

A.V. Pavlov, V.S. Stepashko, N.V. Kondrashova

EFFECTIVE METHODS OF MODELS SELF-ORGANIZATION

Translated in 2026, email:gmdhmail@gmail.com

Table of Contents

Foreword	3
Acknowledgments	3
Introduction	4
Chapter 1.....	8
Analysis of tasks, approaches, methods and algorithms for construction of models based on observational data	8
1.1 Typical tasks, approaches and methods of computer heuristic modeling	8
1.1.1 The Task of Structural-Parametric Identification.....	10
1.1.2 Approaches to solving the problem of forecasting.....	13
1.1.3 Group Method of Data Handling (GMDH) as a part of mathematical methods	15
1.2 GMDH as an inductive method of heuristic self-organization of models.....	17
1.2.1 General Formulation of the Problem of the Inductive Method of Self-Organization of Models	18
1.2.2 Tasks of the inductive method of model self-organization	19
1.2.3 Basic Principles of GMDH	19
1.2.4 The task of finding a model that corresponds to the global extremum of the quality functional.....	20
1.2.5 The problem of partitioning a set of observations.....	21
1.3 Basic algorithms of GMDH	22
1.3.1 Sorting GMDH algorithms.....	23
1.3.2 Iterative GMDH algorithms	24
1.4 Convergence of iterative GMDH algorithms	25
1.5 Classical iterative algorithms	26
1.6 Modern iterative algorithms based on GMDH.....	30
1.7 Discussion	32
Chapter 2.....	34
The Problem of Sample Partition in Heuristic Modeling.....	34
2.1 Optimal Sample Partition as a Result of Experiment Design	34
2.2 Relationship between optimal partitioning and partitioning by ‘variance’	35
2.3 Quasi-optimal partitioning in the case of passive sampling.....	36
2.3.1 Criteria for finding a quasi-optimal partition	37
2.4 Discussion	39
Chapter 3.....	40
Relaxation Iterative Algorithms of GMDH	40
3.1 Multilayered simplified GMDH algorithm	40
3.1.1 Description of a multi-layered simplified algorithm.....	40
3.1.2 Computational complexity of MSA	44
3.1.3 A way to find the best multiplicative monomial	45
3.2 Generalized relaxation iterative algorithm (GRIA)	47
3.2.1 Model construction process.....	48
3.2.2 Algorithm architecture	50
3.2.3 Method for finding a multiplicative monomial	50
3.2.4 Methods for estimating parameters and calculating selection criteria	51
3.2.5 Computational complexity of GRIA	54
3.3 Proof of GRIA convergence.....	55
3.4 Comparison of GRIA and MSA.....	58
3.4.1 Theoretical comparative analysis of the speed of GRIA and MSA	58
3.4.2 Comparison of the performance of GRIA and MSA software implementations	59
3.5 Discussion	60
Chapter 4.....	62
Investigation of the Properties of Iterative Algorithms by Computer Experiments	62
4.1 Investigation of the Convergence Property of Algorithms	62

4.1.1 Methods of conducting computer experiments	64
4.1.2 Study of the GRIA convergence rate	65
4.2 Study of the Identifying Property of Algorithms	82
4.2.1 Study of the Impact of Data Characteristics.....	82
4.2.2 Study of the Influence of Algorithm Parameters	84
4.3 Study of the Noise-Immunity Property of Modeling	85
4.4 Study of the Parameter of Freedom of Choice	89
4.5 Discussion	91
Chapter 5.....	93
Solving practical modeling problems	93
5.1 Space weather forecasting.....	94
5.1.1 Description of initial data	94
5.1.2 Data pre-processing	95
5.1.3 Statement of the modeling problem	96
5.1.4 Problem Solution.....	97
5.1.5 Modeling results	98
5.1.6 Analysis of modeling results	99
5.1.7 Conclusions on the task of space weather forecasting	103
5.2 Evaluation of the effectiveness of medications.....	103
5.2.1 Problem Statement	104
5.2.2 Problem solution	105
5.2.3 Modeling results	108
5.2.4 Analysis of modeling results	109
5.2.5 Conclusions on the task of assessing drugs efficacy	110
5.3 Differential diagnosis of haemostasis pathologies	110
5.3.1 Problem statement	110
5.3.2 Problem solving concept	111
5.3.3 Solving the problem	112
5.3.4 Modeling results	113
5.3.5 Modeling analysis	115
5.3.6 Conclusions on the task of differential diagnosis.....	116
5.4 Discussion	116
Conclusion.....	117
Appendix A	119
Additional computer experiments	119
References	126

Foreword

Dedicated to the blessed memory of A.G. Ivakhnenko

There are quite a lot of people who knew A.G. Ivakhnenko or are familiar with his works in the academic and technical environment. Thanks to his several dozen monographs, numerous articles, specialized websites, annual international conferences and seminars on topics related to his work, the memory of him is not erased.

Alexey Grigoriyevich was an active propagandist of his ideas during his lifetime. Twelve years have passed since his passing. Ivakhnenko's school continues to develop, its adherents - to get new results. We have realized an old dream of writing a book with a retrospective look at the results in inductive modeling, and at the same time looking to the future. We place in it the materials of the latest research and solutions of practical problems of large dimensions, which have not yet been declared as a field of application of the Group method of data handling (GMDH). Scientific literature does not have to be easy to read, nevertheless, we tried to make the book understandable and accessible to many. It brings together the results we have achieved over the past decade. A new exposition of Ivakhnenko's ideas, in our opinion, is necessary because twenty years have passed since the publication of the last monograph written by the master, and few people read them except for students and people who are deeply interested in his legacy. Our task was to revive interest in this problem with a useful and interesting book, both for specialists engaged in mathematical modeling and for a wide range of readers.

With all the large number of A.G. Ivakhnenko's school, a large number of embodiments of his ideas and practical implementations, we must admit that we do not have a complete idea of all the results. This was one of the main reasons why we took up the task (along with the desire to talk about our own work).

The book is not intended for mathematicians – they will not learn anything new. We are writing for those who do not know GMDH well and would like to know better the latest achievements in this field. We cannot say that we have read all the available sources. And yet they tried not to miss important questions, and analysed the information contained therein. Taking care that the book is not boring to read, we have placed many illustrations in it.

The list of references is not exhaustive. With the limited volume of the book, taking into account the forty-five-year history of practical applications of GMDH, we could not help but reduce the number of references in the text to the limit. At the end of the book, we indicate for the most part those works that have been used by us and which we consider important. We did not try to trace the entire history of results.

The content of this book makes it possible to create various courses: on modeling complex systems, artificial intelligence, forecasting, data analysis, decision-making systems, etc.

Acknowledgments

We express our gratitude to Vladimir Anatolyevich Pavlov for his help in solution the problem of medical diagnostics of Section 5.3 of Chapter 5, Alexander Alexandrovich Obodnikov for the data provided for the task of assessing the effectiveness of medicines (Section 5.2 of Chapter 5), Vladimir Vladimirovich Tomilin for the provided data on medical differential diagnosis (Section 5.3 of Chapter 5).

Introduction

"All our knowledge is only more or less probabilistic, and not certain, as philosophers and theologians are accustomed to think.

Two problems have become much more important in modern logic than in earlier times. In first, it is the nature of probability, and in second, the significance of induction."

*Bertrand Russell,
Nobel Laureate in Literature, 1950*

At current stage of the exact sciences development and computerization of society, mathematical methods for models (or systems of models) synthesis and computer methods for their study are often used to solve practical modeling problems. When studying the characteristics of the functioning of any system using mathematical methods, a mathematical model must be created. Regression analysis methods, used in the analysis of experimental data, are among the most popular. This has been largely facilitated by the widespread computerization and the availability of information and programs via the internet. Ever-increasing amounts of computer's memory (RAM) and the increasing speed of computing equipment stimulate the creation of new and the improvement of existing algorithms for solving increasingly complex problems. It has become possible to accumulate huge amounts of real data, making analytics more democratic, as internet information becomes accessible to a wide range of people. Researchers are provided with a wide range of information for analysis, constructing mathematical models, studying process trends, comparing data characteristics, developing hypotheses, testing ideas, and drawing statistically valid conclusions. Statistical methods of mathematical modeling that don't rely on precise a priori hypotheses include the group-based method of argument accounting, an effective method of inductive modeling. This book is devoted to obtaining statistically validated results from inductive modeling approach.

Computer heuristic modeling or computer heuristic self-organization of models involves the use of heuristics when solving problems of observational data analysis on a computer. "Heuristic self-organization of models" [1] is one of the historical names of the group method of data handling [2] proposed by A.G. Ivakhnenko. Later, he called the same method as the inductive method of model self-organization [3]. The inductance of the process of construction models comes down to the transition from numerical data to functions that describe them, the transformation of simple models into more complex ones. Inductive modeling is focused on the creation of information technologies for modeling based on empirical data. In the book, we "reanimate" the relegated name "heuristic self-organization of models", because when modeling complex and poorly studied objects and processes, it is difficult to find another method that would be so simple and at the same time would allow researchers to find dependencies at the level of unknown patterns, that are subject to analysis and interpretation. Computer heuristic modeling helps us to obtain new conclusions, and "these conclusions are unexpected for us" (A.G. Ivakhnenko). It's time to exclaim: Eureka! - "He (computer) found!" The process of self-organization applies not only to models but also to algorithms. Self-organization of solutions obtained by various algorithms is the next stage of modeling. We can trace the development of GMDH algorithms from the simple idea of gradually increasing the models complexity to the wide variety of its implementations. The search for the best solution "drifted" over time toward obtaining the most probable solution. By analogy with an ensemble of models obtained

by a single algorithm, there exists an ensemble of solutions from various algorithms, and the success of solving a problem is an external criterion for selecting this ensemble.

Computer heuristic modeling helps to solve computationally complex problems and obtain "computational proofs" of the statistical validity of the decision. It involves the use of simple assessment methods, facilitation of strategies, reliance on practical experience, improvement of the structure of algorithms and much more.

The tasks of computer heuristic modeling can be structural-parametric identification, forecasting, extrapolation, clustering, recognition and a number of other tasks. Forecasting and extrapolation of the dynamics of environmental, economic and other processes, as well as recognition, classification and clustering in technical and medical problems are impossible without highly effective algorithms and technologies for analysing statistical information.

The purpose of the book is to acquaint a wide range of readers with modern methods of computer heuristic modeling of complex objects. By providing information about modern algorithms, we hope that the book will contribute to expanding the possibilities of heuristic modeling.

Sections of the book include specialized material.

In the *first chapter* is carried out a review and analysis of well-known classical and modern methods of constructing regression models based on empirical data. The problem of inductive modeling is formulated and one of the methods for solving of it is presented; the main algorithms of GMDH are described, and in most detail – iterative algorithms. The analysis allowed us to single out the algorithm from the relaxation-iterative class as the fastest and providing the most accurate solution. To improve it, the advantages of sorting-out algorithms were used. An analysis of the results of the convergence theory of iterative GMDH algorithms allowed us to identify and resolve unaddressed issues of the proof.

The *second chapter* is devoted to the problem of sample partition. It considers the conditions for obtaining an optimal partition and ways to increase the speed of the algorithm for finding the result of quasi-optimal partitioning; a comparison of the quasi-optimal with traditional methods of partitioning is carried out. In addition, the properties of the sample partitioning criteria and their agreement with the criterion of external complement when choosing models according to GMDH are analysed. The accuracy of identification and extrapolation made it possible to find ways to improve the results by visualizing and weighing the solutions obtained as a result of various splits. The use of quasi-optimal sample partition criteria to evaluate image transformations is one of the simple ways to determine the presence of their perspective-affine transformations.

The *third chapter* is devoted to a detailed presentation of the relaxation iterative algorithms (RIA) of GMDH, which are effective for solving of complex problems. A distinctive feature of these algorithms is their speed. The RIA gives good results both in the problem of identification and in forecasting. The chapter describes two GMDH algorithms of the relaxation type: the first is a modern algorithm of this class – the generalized relaxation iterative algorithm (GRIA); the second is its prototype – a multi-layered simplified algorithm (MSA). The description of GRIA is given a section in this chapter, because it is so far the only iterative algorithm of GMDH, which has linear computational complexity at the stage of model's construction. The convergence of iterative algorithms is an important issue in the theory of GMDH. To date, convergence has been proven in theory for only two iterative algorithms, including GRIA. The proof of its convergence is considered in this chapter as well.

The *fourth chapter* is devoted to the study of the main properties of the iterative algorithms of GMDH: noise-immunity, the algorithm's identifying property and the convergence property. The topic of algorithm parameters is also addressed, examining how parameter values influence the

accuracy of the solution. All computer experiments serve as "computational proofs". Analytical results on the rate of convergence to the "true" model are obtained in the absence and presence of noise in a sample data, using the GRIA study as an example. These results are statistical estimates – new to the theory of iterative relaxation GMDH algorithms. The chapter describes a methodology for studying the convergence rate of an iterative algorithm and demonstrates how to conduct experiments to investigate the other two properties of an algorithm. Using the GRIA example, it is demonstrated that it possesses all three properties.

The fifth chapter contains examples of solving practical modeling problems of varying nature and initial data sample size. The first problem - forecasting the onset of a geomagnetic storm is related to preventing the consequences of magnetic field disturbances, which have a negative impact on both energy systems and human health. A large-scale problem can have more than 500000 observations - the new relaxation iterative algorithm GRIA with recurrent evaluation was recently developed for precisely these types of problems. The second task is to model an indirect characteristic associated with the process of changes in the immune system of cancer patients exposed to medications. The third is to determine the diagnosis of mild cases of haemostasis pathology.

