzed. A self-organization method for the synthesis of a multiplicative model with linear private descriptions has been developed. The effectiveness of the method has been tested on test cases.

**Keywords:** Regression, Methods of self-organization, Private description, Iteration, Multiplicative model, Testing

## 1    Introduction

In the task of identifying objects, the model of the object is presented in the form of a "black box" in which the measured quantities are the vector of input actions $X = \{ x_1,..., x_n \}$ and the output $Y$. Based on the measured values of the inputs and outputs of the object, a model of the object is constructed in the form of some analytical function $Y' = f( X )$ that best approximates the output $Y$ in the best way (in the sense of the minimum of standard deviations) [1, 2].

As $Y' = f( X )$, the regression dependence in the form of the Kolmogorov – Gabor polynomial is usually used:

$$y = P(x_1,\ldots,x_n) = a_0 + \sum_i a_i x + \sum_i \sum_j a_{ij} x_i x_j + \sum_i \sum_j \sum_k a_{ijk} x_i x_j x_k \ldots. \qquad (1)$$

The concrete structure of the model (the number of factors and the maximum degree of the polynomial) is determined by the researcher, and the calculation of the model coefficients from the experimental data is carried out by the least squares method (LSM) [3].

The calculation of the coefficients of a polynomial by LSM is reduced to solving a normal system of linear algebraic equations with respect to unknown coefficients of the polynomial. With a significant number of input factors and an increase in the degree of the polynomial, the number of coefficients in (1) grows like an avalanche, which imposes increased requirements on the volume of the training sample (for the implementation of LSM, the number of experimental points should be significantly larger than the total number of coefficients).

In addition, in real data, as a rule, there are groups of strongly related features. In these conditions, the phenomenon of multicollinearity arises [2], which leads to poor conditioning and, in the extreme case, the degeneracy of the covariance matrix.

Herewith, the solution of a normal system of linear equations is not stable or a solution cannot be obtained.

Therefore, in practice, they are usually limited to linear regression models, although they are inaccurate and are used for a "rough" estimate in order to select the set of influencing factors of model $X$.

For the synthesis of regression models from a small number of experimental data, inductive methods of self-organization (GMDH) are effectively used [4-6], which simultaneously determine polynomials and its coefficients.

## 2    Literature review

Self-organization methods use an iterative procedure for sequentially complicating a polynomial with choosing the best solutions at each step of the iteration.

Inductive algorithms are also known as polynomial neural networks [7-9]. GMDH is used in such areas as data mining and knowledge discovery [10, 11], forecasting [12, 13], modeling of complex systems [6, 14], optimization and pattern recognition [15, 16] when solving various applied problems [17-22].

The methods of self-organization are based on the following principles of the theory of heuristic self-organization: the principle of the external criterion and the principle of non-final decision [4, 23].

The principle of the external criterion is that the quality control of the predictive model (the accuracy of the experimental data approximation by the regression equation) is estimated using a criterion that is external to the criterion by which the coefficients are determined.

One of the options for the external criterion can be a breakdown of all experimental points into two parts, where the first of them, the training sample, is used to determine the coefficients using LSM, and the second part of the experimental points, the test (external) sample, is used to evaluate the accuracy of the regression equation.

The use of an external criterion makes it possible to obtain a model of optimal complexity.

The principle of inconclusiveness of the solution is borrowed from evolutionary and genetic algorithms and lies in the fact that at each step of the iteration remains a group of best solutions, that is, hopeless solutions are cut off, and only at the last step the only best (optimal) solution among all equal solutions is selected [5, 24].

The essence of self-organization algorithms is that a complete description of an object of the form (1) is replaced by a set of private descriptions [4, 25]. As private descriptions, two-factor polynomials of degree no higher than the second are used:

Linear private description

$$Y_i(x_k, x_l) = a_{0i} + a_{1i}x_k + a_{2i}x_l. \tag{2}$$

Private description with covariance

$$Y_i(x_k, x_l) = a_{0i} + a_{1i}x_k + a_{2i}x_l + a_{3i}x_k x_l. \tag{3}$$

Quadratic private description

$$Y_i(x_k, x_l) = a_{0i} + a_{1i}x_k + a_{2i}x_l + a_{3i}x_k x_l + a_{1i}x_k^2 + a_{2i}x_l^2, \tag{4}$$

where: $x_k, x_l$ - factors that are included in the $i$-th model.