List of abbreviations

CML	convergence multi-layered algorithm
COMBI	combinatorial algorithm
CP	coagulopathy
CPHS	combined pathology of the haemostasis system
CR	model selection criterion (external criterion)
CS	complete sorting
DM	disparity measure
DT	disaggregation thrombocytopathy
DVP	dissimilar by variance partition
GMDH	group method of data handling
GRIA	generalized relaxation iterative algorithm
GSE	geocentric solar ecliptic - a coordinate system associated with the geocentric solar ecliptic
GSM	geocentric solar magnetospheric - a coordinate system of geocentric solar magnetosphere
IMF	interplanetary magnetic field
LSM	least squares method
MACSO	modified algorithm with combinatorial selection and orthogonalization
MATSO	modeling algorithm with three-stage partitioning
MGP	matrix generation procedure
MIA	multi-layered iterative algorithm
MMDP	multiplicative monomials design procedure
MSA	multi-layered simplified algorithm
OMNI2 DB	OMNI2 database
PDF	partial description function
<i>PRT</i>	trend forecast
QOP	quasi-optimal partitioning
QR	model error (internal criterion)
RIA	relaxation iterative algorithm
RIA CS	RIA with a complete sorting of model structures at each iteration
RIA DS	RIA with directional sorting of model structures at each iteration
RIA NC	RIA with non-recurrence computing
RSS	residual sum of squares
<i>SWF</i>	space weather forecaster – an algorithm for construction models for magnetic storms forecasting
SVP	similar by variance partition
TDS	thiol-disulfide ratio
WD	von Willebrand disease
$\bar{X}$	matrix of false (dummy) arguments
Σ_X	correlation matrix of input variables
F	freedom of choice of iterative algorithm solutions
$AR_B = AR_{B A}$	regularity criterion
d	determinant of the correlation matrix of input variables
D_{st}	disturbance storm time - geomagnetic index that identifies a magnetic storm
n_A, n_B, n_C, n_U, n_W	cardinalities of the sets A, B, C, U, W respectively
$U = A \cup B$	working set (sample) of observations
W	initial set (sample) of observations
$\mathbf{W} = (\mathbf{X}; \mathbf{y})$	source data matrix
$\mathbf{X}$	data matrix of initial variables
$\mathbf{X}^0$	matrix of true arguments
$\mathbf{y}$	vector of values of output variable

Analysis of tasks, approaches, methods and algorithms for construction of models based on observational data

Today, there are many unsolved problems of mathematical modeling, which is an integral part of intelligent control. Many working in this field face the challenges of handling enormous amounts of empirical data. Colossal funds have been spent on obtaining information that must be quickly processed on computers. Increasing the speed of obtaining forecast results, increasing the accuracy in recognition and forecasting tasks are the most urgent tasks.

The solution of these problems is impossible without the development of methods used in information processing. In mathematical modeling, deductive and inductive methods are usually distinguished. Deductive methods construct models based on the theory and laws of an object's functioning for a specific object, using the principle "from the general to the specific," as opposed to inductive methods, which proceed "from the specific to the general." However, deductive methods are unable to provide satisfactory results in modeling complex, poorly understood processes and systems.

The history of the concept of induction goes back to the works of F. Bacon (16th century), it is believed that before him this concept was formed by Democritus (460-370 BC), Socrates (469-399 BC) and Aristotle (384-322 BC) [4]. Bacon called for looking for the forms of the listed facts as something stable in the phenomenon. He introduced observation and experiment into induction. His ideas were developed by D. S. Mill (19th century). It is impossible not to agree with the words of B. Russell: "The proof for a scientific worldview is observation and induction" [5]. Russell was impressed by Mill's work. The statement of N. Wiener, "Inductive logic, the logic of Bacon, is rather something in accordance with which we can act than something we can prove" [6], suggests that it is impossible to prove inductive logic. A well-known development of D.S. Mill's inductive methods in artificial intelligence systems is the work [7]. The leaders in response to the Ukrainian-, English- and Russian-language internet query "inductive modeling" are representatives of the international school of A.G. Ivakhnenko and its adherents.

The goal of inductive modeling, based on numerical empirical observational data, is to determine both the structure and parameters of the model. The *structure* of the model reflects an unknown pattern contained in the data.

1.1 Typical tasks, approaches and methods of computer heuristic modeling

The tasks of computer heuristic modeling can include identification and extrapolation [9], forecasting and object management [10], classification and clustering [11], and computational proofs.

Let us focus on the last task: "computational proofs." Let us cite a well-known definition: "Proof-based computations are targeted computations on a computer, combined with analytical research, which lead to the rigorous establishment of new facts and the proof of theorems" [82]. It is worth emphasizing that in "computational proofs," as opposed to "evidence-based computations" [12], the word "computational" is emphasized. The difference between these concepts is the same as between induction and deduction. One is impossible without the other. But while "evidence-based computations" proceed from theory to individual, unknown, particular results within the framework of the same theory, computational proofs, on the contrary, present numerous computational data, the analysis of the results of which yields new knowledge and new facts. The third and fourth chapters contain both theoretical and computational proofs of the convergence of iterative algorithms. When

these results are consistent, they reinforce each other. This lends greater credibility to the computational proof method used for problems for which theoretical results are not yet available, such as determining the convergence rate of an iterative algorithm.

As it was mentioned, in the inductive approach, empirical initial data, as a rule, are represented by a numerical matrix of $W = (X : y)$ measurements, where X is the matrix of the object's input variables to be registered and y is the vector of the output variable, $\dim X = n_w \times m$, $\dim y = n_w \times 1$, where n_w is the number of measurements, m is the number of input variables.

Fig. 1.1 illustrates the "black box" approach. An object is understood as a process, phenomenon, system, which is a unity of lawfully located and functioning parts. It is assumed that at the input of the "black box" – the input variables $x_1, \dots, x_m$ are measured without interference, and at the output $y = y^0 + \xi$ there is an additive interference ξ .

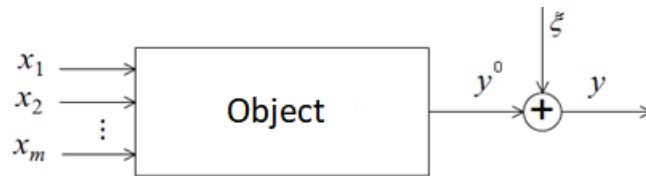


Figure 1.1 – Black box approach in inductive modeling

Now let's consider the data structure for the main problems of inductive modeling. To make the differences clearer, we will give an illustration of the features of using the initial data table in modeling problems using the example of solving them using the method of group argument counting, the data of which are not involved in either the selection of the model structure or the evaluation of its parameters.

Extrapolation	Identification	Forecasting	Recognition																						
<table><tr><td>X_A</td><td>y_A</td></tr><tr><td>X_B</td><td>?</td></tr></table>	X_A	y_A	X_B	?	<table><tr><td>X_A</td><td>y_A</td></tr><tr><td>X_B</td><td>? $f(X)$</td></tr></table>	X_A	y_A	X_B	? $f(X)$	<table><tr><td>X_A</td><td>y_A</td></tr><tr><td>X_B</td><td>y_B</td></tr><tr><td></td><td>?</td></tr></table>	X_A	y_A	X_B	y_B		?	<table><tr><td>X_A</td><td>y_A</td></tr><tr><td>X_B</td><td>y_B</td></tr><tr><td>X_C</td><td>y_C</td></tr><tr><td></td><td>? $f(X)$</td></tr></table>	X_A	y_A	X_B	y_B	X_C	y_C		? $f(X)$
X_A	y_A																								
X_B	?																								
X_A	y_A																								
X_B	? $f(X)$																								
X_A	y_A																								
X_B	y_B																								
	?																								
X_A	y_A																								
X_B	y_B																								
X_C	y_C																								
	? $f(X)$																								

Figure 1.2 – Features of the use of the initial data table in modeling tasks according to GMDH

Fig. 1.2 shows the uniqueness of the forecasting task, in the sense that in order to obtain an accurate forecast of a variable y input variables $x_1, \dots, x_m$, generally speaking, are unknown, are outside the numerical data table. They can be known with a certain probability, including close to zero. Therefore, in order to obtain a real forecast, individual input variables (it is not known in advance which ones) also need forecasting.

Extrapolation is often called a forecast, i.e. having obtained an equation relating the input and output from one sample and using the known values of the input variables of another, the output variable is evaluated with its help and called a forecast. We will distinguish between the tasks of extrapolation and forecasting.

Let us introduce the following definitions:

Extrapolation is the process of obtaining output signals using known input values.

If the model does not have a time shift between the input and output variables, then it is called *static*.

If the output variable in the model has a positive time shift with respect to all input variables, then the solution calculated from such *dynamic* models is called a *forecast*.

Forecasting is the procedure for obtaining the value of the output(s) using the values of completely, partially measured or completely unmeasured (for example, predicted) data.

A *retrospective or ex post forecast* is a forecast based on historical data, when the values of the output variable of the known past are estimated.

The difference between a forecast and an ex-post forecast is significant: a forecast is interesting to all those who want to look into the future, imagine it, characterize it in some way, and a retrospective forecast is important for specialists who construct forecasting models and test their guesses on past data. A retrospective forecast does not differ from extrapolation using a dynamic model in any way, except for the type of the model.

Adaptive forecasting is sequential forecasting that uses the forecast values of the previous forecasting steps to calculate future forecasts. Adaptive forecasting uses the values of partially measured and partially predicted values.

In a forecast problem, compared to extrapolation, it is more difficult to obtain high accuracy. Accordingly, other criteria and justifications are required to assess the accuracy of the forecast.

In the tasks of extrapolation, identification, and retrospective forecasting, the matrix of initial data is used uniformly: on one part of the sample the model is "trained", on the other it is checked.

The difference between the prediction problem and the others is that forecast models derived in this way from measured variable values are used to derive the prediction value beyond the data table on which they are "trained" and validated. In view of the above, two main tasks can be distinguished.

Let us consider in more detail the most important tasks of structural-parametric identification and forecasting.

1.1.1 The Task of Structural-Parametric Identification

Such concepts as model, model structure and class of model structures are given in accordance with the work [13]. A *model* is a function

$$\mathbf{y} = f(\mathbf{X}, \hat{\Theta}_f), \quad (1.1)$$

of the matrix argument $\mathbf{X}$, where $\hat{\Theta}_f$ is the vector of estimates of unknown parameters Θ_f . The column-vectors $\mathbf{x}_i$ of the matrix $\mathbf{X}$ are called *the model arguments*, and the elements of the vector Θ_f are called its *parameters*. The *structure* of the model is a function $f(\mathbf{X}, \Theta)$ specified with an accuracy to the vector of parameters Θ . A *class of model structures* is a set of functions $\Psi = \{f_k(\mathbf{X}, \Theta)\}_{k=1, \overline{K}}$ of a particular type, where K is the cardinality of the set. Examples of such classes are polynomial, difference, trigonometric, and many other sets of classes of functions.

In the group method of data handling, the model that is linear in parameters and nonlinear in input variables is most often considered. An example of this is the polynomial of all variables (so-called Kolmogorov-Gabor polynomial [2]):

$$f(x_1, \dots, x_m, \Theta) = \theta_0 + \sum_{i=1}^m \theta_i x_i + \sum_{i=1}^m \sum_{j=i}^m \theta_{i,j} x_i x_j + \sum_{i=1}^m \sum_{j=i}^m \sum_{k=i+j}^m \theta_{i,j,k} x_i x_j x_k + \dots, \quad (1.2)$$

A special case of the nonlinear model (1.2) is the linear model:

$$y = \sum_{i=1}^m \theta_i x_i. \quad (1.3)$$

Model (1.2) by redesignating the variables is reduced to the form (1.3) with a larger number of input variables $l_p > m$, which depends on the maximum power p of polynomial monomial (1.2), where l_p is calculated using the formula:

$$l_p = C_{m+p}^p.$$

In the problem of identification according to GMDH, the matrix of measurements of the initial data is known $\mathbf{W} = (\mathbf{X} : \mathbf{y})$, $\dim \mathbf{X} = n_W \times m$, $\dim \mathbf{y} = n_W \times l$. It is necessary to find the structure and parameters of the model that meet the maximum accuracy of the specified criterion.

The characteristics of ξ the noise at the output of the object is unknown. It is assumed that it is not correlated with the desired signal. Fig. 1.3 shows the interrelations of the main stages of the solution. Based on a priori knowledge about the object, a class of functions Ψ is given, to which the structure of the desired model belongs, according to assumption. A procedure for gradual complication (generation) of structures in a given class of iterated models is created. Vectors of parameters Θ_k and outputs $\hat{\mathbf{y}}_k$, $k = \overline{1, K}$ are evaluated.

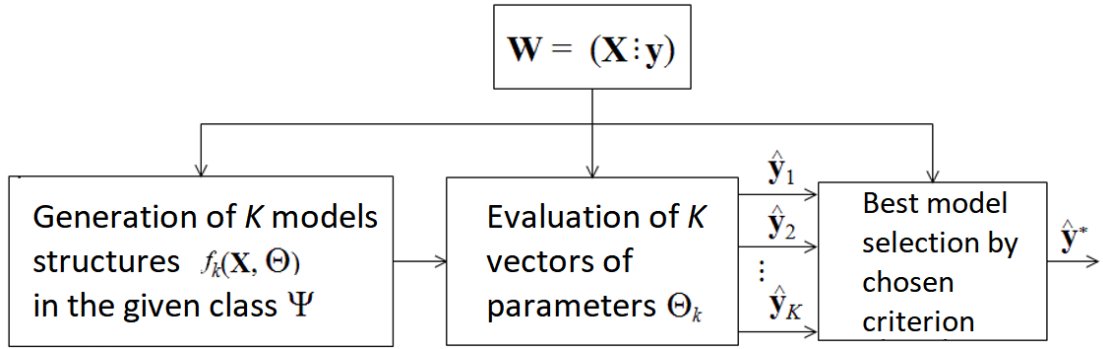


Figure 1.3 – Scheme for the identification problem solution.