With the number of factors $n$, the number of private descriptions is equal to the number of possible combinations of factors as $2 - C_n^2$

For each private description, the response values at all experimental points and the accuracy relatively to the external criterion are calculated.

The complication of the model (an increase of the number of variables and the degree of the polynomial) is performed during the transition to the next step of the iteration, and the responses of the best models of the previous step (relatively to the external criterion) are the initial factors of the next step of the iteration - a recursive substitution of the results of the $i$-$1$- th step into $i$- th.

Many self-organization algorithms are distinguished by the structure of private descriptions, the external criterion (accuracy of the model in the points of the test sequence, balance of coefficients), the method of obtaining the resulting model, etc. [4, 23, 25].

Thus, from the considered features of self-organization algorithms, we can conclude that each algorithm has its own field of application, but for all algorithms, an increase in the model dimension as a result of an iterative procedure automatically leads to a corresponding increase in the degree of the resulting polynomial. An exception is combinatorial algorithms with linear private descriptions, but they allow you to build only a linear multidimensional model.

Therefore, the use of well-known self-organization algorithms for constructing non-linear models of multidimensional objects of low dimension is not very effective. As almost all types of private descriptions are used in well-known algorithms, an approach based on changing the iterative model complication procedure is promising.

In this case, linear private descriptions of each iteration step can be obtained by a combinatorial algorithm, and the resulting model (instead of a recursive substitution) is synthesized in the form of a multiplicative model of linear partial descriptions of the previous *i-1*-th and current *i*-th iteration steps.

# 3     Formal problem statement

Let there be a vector of input actions $X_i = \{ x_{1i}, ..., x_{ni} \}$ and an output $Y_i$ of each i-th instance $i = \overline{1, N}$ of a training sample of volume *N*. Build a model of regression dependence of the form (1) using the basic principles of self-organization methods.

The aim of the work is to develop a method of self-organization in which the degree of the resulting regression equation increased minimally (by one) when moving to the next step of the iteration, and the number of factors can be maximum already in the first step. To achieve this aim, the following tasks are solved:
— to develop a self-organization method for the synthesis of a multiplicative model with linear private descriptions;
— to develop computational procedures for implementing the method;
— to assess the effectiveness of the method on test cases.

# 4     Development of a self-organization method for the synthesis of a multiplicative model with linear private descriptions

In the developed method, all the principles of self-organization algorithms must be observed, therefore, in developing of the method, it is necessary to formalize the following stages of information conversion:

- Generation of linear private description structures;
- Calculation of the coefficients of a private description of a given structure at the experimental points of the training sample;
- Calculation of responses of private descriptions at the points of the test sample and selection of the best solutions;
- Transition to the next step of iteration and implementation of the model complication procedure;
- Formation of a stopping criterion and selection of the optimal solution.

## 4.1     Generation of linear private description structures

As individual descriptions at any step of the iteration, linear polynomials from *n* variables (*n* is the dimension of the object) and the set of all possible polynomials from *i* variables $i = \overline{1, n-1}$ with a different combination of non-repeating arguments are used. The number of possible polynomials is

$$N_p = \sum_{j=1}^{n} C_n^j = 2^n - 1,\tag{5}$$

which is significantly more than the number of private descriptions in known algorithms.

In the software implementation of the generation of structures, an integer array of length $n$ is used, each element of which contains 0 or 1 and is used as a mask of the structure (1-given factor is present in the description; 0-missing). For each value of $i$ $i = \overline{1, n-1}$, $i$ units are written to the beginning of the array, after which the last unit is shifted one position to the right, forming a new mask and so on until the end of the array.

### 4.2 Calculation of the coefficients of a private description of a given structure at the experimental points of the training sample

The calculation of the coefficients of private description of given structure at the first step of the iteration does not present any problems and is performed by the least squares method at the points of the training sequence of input actions $X = \{ x_1,..., x_n \}$ and output $Y$.

The calculation of the responses of private descriptions at the points of the test sample and the selection of the best solutions are also carried out by known methods. For simplicity of the algorithmic implementation, the number of best solutions is chosen equal to the number of input factors $n$.

In order for the degree of the resulting regression equation to increase by one by passing to the next iteration step, the regression equation is constructed in the form of a multiplicative model of the form

$$P_s^k(X) = \prod_{i=1}^{k} P^i(X),\tag{6}$$

where $P^i(X)$ - is the linear polynomial of the $i$-th step of the iteration.

It should be noted that in this method the polynomials of any iteration step use the input factor vector $X$ as arguments. As was noted earlier, in the known methods, the vector $X$ is used only at the first step of the iteration, at all other steps the input is the responses of the best decisions of the previous step.

Expression (6) shows that, in the process of complicating the model, $N_p$ (determined by (5)) of the polynomials of the current $k$-th iteration step are "built in" over each of the best solutions of the previous $k – 1$ th step. The decisions of the previous step are the "parents" of the decisions of the current step - the "descendants". Thus, at each step of the iteration (except the first), the number of particular descriptions is equal to the product of $n$ by $N_p$. Of this set of particular descriptions, only $n$ best ones are skipped to the next step.

The stop criterion is the "left corner" rule adopted in self-organization methods — an increase or stabilization of the error of the best private descriptions at the points of

the verification sequence. In the last step of the iteration, the only model selected from the best is the result of the method. After opening the brackets in (6), a polynomial is obtained in the form (1).

*Calculation of the coefficients of a private description of a given structure at the k-th step of the iteration (k> 1).* According to expression (6), the arguments of the polynomials of the *k*-th step is the vector of input factors. Then, in the classical application of LSM, the coefficients of these polynomials will be determined from the condition of minimizing the standard deviations between the responses of the analytic function $P^k(X)$ and the output value of *Y*. In this case, $P^k(X)$ approximates *Y* independently, without taking into account the results of the approximation of its "parent" in the previous step.

And the multiplicative convolution of the "parent" and "descendant" will give an unpredictable result.

Therefore, it is necessary to develop a system of linear algebraic equations for calculating the coefficients of polynomials by LSM taking into account the fact that in the expression (6) the coefficients of the "parents" of all the previous steps are already defined and their responses *Y\** are known. It is necessary to determine the coefficients of the last factor in (6).

We write the functional of the LSM

$$F = \sum_{i=1}^{N_l}(Y_i - P_s^k(X_i))^2 \rightarrow min , \qquad (7)$$

where *Xi*, *Yi* are the values of the input and output actions, respectively; $P_s^k(X_i)$ is the response of the multiplicative model of the current iteration step, which is determined by (6): the summation is carried out over the points of the learning sequence of volume $N_l$.

We write linear polynomials of the current step in expanded

$$P^k(X_i) = a_0 + \sum_{j=1}^{m} a_j x_{ji} \qquad (8)$$

Taking into account (8), we replace the polynomials of the previous steps in the expression of the multiplicative model (6) with their responses *Y\** and substitute in (7)

$$F = \sum_{i=1}^{N_l}(Y_i - Y_i^*(a_0 + \sum_{j=1}^{m} a_j x_{ji}))^2 \rightarrow min , \qquad (9)$$

This functional in the space of parameters (polynomial coefficients) is a convex function and has one global minimum. To find the minimum, it is necessary to take the private derivatives with respect to each coefficient and equate them to zero. As a result, we get

$$\frac{\partial F}{\partial a_0} = 2\sum_{i}((Y_i - Y_i^*(a_0 + a_1 x_{1i} + ... + a_m x_{mi}))Y_i^*) = 0 ;$$

$$\frac{\partial F}{\partial a_1} = 2\sum_i ((Y_i - Y_i^*(a_0 + a_1 x_{1i} + ... + a_m x_{mi}))Y_i^* x_{1i}) = 0$$

…………………………………………………………………………………

$$\frac{\partial F}{\partial a_m} = 2\sum_i ((Y_i - Y_i^*(a_0 + a_1 x_{1i} + ... + a_m x_{mi}))Y_i^* x_{mi}) = 0.$$