According to the appropriately chosen criterion, the best model is determined, denoted by

$$\hat{\mathbf{y}}^* = f(\mathbf{X}, \hat{\Theta}_f^*) = \sum_{i=1}^{s^*} \hat{\theta}_i^* \mathbf{x}_i, \text{ where } s^* \text{ is the minimum complexity (the number of arguments not}$$

equal to zero) of the model, at which the minimum of the specified criterion is achieved.

In the problem of structural-parametric identification, two subtasks are possible. The first is the search for the so-called "true" (unbiased) model, the second is the finding of an "adequate" model [3].

Several definitions are known for a "true" model. In [3], p.108 it is said: "a true model is a model at the output of which relatively little noise is additively superimposed – a high-frequency interference with a zero average". A mathematical definition of the unbiased ("true") structure of the model can be found in [14], p.52: "Let f^0 us call the structure an unbiased estimate $\hat{f}$, which depends on the parameter θ , if $f^0(x, M[\hat{\theta}]) = f(x)$, where $M[\cdot]$ is the mathematical expectation. The definition of a "true" (unbiased) model is also given in the article [15] "... is an exact (physical) model of the object $\mathbf{y}^0 = \mathbf{X}^0 \theta_0$ ", where " $\mathbf{y}^0$ is the exact (non-noisy) output of the object" [16]. It is assumed that the output of the object is superimposed by $y = y^0 + \xi$ independent noise ξ , and there exists θ_0 an exact (unknown) vector of parameters with "true" arguments $x_1, \dots, x_{s_0}$ where s_0 is the number of "true" arguments.

Taking into account the last two definitions, let us call the "true" structure f^0 , which is the unbiased estimate of the function f , which depends on the parameter vector Θ and the input matrix $\mathbf{X}$, if $f^0(\mathbf{X}, M[\hat{\theta}]) = f(\mathbf{X})$.

In this case, the structure f^0 has a complexity: $s_0 = m - \sum_{i=1}^m \text{Equels}(M[\hat{\theta}_i], 0)$, where $\hat{\theta} = (\hat{\theta}_1, \dots, \hat{\theta}_m)$. The comparison function $\text{Equels}(M[\hat{\theta}_i], 0)$ is equal to 1 if the mathematical expectation of any parameter $M[\hat{\theta}_i]$ is 0.

The "black box" approach to finding the "true" model in the absence of noise at the output is presented in Fig.1.4.

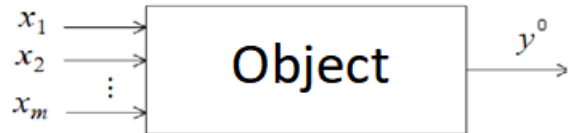


Figure 1.4 – Approach to identifying the true model

The study of the convergence of the solution to the "true" model (linear and nonlinear) is described in detail in Section 4.1 of Chapter 4.

In the case of a polynomial model, a high correlation between the regressors also leads to a reduction in the model, and the model "gets rid of" the dependent variables. If the noise level is low and the number of observation points is large enough, the "adequate" model coincides with the "true" one. In other words, the complexity of the "adequate" model is less than or equal to the complexity of the "true" model ($s^* \leq s_0$) obtained with the same input variables and the output variable $\mathbf{y} = \mathbf{y}^0$. These facts of the study on structural-parametric identification, described in Section 4.2, confirm the theoretical results [111,16].

An important role is played by the criteria according to which the model becomes adequate to noise. The adequacy of the regularity criterion [78] was considered in the theory of noise-resistant modeling in [15], and its numerical comparison with other criteria of structural-parametric identification was carried out in [17]. An *adequate* criterion is a *criterion* that takes into account the presence of noise in the initial data when construction models. At the same time, the dispersion of noise can be included explicitly or implicitly in the formula of the criterion. For example, the Mallows criterion C_p and the regularity criterion $AR_{B/A}$ are adequate, and the Akaike criterion AIC - not. The emphasis on the identification of models with an "undisplaced" structure is made in [14], [18], and on the identification of "adequate" models in [9], [19]. In [9], the analogy between the "true" model and the "true" signal is noted, and the similarity of the problems of inductive (noise-resistant) modeling and Shannon's theory of communication is analysed. The "black box" approach to identifying an "adequate" model is shown in Fig. 1.1.

For what tasks, and in which applications, is it not necessary to obtain a "true" model?

Obviously, in conditions of various kinds of uncertainties. For example, when the variables are partially unobservable or the noise level in the experimental data is unknown, construction "adequate" models leads to fewer errors outside the data table, as they tend to be simpler than "true" models. In the problem of recognition (when construction classifiers for various types of diagnostics) and in predicting the behaviour of poorly studied, complex noisy objects and processes, the type of model is not important, but the accuracy of the result is important.

Answers to the following questions are also important:

- 1) When is a true model needed?
- 2) At what noise parameters and other characteristics of the initial data is it possible to identify the true model?

First, the identification of the "true" model is necessary to verify the operation of iterative algorithms and is used to confirm the fact of its convergence, as well as to study the rate of convergence of the algorithm. In addition, the "true" model is of great importance in many applied tasks, such as, for example, the restoration of a true signal cleared of interference; the discovery of regularities that reflect the physics of the object.

The answer to question 2) is contained in Chapter 4, where the internal convergence of the structure is computationally proved, an analytical form of the dependence of the convergence rate on the complexity of the model is obtained, provided that the input variables are uncorrelated and there is no noise in the output data. In subsection 4.3, the convergence to the "true" model is demonstrated using numerical modeling examples, provided that there is noise in the data, both at the input and output of the object. In the examples considered, the relaxation iterative algorithm "discovers the true" structure of the model even at 10-20% noise level at the output.

If the variance is unknown and sufficiently large, then modeling according to GMDH leads to the construction of a model structure adequate to the critical level of variance [15].

1.1.2 Approaches to solving the problem of forecasting

The conditions of the problem are the same as for identification: the measurement matrix is known, $\mathbf{W} = (\mathbf{X}; \mathbf{y})$ $\dim \mathbf{X} = n_W \times m$, $\dim \mathbf{y} = n_W \times 1$.

In what follows, we will mainly talk about retrospective forecasts, calling them simply forecasts. The result of the forecast passes into the ex-post forecast, at the moment when the future becomes the past and the real value of the output value becomes known (measured), when it is possible to assess the accuracy of the forecast by comparing the predicted and real values.

Let us compare two approaches to solving the forecasting problem, illustrated in Figure 1.5. Let us conventionally call them the *point* and *interval* approaches. They differ in the methods of estimating parameters and initial assumptions.

The first approach, related to the solution of algebraic equations, is used in the inductive method of model self-organization [3]. The second approach is related to the construction of interval systems of algebraic equations [70], [71]. Obtaining a model structure with fuzzy parameter values is often called fuzzy GMDH (FGMDH) in the literature [72], [73].

In both approaches, it is assumed that the predicted process is described by a linear regression model, for example, in the form of a difference equation:

$$y_{k+1} = \sum_{j=0}^{m_1} \alpha_j y_{k-j} + \sum_{j=0}^{m_2} \beta_j z_{k-j},$$

where z_{k-j} , $j = \overline{0, m_2}$, y_{k-j} , $j = \overline{0, m_1}$ are lag input and output variables; m_1, m_2 – maximum number of lags (delays); α_j, β_j – parameters that in FGMDH belong to the range of values with a certain measure of affiliation. Let's redesignate z_{k-j} , y_{k-j} as x_i , $i = \overline{1, m}$, $m_1 + m_2 + 2 = m$, parameters α_j, β_j as θ_i , omit the index k and all lag indices. Then we will get a difference model in a simple form:

$$y_{+1} = \sum_{i=1}^m \theta_i x_i, \quad (1.4)$$

where y_{+1} – model output. The index "+1" means that in order to make a forecast, it is necessary to construct a model of the values of the output 1 step ahead in relation to the previous moment of the variable measurement k . In the model (1.4), all y_{k+1} values of forecast of output value of the moment in time $(k+1)$ are related to variable x_i , $i = \overline{1, m}$, that is known at preceding $(k+1)$ moments of time.

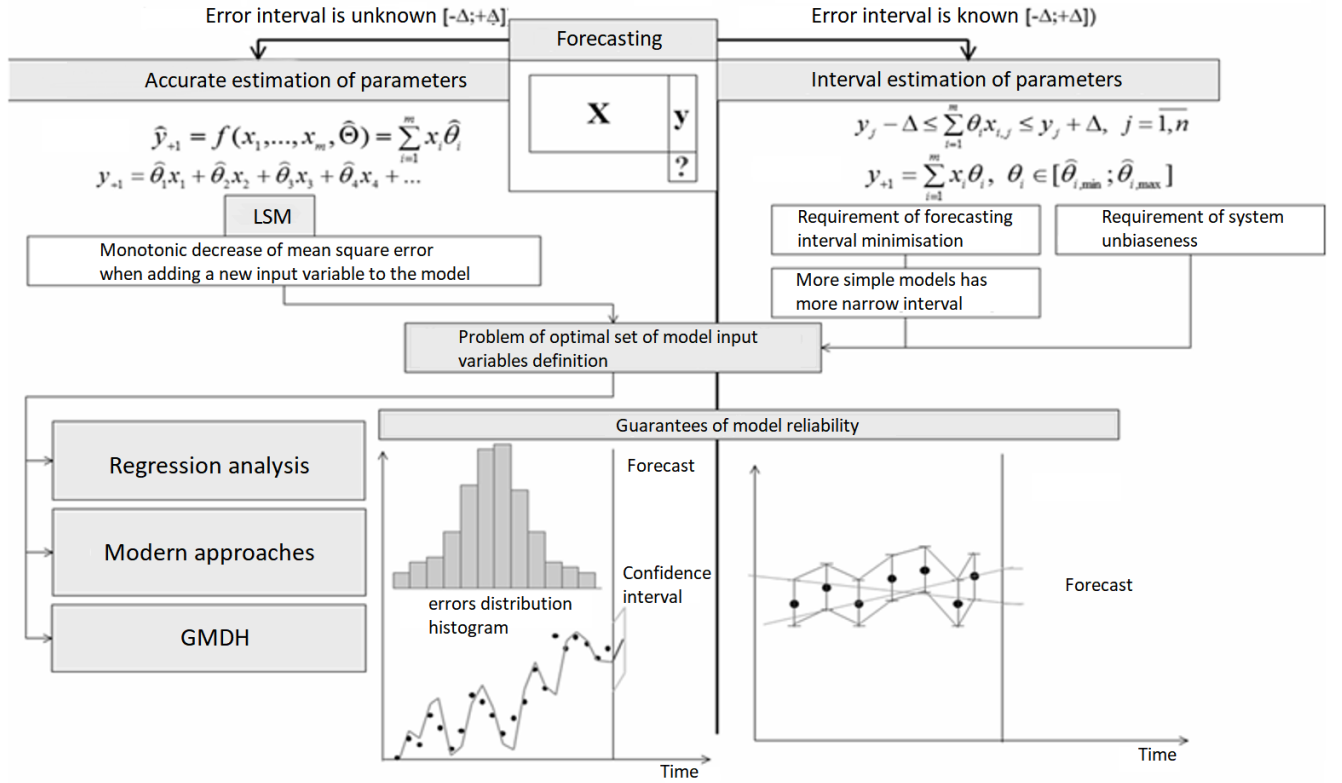


Figure 1.5 – Two approaches to solution of the forecasting problem

The two approaches in the original formulation differ in information about the dispersion of noise σ^2 , more precisely about the standard deviation of σ . In the point approach, it is unknown, in the interval approach – it is considered known. In the second approach, the interval of change in the standard error 2Δ of the retrospective forecast for the entire sample of observations is specified. In the first approach, the estimation of the parameters θ_i is pointed, in the second – interval. In both cases, the value of the forecast error outside the table of data used to obtain the model is minimized:

$$err_C = \sum_{i=1}^{n_C} |y_i - \hat{y}_i| \rightarrow \min ,$$

where n_C is the number of lead points (forecasting steps).

Let us confine ourselves further to an example of the interval approach, when the belonging function for all model parameters in the range of values is a constant, without complicating the presentation by introducing special belonging functions for coefficients, as is customary in the fuzzy GMDH.

In point estimation, the parameters are determined by solution a system of equations, for example, using the method of least squares (LSM), and in interval estimation, as a result of solution systems of inequalities:

$$y_{i+1} + \Delta \leq \sum_{j=1}^m \theta_j x_{j,i} \leq y_{i+1} - \Delta, \quad i = \overline{1, n_W - 1}, \quad \theta_j \in [\hat{\theta}_{\min}, \hat{\theta}_{\max}].$$

The solution of the system of inequalities under the condition of minimizing the error function is reduced to the solution of the linear programming problem [20] (1939), [21] (1960). The condition for the existence of a solution is the consistency of the system of resulting equations and inequalities.

In the point approach, if the entire sample is used, then as the model becomes more complex, the modeling error is monotonically reduced $err_W = \sqrt{\frac{1}{n_W - 1} \sum_{i=2}^{n_W} (y_i - \hat{y}_i)^2}$ (provided $n_W > m$).

If $n_W = m$, then the error for n_W points is equal zero. Identifying the model in this case leads to data interpolation.

Using the example of model identification

$$y_i = a_1 t_i + a_2 t_i^2 + \dots + a_m t_i^m, \quad i = \overline{1, n_W}, \quad n_W = 7.$$

The interpolation is illustrated in Figure 1.6 (a). In the case of $m = n_W$ ($m \geq s_0$), in the presence of noise at the output, the model becomes overcomplicated and unsuitable for forecasting, because it has a large prediction error outside the data table. When approximating the same data with a polynomial of a smaller order $m < n_W$, the standard deviation of the model from the y_i values at points $i = \overline{1, n_W}$ is minimized, assuming that the output of model is noisy. The error of the approximation model outside interval of model construction (which in the example is defined by values at seven points) is less than the error of the interpolation model, see Fig. 1.6 b).