Having opened the brackets and performed simple arithmetic operations, we obtain a system of linear algebraic equations with respect to unknown coefficients $a_i\ i = \overline{0,m}$ of the form:

$$\begin{cases} \sum_i (Y_i^{*2})a_0 + & \sum_i (Y_i^{*2}x_{1i})a_1 + & ... & \sum_i (Y_i^{*2}x_{mi})a_m = & \sum_i (Y_i^* Y_i) \\ \sum_i (Y_i^{*2}x_{1i})a_0 + & \sum_i (Y_i^{*2}x_{1i}x_{1i})a_1 + & ... & \sum_i (Y_i^{*2}x_{1i}x_{mi})a_m = & \sum_i (Y_i^* Y_i x_{1i}) \\ ... & ... & ... & ... & ... \\ \sum_i (Y_i^{*2}x_{mi})a_0 + & \sum_i (Y_i^{*2}x_{mi}x_{1i})a_1 + & ... & \sum_i (Y_i^{*2}x_{mi}x_{mi})a_m = & \sum_i (Y_i^* Y_i x_{mi}) \end{cases} \quad (10)$$

Solving the system of equations (10) by any method, we obtain the coefficients of the polynomial, which gives a minimum of functional (7).

It should be noted that at the first step of the iteration, the previous solutions are absent; therefore $Y^* \equiv 1$, the system of equations (10) turns into the classical system of equations of the LSM method for linear multifactor regression, which confirms the correctness of the above transformations.

## 5      Experiments and results

To study the effectiveness of the developed method, test calculations were performed. The calculations were performed using the mathematical package MATLAB [26]. Programs have been created to implement the developed method and the well-known method with quadratic partial descriptions (4).

For comparative analysis, three types of equations are taken:
1) large dimension, small degree;
2) small dimension, small degree;
3) a large dimension, a large degree.

The equations are given in Table 1.

For each of the given equations, a table of initial data is generated. As values of the input vector, and also as values of the coefficients of the equations, pseudorandom numbers uniformly distributed over the interval [-2, 2] obtained using the standard function *RAND*.

The sample size is 100 points (50-training sequence; 50-verification).

**Table 1.** Initial Functional Dependencies

| Type | № | Equations |
|------|---|-----------|
| 1 | 1 | $Y= 1{,}42–1{,}61x_1+0{,}221x_2–0{,}714x_3+0{,}443x_1x_3+0{,}176x_4+0{,}358x_2x_5$ |
|   | 2 | $Y= –1{,}33+1{,}25x_1–1{,}83x_2+1{,}42x_3–1{,}61x_4+0{,}225x_5x_1–0{,}714x_5$ |
| 2 | 3 | $Y= 1{,}42–1{,}61x_3^2+0{,}221x_4^3$ |
|   | 4 | $Y= 1{,}25–1{,}83x_5^2+1{,}45x_5^4$ |
| 3 | 5 | $Y= –0{,}714+0{,}174x_1^3+0{,}443x_2^4+0{,}358x_3^2–1{,}33x_4^3+1{,}25x_5$ |
|   | 6 | $Y= 0{,}358–1{,}33x_1^3+1{,}25x_2^3–1{,}83x_3^2+1{,}42x_4^3$ |

The external criterion for selecting the best solutions is the error at the points of the verification sequence, which is determined by the expression

$$\varepsilon_v = \frac{\sum_{i=1}^{N_v}(Y_i - P_s^k(X_i))^2}{\sum_{i=1}^{N_v}Y_i}100[\%] \qquad (11)$$

where: $Y_i$ is the value of the output quantity;

$P_s^k(X_i)$ – predicted value of the output value by the model.

Further, according to the table of experimental data, the initial functional dependence is restored using the most common self-organization algorithm with quadratic quotients of the form (4), hereinafter referred to as Algorithm 1, and the developed self-organization algorithm for the synthesis of multiplicative models with linear private descriptions (Algorithm 2). A comparative analysis of the calculations is given in Table 2.

**Table 2.** Comparison of effectiveness

| Type of equation | № of equation | Degree of equation n | Dimension of equation N | Algorithm 1 | | | | Algorithm 2 | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | Number of steps r | Degree of equation n | Dimension of equation N | Error $\varepsilon_v$ (%) | Number of steps r | Degree of equation n | Dimension of equation N | Error $\varepsilon_v$ (%) |
| 1 | 1 | 2 | 5 | 3 | 8 | 4 | 6,603 | 2 | 2 | 5 | 1,34 |
|   | 2 | 2 | 5 | 3 | 8 | 5 | 7,703 | 2 | 2 | 5 | 0,377 |
| 2 | 3 | 3 | 2 | 1 | 2 | 2 | 0,148 | 2 | 2 | 2 | 3,34 |
|   | 4 | 4 | 1 | 2 | 4 | 3 | 0,964 | 3 | 3 | 2 | 2,89 |
| 3 | 5 | 4 | 5 | 3 | 8 | 4 | 4,234 | 2 | 2 | 5 | 6,44 |
|   | 6 | 3 | 4 | 3 | 8 | 4 | 0,366 | 2 | 2 | 4 | 7,87 |