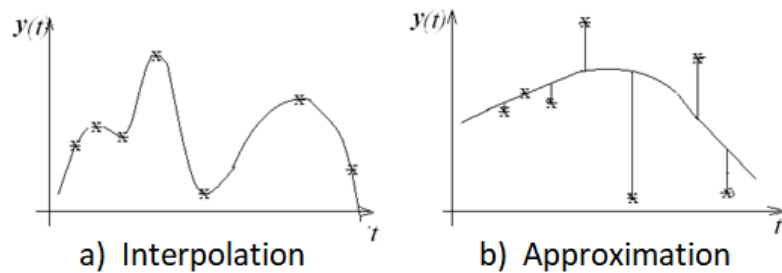


Figure 1.6 – Examples of identification results: a) data interpolation, $m = n_W$;
b) data approximation, $m < n_W$.

In the interval approach, simpler models have a narrower range of prediction error, which is illustrated by the graph in Figure 1.5, but with a narrowing interval, the feasibility of the conditions for the joint solution of the system of equations is less and less guaranteed.

Thus, in both approaches, the task arises of determining the optimal set of arguments and a class of models to find a solution that satisfies the conditions of forecast accuracy.

We will describe what methods exist for solution the problem using a point approach, in which, after obtaining the model, histograms of the distribution of errors of the received forecasts can be created and confidence intervals of errors can be found.

1.1.3 Group Method of Data Handling as a part of mathematical methods

In the introduction to this chapter, the main methods of mathematical modeling were briefly characterized.

On the general scheme of classification of modeling methods, presented in Fig. 1.7, deductive and inductive methods are distinguished. We will not dwell in detail on the varieties of deductive methods, but only list them. According to [8], the following methods of describing an object can be distinguished in deductive methods: analytical, imitative and combined.

In *analytical* modeling, mathematical (abstract) models of a real object are studied in the form of algebraic, differential and other equations or logical conditions in order for the modeling process to lead to an exact solution.

In *simulation* modeling, the process of system functioning is reproduced in the form of separate algorithms, and elementary events that make up the process are simulated, while preserving their logical structure and sequence of time flow. Such an algorithm makes it possible to obtain information about the state of the process at specific moments of time based on initial data, which makes it possible to assess the characteristics of the system as a whole.

Combined (analytical-simulation) modeling in the analysis and synthesis of systems allows you to combine the advantages of analytical and simulation modeling. When during construction of combined model, a preliminary decomposition of the object's functioning process into component subprocesses is carried out, and for those of them, where possible analytical models are used, and for the rest simulation models are constructed.

Inductive methods use a minimum amount of a priori knowledge and are aimed at construction patterns and systems of inter-connections (inter-dependencies). Recently, there has been a popular opinion that there are no inductive methods that do not contain deductive features. In other words: all inductive methods are deductive, and vice versa. Nevertheless, we use already established terminology.

According to the direction of applications, inductive methods can be divided into two groups:

1) Inductive methods in artificial intelligence (AGI) systems, which are based on the causal completeness and consistency of judgments. For example, these include the JSM method of automatic hypothesis generation, described in [7].

2) Methods of inductive modeling, which are used in information systems for processing and analysing data based on empirical measurements of data.

The first group of inductive methods is used to draw conclusions about new properties of a system of objects and uses an assumption about the possible causes of individual properties of objects. The second is to identify explicit and implicit connections and patterns hidden in the data, and to present them in the form of mathematical models – dependencies of the output variable(s) on the input variable(s).

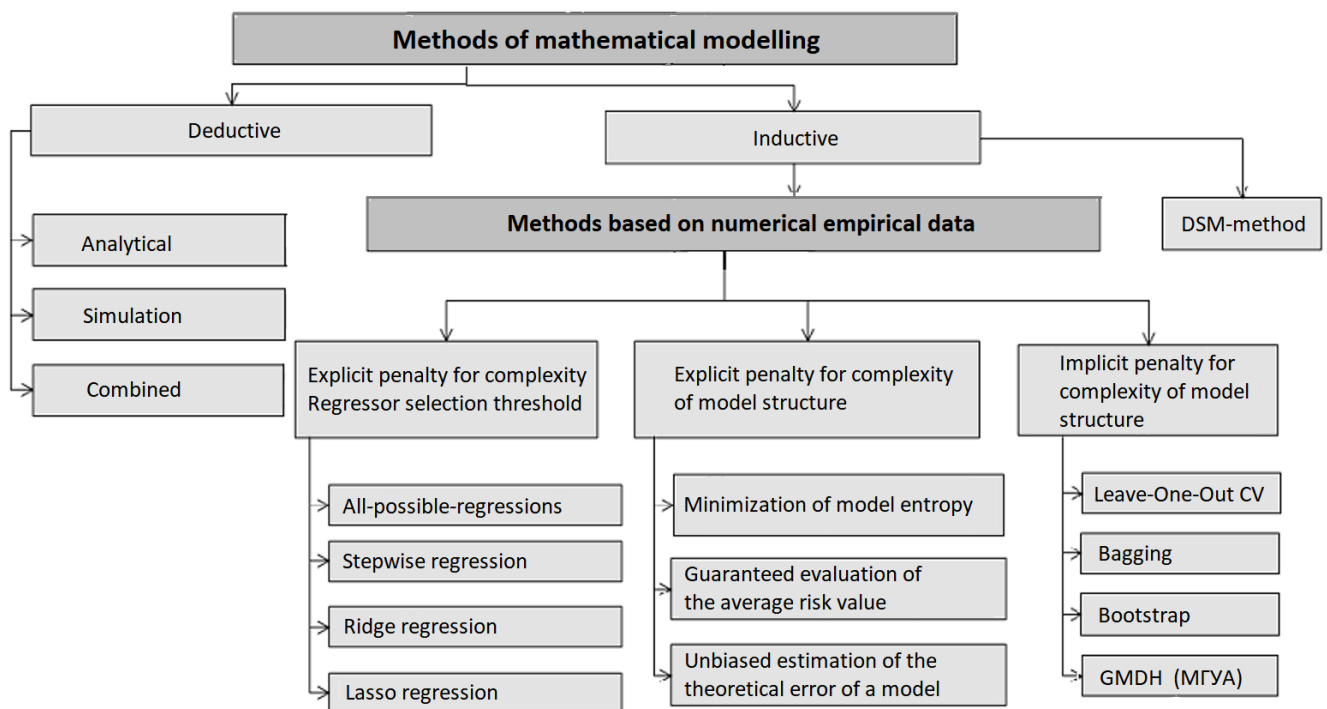


Figure 1.7 – Classification of modeling methods

Let us dwell in more detail on the group of methods of inductive modeling based on empirical data from observations. These methods are mainly used to construct regression equations that are linear in terms of parameters. A variety of inductive modeling methods based on numerical empirical data can be conventionally divided into classical, modern and non-traditional methods according to the method of determining the structure of the regression model [13].

Classical methods use the statistical theory of hypotheses, where the value of the specified levels of significance of a particular regressor is arbitrary (not arising from anything) and often subjective. This group includes, in particular, the method of all regressions, the methods of inclusion, exclusion and inclusion-exclusion used in classical regression analysis to solve the problem of finding the best regression [22] (1973), [23] (1980). models in the selection of regressors according to the threshold value of the Fisher criterion.

The most well-known *modern* methods are: the method of minimizing the entropy of the model [24], the method of guaranteed estimation of the value of the average risk [25] and the method of unbiased estimation of the theoretical error of the model [26], which also use an explicit penalty for the complexity of the regression model, since complexity (the number of regressors) is clearly included in the formula of the "internal" criterion. The "internal" criterion uses the information on the basis of which both the structure and parameters of the models are obtained [14]. The classic "internal" criterion is the *residual* sum squares (*RSS*) criterion calculated on the entire sample. In the Akaike, Mallows, and Schwartz criteria, *RSS* is used as part of an explicit model complexity penalty.

The group of inductive methods, which uses criteria with an implicit penalty for the complexity of the model when choosing the best structure, is based on cross-validation of the model on new data. Various modifications of the cross-validation method [27] are widely known in foreign literature under different names: leave-one-out [28] (1948), jack-knife [29] (1958), [30] (1981), bootstrap [31] (1979) and bagging [32] (1994). The group method of data handling (1968) [2], which belongs to this group, has received the greatest development and application. A.G. Ivakhnenko in GMDH combined the ideas of evolution and selection with "cross-validation" of the model on different samples: the parameters are estimated using one sample, and the best structure is selected based on another, using information from observations that were not involved in the estimation of its parameters. The fact that the selection criterion is based on new information indicates some objective mechanism for determining the structure of model. The selection criterion is called "external" criterion, and models obtained using such a criterion have minimal extrapolation error variance and are noise-immune [15], which is a necessary condition for solving applied modeling problems.

1.2 GMDH as an inductive method of heuristic self-organization of models

The group method of data handling, or the same inductive method of self-organization of models, and the same heuristic self-organization of models, was developed by Academician of the National Academy of Sciences of Ukraine A.G. Ivakhnenko [1]. The basis is empirical data and the principle of external complement in the choice of a mathematical description of input-output dependencies.

GMDH's algorithms make it possible to automatically find interdependencies in data, construct models of optimal complexity, and select relevant arguments (features) from an excessive set of arguments. To date, many varieties of GMDH algorithms of sorting-out and iterative types have been developed. Computer technologies of heuristic self-organization make it possible to solve problems of increasing variety, complexity and dimensionality, to conduct computational proofs that increase the reliability of the results. The group method of data handling, as a competitor to stochastic approximation, which arose in 1968, developed as a method for solution scientific and practical problems, which was reflected in the titles of books. Self-learning recognition systems [33], heuristic self-organization systems [1], inductive method of model self-organization [3], noise-resistant modeling [9], modeling based on experimental data [14], sorting (or enumeration) modeling methods [34], modeling using inductive training algorithms [11]. All stages of the development of GMDH were the result of reflection and summation of the experience of formulating and solving various

problems, such as: automatic control and recognition [33], machine thinking [35], prediction of random processes [36], pattern recognition [37], long-term forecasting and control of complex systems [10], decision-making [38], multivariate modeling [39], clustering [34], synthesis of algorithms for modeling complex systems [11], [40].

A detailed story about all practical tasks where GMDH is used is the topic of a separate study. A far from complete list of the main areas of application of GMDH before 2000, which includes 16 areas with the names of applied works and implementations, is available on the website [95]. New areas and recent applications of GMDH include medical diagnostics (the approach and results are described in practical appendix 2 of chapter 5); analysis of texts and images (works [105-108] were published in the proceedings of the 4th International Conference on Inductive Modeling ICIM'2013).

Problems with large amounts of initial information were not the scope of application of GMDH, since there were no appropriate algorithms for solving them. An example of an applied problem described by a matrix with large dimensions is, in particular, the problem of forecasting "space weather", the number of observations in which is measured in hundreds of thousands. The results of its solution are discussed in section 5.1 of chapter 5. But even with a relatively small amount of data, for example, in problems with medical topics, increasing the speed of algorithm calculations is an urgent task.

Alexey Grigoriyevich Ivakhnenko is rightly called the inventor of algorithms. He himself gave rise to this with the title of the monograph [51] "Self-organizing methods in modeling: GMDH type algorithms", published for his 70th anniversary. Let us determine the areas of application in accordance with the volume of initial information of the main types of GMDH algorithms, without underestimating their advantages.

Sorting algorithms of GMDH construct models in such a way that the computational complexity of main sorting stage of model construction does not depend on the number of observations. Their disadvantage is the exponential increase in the computational complexity of the algorithm with an increase in the number of arguments, so the use of such algorithms is possible only with a small number of features (up to 28), in contrast to iterative algorithms.

Classical iterative algorithms, in contrast to sorting algorithms, make it possible to solve problems with a larger number of arguments (up to 300) in an acceptable time (up to 2 minutes), due to a significantly smaller enumeration of model variants. In turn, all available iterative algorithms have a significant drawback – the computational complexity of the model construction stage depends on the number of observations. The complexity is not lower than quadratic, but this complexity is too high for solving practical problems of large dimensions.

The subclass of iterative algorithms – *relaxation iterative* algorithms are the most efficient, fast-acting among all known algorithms of GMDH, and suitable for solving problems with an even larger number of arguments (up to a thousand). For these algorithms, convergence by the "internal" criterion has been theoretically proven.

1.2.1 General Formulation of the Problem of the Inductive Method of Self-Organization of Models

According to [13], the general formulation of the problem of inductive modeling is as follows. Given a data matrix $\mathbf{X}$, $\dim \mathbf{X} = n_w \times m$, where m is the number of input variables, n_w is the number of observations (points). The output vector $\mathbf{y}$, $\dim \mathbf{y} = n_w \times 1$ is assumed to be noisy by an additive uncorrelated interference ξ with an unknown distribution law and zero mathematical expectation $E(\xi) = \mathbf{0}_{n_w}$, where $\mathbf{0}_{n_w}$ is a vector of n_w zeros and a diagonal matrix of noise covariance $E\xi_i\xi_k = 0$, $k \neq i$, with limited variance $E\xi_i^2 = \sigma^2 < \infty$. Let be given a class of model structures Ψ .

It is needed to find the optimal model $\hat{\mathbf{y}}_f^* = f^*(\mathbf{X}, \hat{\Theta}_f)$ for minimum of the specified criterion CR as

$$f^* = \arg \min_{f \in \Psi} CR(\mathbf{y}, f(\mathbf{X}, \hat{\Theta}_f)), \quad (1.5)$$

where the estimation of parameters vector $\hat{\Theta}_f$, $\dim \hat{\Theta}_f = s_f \times 1$, for each $f(\mathbf{X}, \Theta_f) \in \Psi$ is solution of the problem:

$$\hat{\Theta}_f = \arg \min_{\Theta_f \in \mathbb{R}^{s_f}} QR(\mathbf{y}, f(\mathbf{X}, \Theta_f)), \quad (1.6)$$

$QR(\cdot)$ is a criterion for quality of solving the problem of parameters estimation,

$CR(\cdot)$ is a criterion for quality of solving the problem of choosing an optimal model.

The QR criterion, that is used to estimate parameters of the model with a given structure is usually called an *internal* criterion, according to GMDH theory of modeling. Criterion for choosing structure of CR model is called *external*, since it uses new ("external") information, that is not used to evaluate parameters of the model.

1.2.2 Tasks of the inductive method of model self-organization

Let's distinguish two NP-complete problems of inductive modeling: 1) the problem of finding the structure and parameters of a model corresponding to the global extremum of the performance functional (maximum solution accuracy), and 2) the problem of finding an optimal sample partition. These two problems have much in common in their solution approaches, but there are also differences that we would like to highlight. In model search, the structural-parametric identification problem is the primary one, and formally, the second problem is a consequence of the first. Conversely, if the problem of classification of observations is important, i.e., the problem of partitioning sets takes precedence, then the problem of constructing the relationship functions of regressors takes second place. In fact, in GMDH, both problems are equally important and computationally complex. In practice, the problem of finding a model for the global extremum of a functional is limited by the number of arguments, while the partition (classification) problem is limited by the number of observation points.

1.2.3 Basic Principles of GMDH

The construction of models using GMDH is based on three simultaneously applied principles: *gradual complication of the model structure*, *"external supplementation"* [41], and *inconclusive solutions*.

The author of GMDH borrowed the idea for the method of model complication from nature (idea was inspired by a nature). The structure of GMDH algorithms for searching of mathematical descriptions of objects has much in common with F. Rosenblatt's perceptron [76]. It is in his works [76, 77] that A.G. Ivakhnenko refers in his first monographs on GMDH. In [3], references appear to theoretical results of Russian authors from 1957-1958 [42], [43]. Their works provide a rigorous mathematical justification for both the representation of a complex structure as a superposition of individual structures and the ***principle of successive increase of complexity***.

The principle of gradual complication of the model structure is also based on the idea of self-organization, which consists in construction a model of such complexity that is necessary for an adequate description of the object in the sense of the minimum of the specified criterion CR of model quality. The model structure generation algorithm performs a layers of successive iterations $r = 1, 2, \dots$ - each of which generates candidate-models with a structure more complex than previous

ones. Iterations are carried out until the value of criterion CR_r decreases ($CR_r < CR_{r-1}$) or reaches a value that specifies the predetermined accuracy of the problem solution. It is considered that when the criterion begins to increase

$$CR_r > CR_{r-1},$$

The model structure of the r -th layer becomes overcomplicated, and the process stops.

The principle of "external supplement" is based on Gödel's incompleteness theorem [45]. The "external complement" principle is based on Gödel's incompleteness theorem [45]. According to this principle, criteria based on the use of new information and having a minimum with an increase in the model complexity, should be used as a CR criterion [3].

The principle of inconclusive solutions [46] in GMDH is that not one, but several candidate-models, selected at the current iteration, are transferred to the next iteration of the algorithm. This makes it possible to increase the accuracy of models and the probability of finding a global minimum of the CR criterion.

1.2.4 The task of finding a model that corresponds to the global extremum of the quality functional

Suppose a matrix of source data is given $W = (X : y)$. It is necessary to construct an object model, in the form (1.3), which provides a minimum of CR function

$$f^*(X, y) = \arg \min_{f \in \Psi} CR(X, y).$$

The parameters included in this function are linearly determined by LSM

$$\hat{\Theta}_i = (X_i^T X_i)^{-1} X_i^T y, \quad i = 1, 2, \dots, (2^m - 1), \quad \dim(X_i^T X_i) = s \times s, \quad s = \overline{1, m}. \quad (1.7)$$

The compact notation for LSM parameters estimation (1.7) has a theoretical significance, and its own features in practical implementation in the GMDH algorithms. The problem of finding a model that corresponds to the global extremum of the quality functional can be solved by various algorithms. A complete sorting of structures is carried out in the method of all regressions [22] (1973). In the theory and practice of inductive modeling, recurrent methods have been developed for obtaining such estimates during consistent increase of model's complexity. GMDH's algorithms differ, firstly, in how the process of model complication is arranged and what formulas are used to calculate parameter estimates.

The second feature of GMDH is the use of the external criterion CR in the construction of models. This criterion operates with the results of calculations on two samples. The external criteria used in GMDH are based on dividing the initial set of observations W into at least two subsets A and B : $A \cap B = \emptyset$, $A \cup B = W$. As already mentioned in subsection 1.1.2, a larger number of subsamples are possible when solving forecasting and recognition problems. This will be shown in Chapter 5 also.

The best solution is determined by the minimum of external criterion CR .

The varieties of the GMDH external criteria are described in detail in [3], and their canonical forms are described in [14], [104]. Taking into account the partition, the matrix $(X : y)$ consists of two submatrices:

$$(X_A : y_A), \quad \dim(X_A : y_A) = n_A \times (l + 1) \text{ and } (X_B : y_B), \quad \dim(X_B : y_B) = n_B \times (l + 1).$$

One of the first external criteria was the regularity criterion – hold-out error on deferred data [78], 1969:

$$AR_B = AR_{B/A} = \|\mathbf{y}_B - \mathbf{X}_{B,i} \hat{\Theta}_{A,i}\|^2, \quad i = \overline{1, I}, \quad (1.8)$$

where $\|\cdot\|^2$ denotes the quadratic norm of the vector.

The parameter vector is determined from the LSM on sample A

$$\hat{\Theta}_{A,i} = \arg \min_{\Theta_{i,A} \in \mathbb{R}^{s_i}} \|\mathbf{y}_A - \mathbf{X}_{i,A} \Theta_{i,A}\|^2, \quad (1.9)$$

where s_i is the number of arguments (complexity of the structure) of the model.

Indices A and B in (1.8), (1.9) indicate from which parts of the samples the corresponding values are formed and calculated. Let us describe the main approaches to dividing the initial set of observations.

1.2.5 The problem of partitioning a set of observations

In the first work of A.G. Ivakhnenko (1968) on the topic of GMDH, it was said that "classification ... is the division of initial variables (with almost any number of them) into two classes: "good" and "bad" or into "useful" and "harmful" [2]. With the formulation of the criterion for the selection of models [78], which implements the "principle of external complementation" at GMDH, the task arose of dividing the sample - "classifying" of observation points. In one of his later works [79], A.G. Ivakhnenko formulates a two-dimensional analogue of the rows division problem: the problem of selecting the "core" – the optimal size of the "sliding window" in the form of a square data matrix (where $m = n_A$) to determine the best adaptive, predictive model. Next, we consider only one type of sample division – row partition.

Not much work has been devoted to a detailed study of the problem of sample partition, which indicates both the difficulty of its analysis and the fact that, with some experience in the application of GMDH algorithms, simple computationally inexpensive methods of partitioning are usually used in practice.

Usually, when forecasting, the initial sample is divided into parts in chronological order. Early observations make up a training subsample, and later observations make up a test and examination sample. If it is necessary to approximate or classify data, then "dispersion division" is used; if you test the GMDH algorithm, then a random multiple partition is used, and the result is averaged. Traditionally sample can be divided by ratio $n_A : n_B = 2:1$.

Random partitioning also makes sense in the construction of static models of an object, since it is believed that all the relationship equations in the so-called system of conditional equations are equal and equally likely can be attributed to both the training subsample and test subsample [1]. In forecasting problems, due to the obsolescence of information and its often huge volume, the natural breakdown is the breakdown in chronological order. This principle of "window" data processing is borrowed from the tasks of tracking control. In forecast models, the structure and parameters are calculated on the basis of time-limited observation data [10].

In the so-called "variance" division [49], the semantic content of variables is abstracted (abolished) and only their statistical nature is taken into account. In this partition, each observation of the object is characterized by the sum of the squares of the deviations of the variables from their mean values ("variance"). The entire set of observations is ranked according to this parameter and divided into two parts in two ways: to get 1) a "variance-like" partition and 2) a "variance-dissimilar" partition.

In [50], a constructive proposal is given for partition the sample, related to the concept of variance as the value of the standard error of modeling. Since this error is calculated uniformly on both training and test subsamples, it is natural to require the partition to be such that the calculated variance is minimal on the entire sample, and the model structure is the same on both subsamples.

Quasi-optimal (ρ^2 -proportional) sample partitioning embodies exactly this idea: for the minimum criterion of the total quadratic error in the assumption that the additive noise at the output has zero mathematical expectation and finite variance σ^2 , the condition of the quadratic dependence of information matrices on both samples is obtained:

$$\mathbf{X}_A^T \mathbf{X}_A = \rho^2 \mathbf{X}_B^T \mathbf{X}_B, \quad (1.10)$$

where ρ is a certain number, $\dim \mathbf{X}_B^T \mathbf{X}_B = \dim \mathbf{X}_A^T \mathbf{X}_A = m \times m$. In problems with empirical data, the relation (1.10) is not fulfilled. Therefore, the search for a quasi-optimal partition is carried out by minimizing the norm of the two matrices:

$$\min_{\rho \neq 0, \ell=1, L} \left\| \mathbf{X}_A^T \mathbf{X}_A - \rho^2 \mathbf{X}_B^T \mathbf{X}_B \right\|, \quad (1.11)$$

To assess the norm in [50], the following criterion is proposed:

$$H = \min_{\ell=1, L} (\rho_{\max \ell} - \rho_{\min \ell}) / (\rho_{\max \ell} + \rho_{\min \ell}), \quad (1.12)$$

as a result of minimizing of which the optimal sets of observation points and their number in subsamples A and B are found. This criterion assumes that the following values are positive:

$$\rho_{\max} = \max_{i,j} \rho_{ij}, \quad \rho_{\min} = \min_{i,j} \rho_{ij}, \quad \rho_{ij} = \chi_{ij,A} / \chi_{ij,B}, \quad i, j = \overline{1, m}, \quad j \geq i$$

$$\begin{aligned} \chi_A^\Delta &= \mathbf{X}_A^T \mathbf{X}_A, \quad \chi_B^\Delta = \mathbf{X}_B^T \mathbf{X}_B, \quad \dim(\chi_B) = \dim(\chi_A) = m \times m, \\ \dim \mathbf{X}_A &= n_A \times m, \quad \dim \mathbf{X}_B = n_B \times m, \quad n_A + n_B = n_w. \end{aligned}$$

Total number of options for sorting through different sets of points

$$L = \sum_{i=1}^{\lceil n_w/2 \rceil} C_{n_w}^i = 2^{n_w-1} - 1,$$

where $\lceil n_w/2 \rceil$ is rounding to the nearest whole number with excess.

Thus, taking into account the calculation of the above sample split, the problem of construction a model has hyper-exponential computational complexity. The second chapter of the book is devoted to the presentation of problems and the results of research related to the division of the sample.

In many applied problems with small data samples, in cases where it is difficult to construct a model of satisfactory accuracy, algorithms for partition the original sample are used. One of these effective algorithms is the quasi-optimal partitioning algorithm, which searches for the best partition by completely sorting the options for partition observation points. An example of solution such a problem is described in Section 5.2 of Chapter 5.

1.3 Basic algorithms of GMDH

During solution of real problems, the result of modeling depends on algorithm for constructing the model structure. In combinatorial algorithms, all possible partial models from a given class (polynomial, difference, harmonic) are sorted and one or a set of the best ones is selected according to a given selection criterion. If the coefficient estimates are determined by least squares in a finite number of steps, then such an algorithm is called a *sorting* algorithm.. The literature on GMDH describes in detail a harmonic algorithm with analytical selection of non-multiple frequencies [109] (1975), which is classified as sorting-out, and a polyharmonic algorithm for the identification of multidimensional fields [100, 110], (1980, 1982).

We will limit ourselves to the presentation of the combinatorial algorithm that determines the structure of the model by a complete sorting of the polynomial model (1.3), which was developed in connection with the use of parallelization [113] (2008) and recurrent calculations [17] (2009) in combination with parallelization [114] (2013).

Let us consider algorithms in accordance with the scheme of their classification presented in Fig. 1.8.

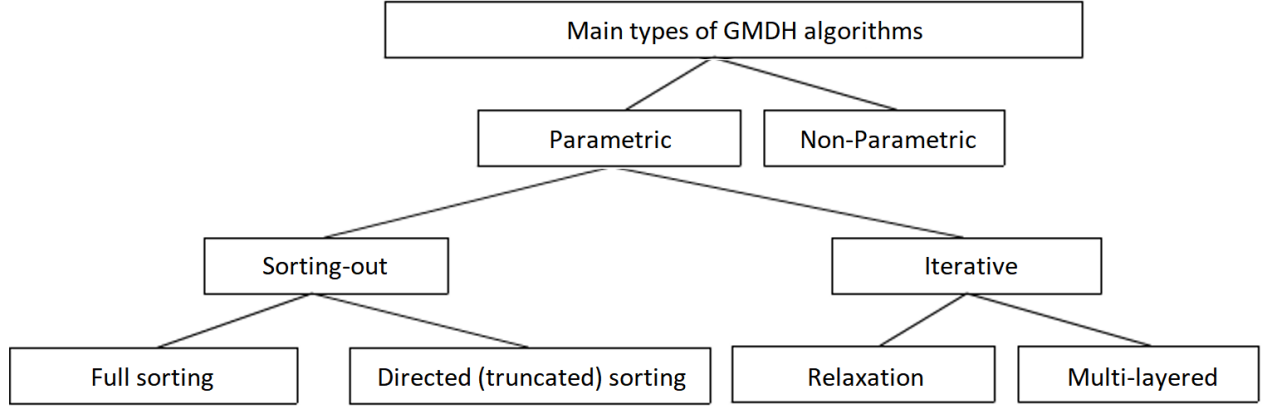


Figure 1.8 – Classification of the main types of GMDH algorithms

1.3.1 Sorting-out GMDH algorithms

Let's describe sorting-out algorithms and highlight their main positive qualities. GMDH's sorting algorithms were created to obtain a model that corresponds to the global extremum of the external criterion among the models in a given set of their descriptions.