From a comparative analysis of the data table. 2 by criteria [27]:

- error of the resulting regression equation $\varepsilon_v$;
- number of iteration steps $r$;
- dimension $N$ of the resulting regression equation $Y*$
- degree $n$ of the resulting regression equation $Y*$,

the preferred application of Algorithm 2 for the first type of equations is visible (equation of large dimension of a low degree).

To the previously noted results of the comparative analysis, we should add the simplicity of obtaining the resulting regression equation obtained by Algorithm 2.

The resulting regression equation in Algorithm 2 is the multiplication of linear private descriptions of the various iteration steps, but to obtain the resulting regression equation in Algorithm 1, it is necessary to do the step-by-step operation of substituting private descriptions of the previous iteration steps in the subsequent ones until we obtain the dependence of the output function $Y$ from the input vector $X$.

As an example, we show the form of the best solution for approximating the output values of $Y$ of the original equation 1, the form of which is shown in Table 1.

In accordance with the data Table 2, the results are as follows:

- Algorithm 1:
  - error $\varepsilon_v = 6,603\%$
  - the number of iteration steps $r = 3$;
  - the type of analytical expressions (in the induced expressions, the results of the previous iteration steps are indicated by new variables).
    - 3-rd step of iteration:
    - $Y(z) = -0,0407+0,53z_4+0,481z_7-1,26z_4z_7+0,649z_4^2+0,616z_7^2$
    - 2-nd step of iteration:
    - $z_4(v) = 0,0349+0,938v_1+0,16v_6+0,283v_1v_6-0,077v_1^2-0,288v_6^2$
    - $z_7(v) = -0,017+0,885v_2+0,558v_6+0,298v_2v_6-0,168v_2^2-0,175v_6^2$
    - 1-st step of iteration:
    - $v_1(x) = 1,19-1,79x_1+0,57x_2+0,014x_1x_2+0,242x_1^2+0,024x_2^2$
    - $v_2(x) = 2,91-2,79x_1-0,92x_3+0,78x_1x_3+0,369x_1^2-0,09x_3^2$
    - $v_6(x) = -0,98+0,69x_2-0,08x_4-0,573x_2x_4+0,38x_2^2+0,34x_4^2$
- Algorithm 2:
  - error $\varepsilon_v = 1.34\%$
  - the number of iteration steps $r = 2$;
  - type of analytical expressions
    - 1-nd step of iteration:
    - $Y(x) = 0,49-1,24x_1+0,67x_2-0,19x_3+0,23x_4+0,33x_5$
    - 2-st step of iteration:
    - $Y(x) = 1,12-0,01x_1-0,25x_3-0,04x_4+0,11x_5$

The change in the average error $\varepsilon_v$ of the best private descriptions while complicating the model for the previously considered functional dependences is given in Table3.

Analysis of the results of the Table 3 shows:

- compared with Algorithm 1 in Algorithm 2, the first step of the iteration makes a significant contribution to reducing the error εv, since already at the first step the algorithm selects the most significant factors.
- to obtain the resulting regression equation, Algorithm 2 requires no more iteration steps than Algorithm 1 for almost all types of equations.