The classical *algorithm of complete sorting*, that constructs models of the form (1.2) is the combinatorial algorithm of GMDH COMBI [52-54]. Combinatorial sorting of model structures was discussed in subsection 1.2.4 in connection with the task of searching for a global extremum. COMBI refers to sorting algorithms for the inclusion of arguments. In [53] (1981), in order to accelerate the work of the combinatorial algorithm, it was proposed to use the well-known recurrent method of estimating parameters – the matrices framing method [55] (1963), the features of which were organically included in the structure of algorithm for sorting variants of GMDH models. This made it possible to reduce the computational complexity of the procedure for estimating parameters of each model to the second power of the number of estimated arguments compared to using the classical (non-recurrent) Gaussian method, which is known to be proportional to the third power [83]. In [9] (1985) a recurrence relation is given for calculating the regularity criterion in a combinatorial algorithm, which however, is not computationally efficient.

A useful feature of the algorithms of this class is the allocation of a separate stage of the construction of the matrix of normal equations when construction models, which makes it possible to achieve independence of computational complexity of the stage of construction models-candidates from the number of observations. In accordance with [53,54,99], method consists in performing the following actions:

- Step 1. Calculate the extended Z matrix $\dim \mathbf{Z} = n_w \times l_p$ according to the polynomial (1.2).
- Step 2. Calculation of the matrix of normal equations $\mathbf{W}^T \mathbf{W}$, $\mathbf{W} = (\mathbf{Z} : \mathbf{y})$ on samples A and B.
- Step 3. Construction of models-candidates (evaluation of parameters and calculation of the model selection criterion) using only the elements of the matrices $\mathbf{W}_A^T \mathbf{W}_A$, $\mathbf{W}_B^T \mathbf{W}_B$, $\dim \mathbf{W}^T \mathbf{W} = \dim \mathbf{W}_A^T \mathbf{W}_A = \dim \mathbf{W}_B^T \mathbf{W}_B = (l_p + 1) \times (l_p + 1)$.

As we can see, the computational complexity of the first two steps of the method depends on the number of observations n_w , but the most time-consuming third step, where the generation and calculation of models-candidates is directly performed, does not depend on this number.

The advantage of combinatorial algorithms is that the global minimum of the CR criterion is found on a given limited class of model structures Ψ , the disadvantage is the exponential growth of the computational complexity of the algorithm L with an increase in the number of l_p arguments,

$$L = 2^{l_p} - 1 = \sum_{i=1}^{l_p} C_{l_p}^i .$$

The goal of *truncated (directed) sorting algorithms* is to reduce the computational complexity of algorithm (increase its performance), to find a solution that would be close to the result of a complete sorting or would coincide with it. In each truncated sorting algorithm, based on the results of a numerical experiment, an estimate of the probability of finding an exact solution can be given.

One of these algorithms is the MULTI algorithm [9], which uses a truncated (directed) sorting of solutions. Let's describe its work using the example of construction a linear model. At the first stage, a set of models consisting of one argument is built, at the second stage – from two, and so on to a model containing all m arguments. However, at each stage, a given number of $F_s = F$ of the best models is selected (s is the number of the stage), and then one of the missing arguments is added to each best model from the previous stage. Note that the choice at first stage $F_1 = m$ of models is a guarantee that the global minimum will not be lost at first stage already. At other stages, at a constant degree of freedom $F_s = m$ there is no such guarantee. Only if at each stage $F_s = C_m^s$, then result of a complete sorting will not be lost, while the MULTI algorithm loses polynomial complexity and coincides with COMBI.

1.3.2 Iterative GMDH algorithms

A mathematical definition of the iterative algorithm of GMDH is given in [14] (1987). We will adhere to the designations adopted in this work.

Suppose a matrix $\mathbf{G}_0 = (\mathbf{0} : \mathbf{1} : \mathbf{X})$ is given, where $\mathbf{0}$, $\mathbf{1}$ is a vector-sob of zeros and units of dimension n_w ; $\hat{\mathbf{Y}}_r^* = (\hat{\mathbf{y}}_{r,1}^* : \hat{\mathbf{y}}_{r,2}^* : \dots : \hat{\mathbf{y}}_{r,F}^*)$ is a matrix of solutions obtained at the r -th iteration.

The function $g(\mathbf{Z}_r, \Omega)$ specified with parameters precision (hereinafter referred to as the function), where the column-vectors of the matrix $\mathbf{Z}_r$ are some of the column-vectors of the matrix $\mathbf{G}_r = (\mathbf{0} : \mathbf{1} : \mathbf{X} : \hat{\mathbf{Y}}_r^*)$, and Ω - vector of unknown parameters, maps the solutions of the r -th iteration into solutions of the $\hat{\mathbf{y}}_{r+1,j}$ ($r+1$) iteration, $r \geq 1$.

The function $g_0(\mathbf{Z}_0, \Omega)$, specified with parameters precision, where the column-vectors of the matrix $\mathbf{Z}_0$ are some of the column-vectors of the matrix $\mathbf{G}_0 = (\mathbf{0} : \mathbf{1} : \mathbf{X})$, constructs the solutions of the $\hat{\mathbf{y}}_{1,j}$, first iteration. Estimates of parameters $\hat{\Omega}$ of the functions g and g_0 are determined by the LSM. The functions g and g_0 - functions of the *partial* description of algorithm. The iterative algorithm operates on the matrices $\mathbf{G}_0$ and $\mathbf{G}_r$, functions g , g_0 , obtaining solutions for the next iteration until the stopping rule is satisfied.

To determine the iterative algorithm of GMDH, it is necessary to specify:

- 1) the initial condition – the matrix $\mathbf{G}_0$; 2) the functions g and g_0 ; 3) the stop rule.

A stop rule, for example, can be:

- 1) criterion $Cr_r > Cr_{r-1}$ that defines the iteration at which it will begin to increase, or
- 2) a specified number of iterations.

Iterative algorithms of GMDH are divided into multi-layered and relaxation algorithms, which, in turn, are included in combined ones as constituent elements.

Let us consider the convergence of iterative algorithms of GMDH.

1.4 Convergence of iterative GMDH algorithms

The problem of convergence of iterative GMDH algorithms has received very little attention in the literature. The problem of convergence of algorithms is considered in [14], where the author first formulates definitions of the main types of convergence of GMDH algorithms.

According to [14], three types of convergence are distinguished: to a limit point, to a solution, and "by structure".

Let be $\hat{\mathbf{y}}_{i,r}$ - the exit vector of i -th model, $i = \overline{1, F}$, obtained on the iteration r of a certain GMDH algorithm (F - freedom of choice).

If the sequence $\{\hat{\mathbf{y}}_{i,r}\}$ at $r \rightarrow \infty$ has a limit vector $\mathbf{y}^* \neq \mathbf{y}$, then the GMDH algorithm converges to the limit point. The presence of a limit point is the weakest type of convergence.

A stronger convergence is convergence to a solution. An algorithm converges to a solution if there is a limit for the sequence of vectors $\{\hat{\mathbf{y}}_{i,r}\}$, $i = \overline{1, F}$ at $r \rightarrow \infty$ and it is equal to $\mathbf{y}$.

The strongest is the convergence in structure. Let Φ be a set of functions that satisfies the conditions:

- a) Functions $g \in \Phi$ are linear on parameters Θ .
- b) Any function from Φ is one of the following functions: 1) a constant; 2) one of the columns of the matrix $\mathbf{X}$; 3) can be obtained from 1) and 2) substitution in g and g_0 ; 4) the result of a finite number of superpositions of the functions g and g_0 .
- c) The functions of the partial description f and f_0 do not extend beyond the class Φ , i.e. if $f_1, f_2, \dots, f_p \in \Phi$, then $g_0(f_1, f_2, \dots, f_p, \Omega) = f_1(\bullet, \Theta) \in \Phi$, $g(f_1, f_2, \dots, f_p, \Omega) = f_v(\bullet, \Theta) \in \Phi$.

And let among the set of functions Φ there be a single function $\check{f}$, such that for a certain value of the parameters Θ , the equations

$$\check{f}(\check{x}_{i,1}, \check{x}_{i,2}, \dots, \check{x}_{i,\bar{M}}, \check{\Theta}) = y_i, \quad i = 1, \dots, n_w.$$

An algorithm converges "in structure" if the sequence of models $\{\hat{\mathbf{y}}_{i,r}\}$ at $r \rightarrow \infty$ has a limit and it is equal to $\check{f}(\check{\mathbf{X}}, \check{\Theta})$.

All three types of convergence depend on the CR criterion, which determines the structure of the model. Depending on its type, convergence to the limit point, solution, and "structure" can be internal or external.

Internal convergence (to the solution, limit point, "by structure") is the convergence when the CR criterion is the residual sum of the squares RSS_w , and the values of the parameter vector $\hat{\Theta}$ are obtained by LSM.

"If additive noise is present in the output variable $\mathbf{y} = \check{\mathbf{y}} + \xi$, convergence to the solution, using both an external and an internal criterion, is obviously impossible (at the moment, there are no known criteria that provide such convergence). In this case, the algorithm converges to the limit point" [14].

Below is an overview of the iterative algorithms of GMDH in chronological order of their appearance.

1.5 Classical iterative algorithms

In the first (classical) *combinatorial multi-layered algorithm of GMDH*, proposed by A.G. Ivakhnenko (1968) [2], the trend of time series was identified and its residuals were approximated. Later, the *multi-layered algorithm* was called the *multi-layered iterative algorithm* (MIA) of GMDH, that was described in most monographs on GMDH. The multi-layered iterative algorithm (see Fig. 1.9) was primarily developed to solve the problem of approximation the polynomial (1.2) of m variables of degree p_m by a superposition of several polynomials (1.2) of the lower degree from a pair of variables.

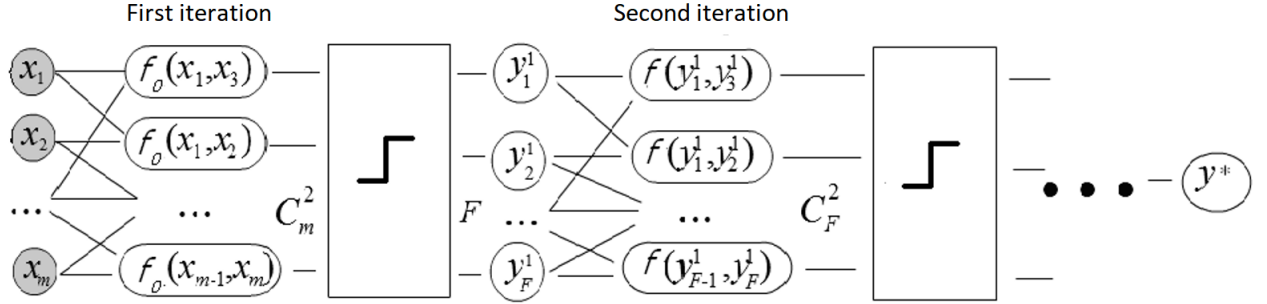


Figure 1.9 – Diagram of the multi-layered iterative GMDH algorithm

⌋ - sign of the threshold selection of the best F models.

In [1], a multi-layered iterative algorithm without calculation of residuals has fixed functions of a partial description of the following form:

- "with covariances" at the length of the polynomial $l = 4$

$$f_0(x_i, x_j, \Theta) = y_k^0 = \theta_0 + \theta_1 x_i + \theta_2 x_j + \theta_3 x_i x_j, r = 1$$

$$f(y_i^{r-1}, y_j^{r-1}, \Theta) = y_k^r = \theta_0 + \theta_1 y_i^{r-1} + \theta_2 y_j^{r-1} + \theta_3 y_i^{r-1} y_j^{r-1}, r \geq 1$$

- "with covariances and quadratic descriptions" at $l = 6$

$$f_0(x_i, x_j, \Theta) = y_k^0 = \theta_0 + \theta_1 x_i + \theta_2 x_j + \theta_3 x_i x_j + \theta_4 x_i^2 + \theta_5 x_j^2, r = 1, \quad (1.13)$$

$$f(y_i^{r-1}, y_j^{r-1}, \Theta) = y_k^r = \theta_0 + \theta_1 y_i^{r-1} + \theta_2 y_j^{r-1} + \theta_3 y_i^{r-1} y_j^{r-1} + \theta_4 (y_i^{r-1})^2 + \theta_5 (y_j^{r-1})^2, r \geq 1. \quad (1.14)$$

MIA is the very first GMDH algorithm that has found wide application in solving practical problems, has the longest history of getting rid of shortcomings, some of which were eliminated in the subsequent works of A.G. Ivakhnenko and his team. Let us describe the shortcomings of the MIA and indicate the works in which they are given attention, which have proposals and the results of their elimination.

1. A sharp increase in the complexity of model from layer to layer (from iteration to iteration). In [14, 56 – 63, 66], the function of the partial description of f_r and f_0 is optimized at each iteration. In [67] a linear partial description function (PDF) over input variables is used.

2. "Inzucht" (degeneration/degradation) of models. In [63], close solutions obtained in the current iteration are excluded; in [64] the principle of selection of the best models is changing; In [60, 65], niche of solutions (models) is used.

3. The impossibility of excluding "false" arguments from the model. In [56], the GMDH relaxation algorithm is proposed, the feature of which is the possibility of both including and excluding any of the arguments that may be present in the model at each iteration. Relaxation iterative algorithms described in [14, 66, 62] are able to exclude "false" arguments from models also. In [67]

there is a possibility of exclusion, since the PDFs in each layers are linear models from the initial variables.

4. *Possibility of losing "true" arguments on the first iteration.* In [56, 14, 60, 66, 68, 62], the original arguments are added to the model on each layer (iteration).

5. *Poor conditioning of the matrix in problem of estimating coefficients by least squares method* for functions of partial description f and f_0 . The problem is partially solved in [63, 60, 65] when solving the problem of models "inzucht". In the algorithm [69], this problem does not arise with uncorrelated (weakly correlated) input variables.

6. *Impossibility of convergence of the algorithm to the true model.* For the relaxation iterative algorithm CML (convergent multi-layered) in [14] (and hence for special cases of construction linear models of algorithms [56, 66]), it has been theoretically proven under what conditions its internal convergence occurs. In [62], numerically, several test examples show the convergence of some particular algorithms of this shell of algorithms to the true model. In [67] was proven the monotonicity of the decrease in the error of the internal criterion from layers to layers for MIA and the existence of a multi-layer's error due to the loss of the basis of the original variables, which means the impossibility of convergence to the true model.

7. *The computational complexity of the stage of constructing models-candidate depends on the number of observations n_w .* In [60], a heuristic procedure is used to reduce the number of observations, which however, does not make the algorithm independent of n_w at the model construction stage.

The results of the analysis of the listed problems solutions in each of the works are presented in Table 1.2. A line "advantage" has been added at the bottom – the ability to build nonlinear models.

Table 1.2 – Results of the analysis of works in which there are proposals and results for eliminating the shortcomings of the MIA: "+" means that the work proposes an algorithm that does not have a disadvantage under the specified number, as well as has the advantage of nonlinear models' construction; "?+" means that the disadvantage is eliminated under certain conditions.

Drawback	References to works										
	[60]	[14]	[57-59]	[56,66]	[61]	[62]	[63]	[64]	[65]	[68]	[67]
1	+	+	+	+	+	+	+				+
2	+						+	+	+		
3		+		+		+					+
4	+	+		+		+				+	
5	+	?+		?+			+		+		+
6		+		+		+					
7											
Advantage	+	+	+	+	+	+		+	+	+	

The "inzucht" of models means the generation of models whose structures are similar, i.e. have a similar set of monomials of polynomial (1.2) (set of arguments). Proximity is also estimated by the number of coinciding monomials in the structures of models. The reason for this drawback is the rule of algorithm selection, namely: the PDF structures of the current layer are formed only from the outputs of the best models of the previous layer. In foreign literature, is called as lack of *diversity problem*.

Let us dwell on the main ideas embedded in iterative algorithms in more detail and describe them in chronological order.

$$f_0(q(\mathbf{G}_k, \omega) = \omega_0 q(\mathbf{G}_k), \quad (1.15)$$

$$f(\hat{\mathbf{y}}^r, \mathbf{G}_k, \Omega) = \omega_1 \hat{\mathbf{y}}^r + \omega_2 q(\mathbf{G}_k), \quad (1.16)$$

where the matrix $\mathbf{G}_k = \{\mathbf{x}_i\}$, $\mathbf{x}_i \in \mathbf{X}$ $i = \overline{1, k}$, $k \leq m$, vector $\hat{\mathbf{y}}^r \in \hat{\mathbf{Y}}_r^*$, matrix $\hat{\mathbf{Y}}_r^* = (\hat{\mathbf{y}}_{r,1}^* : \hat{\mathbf{y}}_{r,2}^* : \dots : \hat{\mathbf{y}}_{r,F}^*)$, freedom of choice at each iteration $F = 1$, r is the number of the iteration where the summation occurs (1.16). In this algorithm, in addition to the ability to add variables from the original set of variables, i.e. not to lose the "true" ones, and to exclude "false" arguments, it is possible to construct nonlinear models based on input variables.

$$z_j = \prod_{i=1}^k x_{ij}, \quad j=1, n_W, \quad f_0(\mathbf{z}_i, \Omega), \quad r=1, \quad f(\mathbf{z}_i, \hat{\mathbf{y}}_r, \Omega), \quad r>1. \quad (1.17)$$

* - multiplication sign; $\textcircled{x_1}$ – calculated monomial; $\textcircled{x_2}$ – uncalculated monomial; $\textcircled{x_3}$ – the best monomial on the iteration of multiplications; $\textcircled{x_4}$ is a monomial that must be introduced into the model on the $(r+1)$ th iteration.

The main advantages of the MSA algorithm are that it does not lose the original basis of variables in truncated sorting and on each row it has a description of the model output, represented in

the original input variables. Given these advantages, it was chosen as the basic algorithm for further improvement. Its detailed description will be given in section 3.1.

In [57] (1978) it was proposed to use a *combinatorial sorting* of the function of a partial description in MIA "with covariances and quadratic descriptions":

$$f_i(v_1, v_2) = \omega_0 + \omega_1 v_1 + \omega_2 v_2 + \omega_3 v_1 v_2 + \omega_4 v_1^2 + \omega_5 v_2^2. \quad (1.18)$$

The difference between the *algorithm of directional sorting* of "covariances and quadratic descriptions" described in [59] (1994) and the algorithm of sorting of "covariances and quadratic descriptions" [57] is in the degree of freedom when sorting the function of a partial description. In [59] a truncated sorting is used. A set of structures equal to five is obtained by the method of nested structures.

In [69] (1981), the "intrinsic" convergence of an algorithm with a linear partial description, called a Convergent multi-layered (CML) algorithm, was proved.

The CML algorithm constructs a partial description of the species that is linear in terms of variables and parameters

$$f(v, u, \Omega) = \omega_1 u + \omega_2 v, \quad v \in \mathfrak{R}, \quad u \in \mathfrak{R}, \quad (1.19)$$

where v is the value of the element in a certain row of the matrix, $v \in \mathbf{x}_i$, $\mathbf{x}_i$ is the column-vector of the matrix $\mathbf{X}$; u is the value of the element in the same line of the vector, $u \in \hat{\mathbf{y}}_{i,r}$, $i = \overline{1, F}$, r is the iteration number.

Obviously, the generator of the structures of this algorithm is a special case of the MSA generator of the GMDH. In the CML algorithm, unlike the MSA, the freedom of choice $F > 1$ is implemented in all iterations, but this does not affect the fact of the algorithm's convergence, but only its speed.

Paper [58] contains the first statement of a problem solved by a multi-layered algorithm. As a result of the analysis of the process of obtaining solutions, it is noted that the formulation of a problem for a combinatorial algorithm is a special case of formulation for multi-layered algorithms.

In [58] (1981), a polynomial *algorithm GN (with Gödelian numbering)* was proposed, in which the functions of partial descriptions generally depend on three arguments:

$$f(v_1, v_2, v_3, \Omega) = \omega_1 v_1 + \omega_2 v_2 v_3, \quad v_i \in \mathfrak{R}, \quad (1.20)$$

where v_i is the value of the element in some vector row $\mathbf{v}$ of the matrices $\mathbf{G}$ and $\mathbf{G}_0$; $v_i \in \{x, \hat{y}, 0, 1\}$, $i = \overline{1, 3}$ where $x \in \mathbf{x}_j$ is the column-vector of the matrix $\mathbf{X}$, $j = \overline{1, m}$; $\hat{y} \in \hat{\mathbf{y}}_{r,k}$ is a column-vector of the matrix $\hat{\mathbf{Y}}_r$; $k = \overline{1, F}$, $r = 1, \dots, R$; $\mathbf{0}$, $\mathbf{1}$ are vectors of zeros and ones of the same dimension as vectors $\mathbf{x}_j$, $\hat{\mathbf{y}}_{r,k}$, $0 \in \mathbf{0}$, $1 \in \mathbf{1}$.

In this algorithm, the stop is carried out according to the stabilization model criterion.

It can be shown that in the GN algorithm, the sorting in the nodes of the network structure of the algorithm is limited to the 4-th degree of the polynomial, both in terms of the number of arguments and the number of degrees of freedom.

The advantages of the GN algorithm are:

1. Construction of nonlinear polynomial models;
2. Implementing complexity control at each iteration;
3. Simple representation of the model in an expanded form when returning to the original variables;
4. The presence of the initial basis (variables of the matrix $\mathbf{X}$) at each iteration.

"Theoretical proof of convergence to the limiting point of this algorithm cannot be found due to the search for a global extremum in an infinite-dimensional space of variables" [14]. It can be shown that the type of functions iterated over at each iteration in the partial descriptions of the GN

algorithm, as a special case, has the form of the PDF of the MIA algorithm with combinatorial sorting, and therefore no evidence of convergence for the GN algorithm can be found.

A modification of this algorithm developed in the applications is called as *the polynomial neuron network* (GMDH-PNN) [80].

1.6 Modern iterative algorithms based on GMDH

Speaking of modern iterative algorithms, we mean works with their descriptions published after 2000.

The GAME algorithm was proposed to eliminate the following shortcomings of the classical multi-layered algorithm of GMDH: 1) overcomplication of the model due to the exponential growth of the degree of the polynomial; 2) the possibility of losing the original true arguments on the first iteration; 3) "inzucht" or degeneration of models.

On the basis of it, *the technology of fully automated knowledge extraction using group of adaptive model evolution* (FAKE GAME) was developed [60] (2006). It is assumed that the first two drawbacks are eliminated by complicating the structures of models with the addition of initial variables in subsequent iterations. The algorithm has a greater variety of PDFs, a pyramidal structure of a multi-connected network (there are connections not only between successive layers, but also through the layers); it uses ensemble outputs of several better models to obtain greater accuracy and visualization of solutions to facilitate the interpretation of the obtained functional dependencies.

As partial descriptions of GAME, unlike MIA, not only polynomial functions are used, but can be: gaussians, logistic functions and other functions inherent to the nodes of a neural network, so this algorithm belongs to GMDH-like neural networks. There can be more than two variables at the input of nodes, and the algorithm implements the sorting similar to genetic algorithms. Nodes are active neurons in which there is a complete sorting and genetic selection of the arguments composition.

The problem of "inzucht" is proposed to be solved with the help of niches. Niche involves dividing the constructed models-candidates into sets (niches) according to the distance between them. For calculation of the distance between models, are taken into account both the paired correlation of their error vectors in the training sample and the distances between unit vectors, that encoding the presence of inputs. When selecting the F models to move on to the next layer, one model from each niche is selected in sequence.

In [66] (2006) a modification of the MSA is proposed – *a multi-layered algorithm with combinatorics and selection of generalized variables* (MACSO). The network structure of the algorithm differs from the MSA by greater freedom of choice at each iteration. The partial description functions of MACSO and MSA are identical. Compared to MSA, it has an expanded basis of partial description functions (added $\sqrt[3]{x_i}$, $1/\sqrt[3]{x_i}$), two additional "external" criteria for constructing smooth functions between the object's observation points, variable degrees of freedom at the summation iteration $F > 1$ and multiplication $F_1 > 1$, and control of complexity at each multiplication iteration. MACSO has the ability to construct a system of equations with reassignment of output and input variables and a user-friendly interface.

Semi-random model selection algorithm [68] (2007). According to the description, it generates model structures obtained by the inclusion (embedding) method. The number of terms increases by one from iteration to iteration. No monomial is removed during this process. In the selection procedure, the probability of a given model being connected to the next monomial is inversely proportional to the value of the regularity criterion. Thus, at each iteration, it is possible to construct a new model using the initial arguments. This algorithm can be classified as an iterative algorithm

with unlimited freedom of choice, constructing linear partial descriptions, with a Markov chain-type connection structure at the iterations.

In [74] (2007) the *GMDH feedback algorithm* self-selecting optimum of a neural network is proposed.

The network structure of one layer of the algorithm is an ANN with feedback [76]. Structural parameters, such as the number of layers, the number of neurons in the hidden layers, and the number of input variables, are selected automatically. This is conceptually related to GMDH. However, the comparison of models and selection are performed using data from the same sample. The selection criterion is the prediction sum square (PSS) – the Akaike information criterion $AIC = 2s - 2\ln(RSS)$, which is considered as internal criterion in GMDH theory. That is, from the three main principles of GMDH, this algorithm includes two: the complication of models from series to series and incomplete solutions.

There are three types of neural network elements in network nodes. As a neural network element are used the PDF sigmoid function, the radial-basis function (Gaussian) and the polynomial type PDF.

This algorithm was applied in medicine to recognize a three-dimensional image of the lung, pulmonary vessels and bronchial trees.

In [61] (2008), *hybridization* of the classical MIA GMDH with the *Particle Swarm Optimization* (PSO) algorithm is proposed. Hybridization consists in the use of PSO for PDF optimization. At the same time, the optimal composition, the number of input variables in PDF, and their maximum power are determined. This algorithm belongs to the same class as ant algorithms, namely, the class of algorithms, the ideas for the creation of which are borrowed from nature. Such algorithms do not guarantee finding a global optimum. The idea of PSO is based on observing the behaviour of a swarm of bees or flocks of fish and birds.

The GMDH-like algorithm based on an artificial neural network described below contains all the principles of GMDH.

In [62] (2011), in order to create an internet technology for inductive modeling, the *generalized iterative algorithm* (GIA) of GMDH was developed. In fact, it is a shell of well-known classical algorithms described above in subsection 1.5, such as MIA [1], MIA with combinatorial sorting of quadratic partial descriptions and covariances [57] and GN algorithm [58].