**Table 3.** The change in the average error of the best private descriptions

| № equation | Average error $\varepsilon_v$ of the best private descriptions (%) | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Algorithm 1 | | | | Algorithm 2 | | | |
| | Iteration step | | | | Iteration step | | | |
| | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 |
| 1 | 29,02 | 8,43 | 6,84 | 6,8 | 5,25 | 1,38 | 1,32 | - |
| 2 | 17,9 | 11,4 | 8,6 | 8,51 | 4,17 | 0,345 | 0,336 | - |
| 3 | 14,08 | 1,805 | 0,139 | 0,138 | 7,82 | 3,41 | 3,23 | 3,24 |
| 4 | 19,14 | 1,191 | 1,175 | 1,2 | 34,2 | 4,65 | 2,91 | 2,76 |
| 5 | 39,4 | 7,8 | 5,31 | 5,34 | 20,8 | 6,71 | 6,43 | 6,41 |
| 6 | 45,5 | 17,3 | 9,89 | 9,51 | 16,0 | 8,91 | 8,66 | 9,61 |

## 6    Conclusion

A self-organization method is developed in the work for the synthesis of a multiplicative model with linear private descriptions, in which the degree of the resulting regression equation increases minimally (by one) when moving to the next iteration step, and the number of factors can be maximum already in the first step. This distinctive property of the method cannot be implemented by well-known methods of self-organization, it determines the area of effective use of the method for constructing non-linear regression multifactor models of low-dimensional multidimensional objects. The stages of information conversion are formalized and computational procedures for the implementation of the noted stages are developed. The features of calculating the coefficients of polynomials of the current iteration step in the iterative procedure for the synthesis of the multiplicative model are considered, and analytical expressions are obtained for the first time to calculate the coefficients of linear polynomials by least squares taking into account the noted features. A software implementation of the method using the MATLAB package has been performed. The developed program was tested using test examples, which confirmed its operability, determined its effectiveness and scope. The prospect of further work is the development of a full-fledged software product and the processing of real data.

## References

1. Draper N., Smith G. (2016) Prikladnoj regressionny`j analiz. 3-e izdanie [Applied regression analysis. 3rd edition]. Moscow, «DIALEKTIKA» (in Russian)

2. Radchenko S. G. (2011) Metodologiya regressionnogo analiza: Monografiya. [Regression analysis methodology: Monograph] Kiev «Kornijchuk», (in Russian)

3. Demidova O. A., Malahov D. I. (2019) Ekonometrika. Uchebnik i praktikum dlya prikladnogo bakalavriata [Econometrics. Textbook and workshop for applied baccalaureate]. Moscow, Publishing House Yurayt (in Russian)

4. Ivakhnenko A.G. (1982) Induktivny`j metod samoorganizaczii modelej slozhny`kh sistem. [Inductive method of self-organization of models of complex systems]. Kiev, Naukova Dumka (in Russian)

5. Ivakhnenko, A.G., Müller, J.A. (1992) Parametric and nonparametric selection procedures. Experimental Systems Analysis. In: Systems Analysis, Modelling and Simulation (SAMS), 1992, vol.9, pp 157-175

6. Stepashko V. S. (2016) Kontseptualnyye osnovy intellektualnogo modelirovaniya [Conceptual Foundations of Intelligent Modeling] In: Control Systems and Computers, 2016, 4, pp 3-15 (in Russian)

7. Schmidhuber, Jürgen. (2015) Deep learning in neural networks: An overview Neural Networks, 2015, vol. 61, pp 85-117. doi:10.1016/j.neunet.2014.09.003. – arXiv:1404.7828

8. Lytvynenko, V., Wojcik, W., Fefelov, A., Lurie, I., Savina, N., Voronenko, M., Boskin, O., Smailova, S. (2020) Hybrid Methods of GMDH-Neural Networks Synthesis and Training for Solving Problems of Time Series Forecasting Advances in Intelligent Systems and Computing, 1020, pp 513-531. doi: 10.1007/978-3-030-26474-1_36

9. Kondrashova N. (2015) Network Structures Algorithms of Group Method of Data Handling. In: Inductive modeling of complex systems: Sat. sciences. works. Kyiv, IRTC ITS NAS and MES of Ukraine, 2015, 7, pp 21-38.

10. Anastasakis, L., and N. Mort. (2001) The development of self-organization techniques in modelling: A review of the group method of data handling (gmdh). Tech. Rep.; University of Sheffield Department of Automatic Control and Systems Engineering

11. Timashova L., Vitkovski T. (2015) Tehnologiya intellektualnogo proizvodstvennogo modelirovaniya virtualnyih predpriyatiy [Intelligent Manufacturing Technology for Virtual Enterprises]. In: Bulletin of NTU "KhPI". Series Informatics and Modelingб 2015, 32, pp 136-147 (in Russian)

12. Efimenko S.M. (2015) Prognozirovanie slozhnyih protsessov v klasse modeley vektornoy avtoregressii na osnove rekurrentno-parallelnogo algoritma COMBI MGUA [Prediction of complex processes in the class of vector autoregression models based on the recurrent-parallel algorithm COMBI MGUA]. In: Inductive modeling of complex systems: Sat. sciences. works. Kyiv, IRTC ITS NAS and MES of Ukraine, 2015, 7, pp 129-139. (in Ukrainian)

13. Srinivasan, D. (2008) Energy demand prediction using GMDH networks. In: Neurocomputing, 2008, 72, pp 625-629

14. Onwubolu, G. C. (2009) Hybrid self-organizing modeling systems, Vol. 211. Springer, Ontario.

15. Takao S., Kondo S., Ueno J., Kondo T. (2017). Deep feedback GMDH-type neural network and its application to medical image analysis of MRI brain images. In: Artificial Life and Robotics, 23 (2), pp 161-172. doi:10.1007/s10015-017-0410-1

16. Timofieva N.K. (2016) Pro universalnist metodu strukturno-alfavitnogo poshuku [On the universality of the method of structural-alphabetical search] In: Inductive modeling of complex systems: Sat. sciences. works. Kyiv, IRTC ITS NAS and MES of Ukraine, 2016, 8, pp 185-193 (in Ukrainian)

17. Teng, G., Xiao, J., He, Y., Zheng, T. & He, C. (2017) Use of Group Method of Data Handling for Transport Energy Demand Modeling. In: Energy Science and Engineering, Vol. 5, No. 5, pp 302-317. doi: 10.1002/ese3.176.

18. Viacheslav V. Zosimov, Oleksandra S. Bulgakova (2019) Calculation the Measure of Expert Opinions Consistency Based on Social Profile Using Inductive Algorithms. In: ISDMCI 2019: pp 622-636

19. Viacheslav V. Zosimov, Volodymyr Stepashko, Oleksandra S. Bulgakova (2015) Inductive Building of Search Results Ranking Models to Enhance the Relevance of Text Information Retrieval. In: DEXA Workshops 2015: pp 291-295

20. Iutinska G. O., Moroz O. G. (2017).Induktivne modelyuvannya zmini chiselnosti amilolitichnih mikroorganizmiv na zabrudneniy dilyantsi gruntu [Inductive modeling of changes in the number of amylolytic microorganisms in a contaminated soil area] In: Inductive modeling of complex systems: Sat. sciences. works. Kyiv, IRTC ITS NAS and MES of Ukraine, 2017, 9, pp 101-107  (in Ukrainian)

21. Simjanovska M., Gusev M., Madevska-Bogdanova A. (2014) Intelligent modelling for predicting students' final grades In: Proc. of 37th Int. Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), Opatija, 2014, pp 1216-1221.

22. Stepashko V., Samoilenko O., Voloschuk R. (2014) Informational Support of Managerial Decisions as a New Kind of Business Intelligence Systems. In: Computational Models for Business and Engineering Domains, Rzeszow, Poland; Sofia, Bulgaria: ITHEA, 2014, pp 269-279.

23. Stepashko V.S. (2017) Dostizheniya i perspektivyi induktivnogo modelirovaniya [Achievements and prospects of inductive modeling] In: Control Systems and Computers, 2017, 2, pp 58-73 (in Russian)

24. Pavlov A.V., Stepashko V.S., Kondrashova N.V. (2014) Effektivnyie metodyi samoorganizatsii modeley [Effective methods of self-organization of models] Kyiv, Akademperiodika (in Russian)

25. Stepashko V. S., Bulgakova A. S. (2013) Obobshhenny`j iteraczionny`j algoritm metoda gruppovogo ucheta argumentov [A generalized iterative algorithm for the method of group accounting of arguments] Control Systems and Computers. – 2013. – № 2. – pp. 5-17  (in Russian)

26. Dyakonov V.P. (2012) MATLAB. Polny`j samouchitel` [MATLAB. Complete tutorial] Moscow, "DMK-Press" (in Russian)

27. Stepashko V. S., EfImenko S. M., Savchenko E. A. (2014) Komp'yuterniy eksperiment v induktivnomu modelyuvanni [Computer experiment in inductive modeling], Kyiv, Naukova dumka (in Ukrainian)