In the GIA in addition to the function of the first iteration of the form [1]

$$f_0(x_i, x_j, \Theta) = \theta_0 + \theta_1 x_i + \theta_2 x_j + \theta_3 x_i x_j + \theta_4 x_i^2 + \theta_5 x_j^2, \quad (1.21)$$

At subsequent iterations, a partial description, depending on the set of algorithm parameters, can be:

- as a function of the form (1.14) [1], we write it in the form:

$$f(\mathbf{v}, \Omega) = \omega_0 + \omega_1 v_1 + \omega_2 v_2 + \omega_3 v_1 v_2 + \omega_4 v_1^2 + \omega_5 v_2^2, \quad (1.22)$$

where $v_i \in \mathbf{v}$ is the row-vector of the i -th submatrix $\hat{\mathbf{Y}}_{r,i}$ of the matrix $\hat{\mathbf{Y}}_r$ - of r -th iteration: $\dim \hat{\mathbf{Y}}_{r,i} = n_W \times 2$, $i = \overline{1, I}$, $I = C_F^2$;

- as a function for which (it can be shown) also performs a combinatorial sorting of the PDF in the GN algorithm [58]:

$$f(v, u, \Omega) = \omega_0 + \omega_1 v + \omega_2 u + \omega_3 v u + \omega_4 v^2 + \omega_5 u^2, \quad (1.23)$$

where the element $v \in \mathbf{x}_j$, $j = \overline{1, m}$, $\mathbf{x}_j$ is the column-vector of the matrix $\mathbf{X}$, $\mathbf{x}_j \in \mathbf{X}$, and the variable $u \in \hat{\mathbf{y}}_{i,r}$, $\hat{\mathbf{y}}_{i,r} \in \hat{\mathbf{Y}}_r$, $i = \overline{1, F}$. Obviously, this algorithm belongs to *combined iterative algorithms*, as well as the algorithm with Gödelian numbering. The difference is that in the PDF (1.22), (1.23) a complete sorting of the model structures is performed, including the maximum total degree of monomials equal to four, while in the GN algorithm the sorting is limited to the maximum total degree of monomials equal to three.

GIA has the ability to separate individual algorithms from the general network structure (MIA, RIA, MIA "with combinatorial sorting of covariances and quadratic descriptions", RIA "with combinatorial sorting of covariances and quadratic descriptions", etc.).

GIA has a fairly extensive network and a fairly large number of calculations if you use all its combined capabilities. Its main drawback: it has a low computational speed.

The presented analysis allowed us to select a relaxation-iterative algorithm for development. The advantages of sorting-out algorithms were used to improve it.

The Generalized Relaxation Iterative Algorithm (GRIA) (2012) is the fastest algorithm among all the GMDH algorithms and GMDH-like neural networks. It is capable of solving modeling problems with a large dimension of the initial data matrix. We will not describe its structure here, since it and its prototype will be described in detail in Chapter 3. We will only say that after creation of this algorithm, a comparative characteristic of the main types of algorithms has changed and is shown in Table 2.2. Using the example of GRIA, Chapter 4 demonstrates a methodology for numerically studying algorithms and obtaining new knowledge through computational proofs. In a numerical experiment, analytical estimates of the convergence rate were obtained. For its variation, with full sorting RIA CS convergence was theoretically proven. Chapter 5 shows examples of solving forecasting and classification problems, in particular, the problem of one-step-ahead "space weather" forecasting, the input matrix of which contains hundreds of thousands of observations and a thousand variables.

Table 2.2 – Results of the analysis of the main types of GMDH algorithms taking into account the GRIA.

GMDH Algorithms	Sorting-out		Iterative	
	Full sorting	Truncated sorting	Multi-layered	Relaxation GRIA
Max no of observations, n	upto 500 000		upto 10000	to 500 000
Max no of variables, m	upto 30	upto 300	upto 1000	to 10000
Complexity of computation of model on one iteration (layer)	Quadratic	Quadratic	Quadratic	Linear
Independence of the model calculation algorithm from number of observations	Yes		No	Yes
Use of recurrent computations	Yes		No	Yes

1.7 Discussion

The first chapter defines the fundamental concepts of mathematical modeling and describes computer heuristic modeling tasks such as identification, extrapolation, forecasting, and computational proofs. We highlighted the significant differences between identification, extrapolation, and forecasting tasks when modeling using tables of numerical data. The differences between extrapolation and forecasting tasks, as well as interpolation and approximation, are controversial. We defined extrapolation and various types of forecasting and illustrated the differences between the above-mentioned tasks with examples. The formulation of the structural-parametric identification problem includes the extrapolation task as an integral part of choosing the best model structure. This model approximates the data, since data interpolation by the model yields

large errors outside the tabular values on which it is based. Furthermore, an interpolation model has a greater error than an approximation model in the forecasting domain and within the data sampling interval, since it should be borne in mind that the observed processes and natural objects are mostly continuous. The difference equations we construct for forecasting are essentially difference schemes [96] used to describe a continuous process. We have considered alternative approaches to solving the forecasting problem. Attention is paid to methods of mathematical modeling, their features, and areas of application. Deductive and inductive methods of modeling based on experimental data, artificial neural networks, are briefly described, and methods using regression models are singled out for a more detailed presentation. This brief overview is necessary to explain to the reader the specific "niche" of the inductive method of self-organization of models in relation to all others in the edifice of mathematical methods, and what its features are. We have discussed in detail the method of group accounting of arguments, which uses regression models and an implicit penalty for their complexity. We have emphasized the noise-immunity of GMDH models and considered definitions important in the theory of inductive modeling, such as a "true" and "adequate" model. These two types of models arose from a debate over what exactly constitutes a "physical" model of an object. We focused on the concept of criterion adequacy and the cases in which a "true" model is needed, and in which cases an "adequate" model is needed. Two NP-complete problems of the inductive method of model self-organization were distinguished: 1) the problem of finding the structure and parameters of a model corresponding to the global extremum of the performance functional, and 2) the problem of finding the optimal sample partitioning. Types of partitioning the observation set are briefly discussed.

The review and analysis of the existing algorithms of GMDH made it possible to identify a group of relaxation iterative algorithms that have important advantages in modeling of complex objects. The creation of the GRIA significantly expanded the capabilities of the iterative algorithms of GMDH, made it possible to solve modeling problems based on large data sets. The modern generalized relaxation-based iterative algorithm for model self-organization, based on recursive parameter estimation and calculation of an external criterion during the iterative search and selection of the best models, is independent of the number of observations and has a computational complexity linear in the number of arguments.

The presented review of GMDH algorithms made it possible to outline the prospects for their further improvement.

Chapter 2

The Problem of Sample Partition in Heuristic Modeling

Let's take a closer look at the problem of partition the sample, briefly described in subsection 1.2.5.

The effectiveness of applying 'cross-validation' type criteria, which underlie GMDH algorithms, to solving practical modeling problems depends, in particular, on the method of partitioning the data sample.

2.1 Optimal Sample Partition as a Result of Experiment Design

In [50], a partition of a set of observations was proposed, the so-called quasi-optimal partition of the initial matrix $\mathbf{X}$, in which the matrix norm of the deviations of its submatrices is minimized.

It is known that multi-layered algorithms (which are used in modeling complex objects) do not guarantee a solution that provides a global minimum of the accuracy criterion on the entire set of possible structures. The rationale for the use of ρ -proportional partitioning is that it helps to obtain a model structure and parameters that provide a minimum variance of the error of the ideal criterion on all samples for which it is used (training, testing, examination), $s_A^0 = s_B^0$. An ideal criterion is a criterion, whose minimization leads to the search for the true model of the object, provided that it is present in the class of structures to be sorted out:

$$J_G(s) = E \left\| \mathbf{y}_{Gs}^0 - \mathbf{X}_{Gs} \hat{\theta}_{As} \right\|^2, \quad G = A, B,$$
$$s_G^0 = \arg \min_{s=1, m} J_G(s),$$

where $\mathbf{y}_{Gs}^0$ is the exact output of the object (true signal). Since the ideal criterion is incomputable on real data $\mathbf{W} = (\mathbf{X}; \mathbf{y})$ due to their "noisiness", in practice its "physically feasible" estimate is used, which in self-organization theory is the criterion of regularity:

$$AR(s) = \left\| \mathbf{y}_{Bs} - \mathbf{X}_{Bs} \hat{\theta}_{As} \right\|^2,$$

where A and B are two non-overlapping disjoint subsamples. Here $\hat{\theta}_{As}$ is the vector of parameters of the model with structure s , estimated by LSM on the training sample A , $\dim \mathbf{X}_{Bs} = n_B \times m$, $\dim \mathbf{y}_{Bs} = n_B \times 1$.

Is it possible to construct such a partition? A rigorous solution to this problem involves experiment planning. The results obtained in [50] can be summarized as the following statement.

If the partitioning of sample W into two subsamples A and B is such that the design matrices $\mathbf{X}_A$ and $\mathbf{X}_B$ with $\text{rank } \mathbf{X}_A = \text{rank } \mathbf{X}_B = m$ satisfy the *quadratic dependence condition*

$$\mathbf{X}_A^T \mathbf{X}_A = \rho_B^2 \mathbf{X}_B^T \mathbf{X}_B, \quad \rho_B^2 \neq 0 \quad (2.5)$$

then the minimum mathematical expectation of the regularity criterion $\overline{AR}(s, \rho_B^2)$ corresponds to the optimal complexity $s_{AR}^* = s_A^0 = s_B^0$.

Specific cases of such partitioning: a) orthogonal regressors; b) linearly dependent (ρ -proportional) partition ($\rho_B \mathbf{X}_B = \mathbf{X}_A$, $\rho_B \neq 0$) (see Fig. 2.3), including the case of the repeated experiment $\mathbf{X}_A = \mathbf{X}_B$, ($\rho=1$).

For models with a free term $\rho_B^2 = n_A/n_B$, since the first diagonal elements of the matrices $\chi_A \triangleq \mathbf{X}_A^T \mathbf{X}_A$, $\chi_B \triangleq \mathbf{X}_B^T \mathbf{X}_B$ are equal to n_A , n_B and $(n_A + n_B = n_W)$ respectively.

Relation (2.5) characterizes the optimal experimental design and the corresponding quadratic (ρ^2 -proportional) partitioning of the sample. The problem of constructing such a design itself is not considered here, since the case of passive sampling is being investigated. We emphasize that condition (2.5) applies to all regressors simultaneously, i.e. it is valid for any increase in the complexity of the model structure. It is also important that in this formulation, the sample partitioning problem is formulated and solved before the start of the modeling process, as with other partitioning methods.

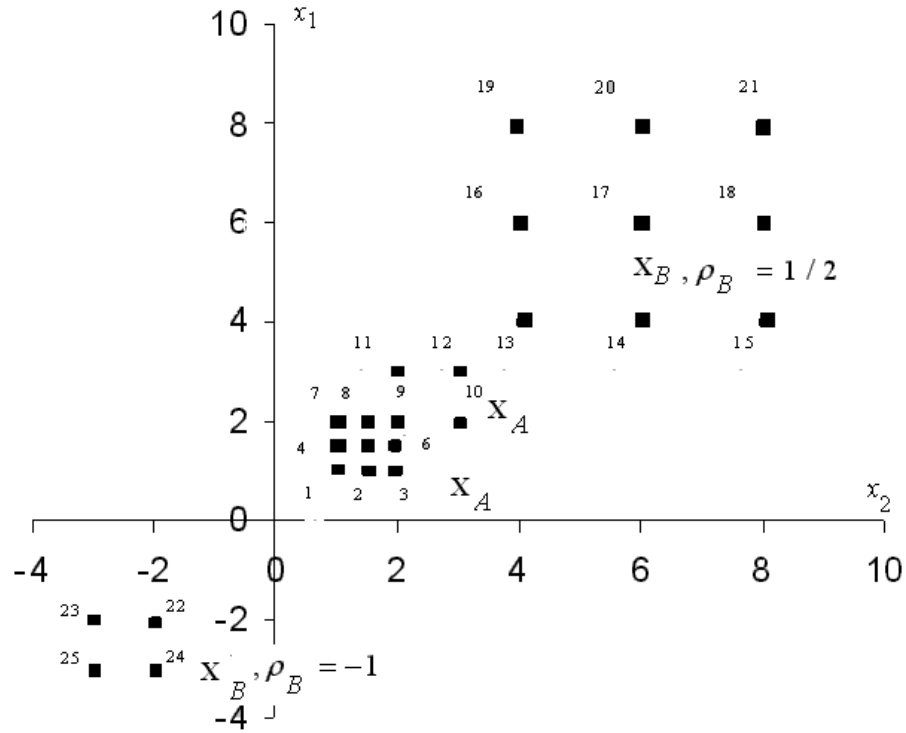


Figure 2.1 – Examples of linearly dependent partitions of design points $\mathbf{X}_A$ and $\mathbf{X}_B$, $m = 2$

Thus, ρ^2 -proportional partitioning facilitates the search for the true structure of object model under conditions of insignificant noise, or under conditions of noise level uncertainty – a structure adequate to the noise level.

2.2 Relationship between optimal partitioning and partitioning by 'variance'

Since ρ^2 -proportional partitioning procedure is performed before the structural identification algorithm, just like the traditionally used GMDH method of partitioning observation points by the total squared deviation from the mean ("variance"), it is of interest to establish the relationship between these two partitioning methods.

Partitioning by "variance". First, the sample mean $\bar{x}_i = \frac{1}{n_W} \sum_{j=1}^{n_W} x_{ji}$, $i = \overline{1, m}$ of each

variable in the design matrix $\mathbf{X}$ is determined, relative to which the sum of the squared deviations

from the mean ("variance") at each point is calculated:

$$\eta_j^2 = \sum_{i=1}^m (x_{ji} - \bar{x}_i)^2, j = 1, n_W,$$

and all obtained values η_j^2 are ranked in descending order. Division η_j^2 by $(m-1)$ is meaningless due to the application of the ">, <" relations to them and is therefore not performed. This method differs from conventional variance in that the squared deviations from the mean values for different variables are summed – this demonstrates that this method abstracts from the semantic content of the variables and considers only their statistical properties. The ranked points are then assigned to subsamples A and B in two ways:

1) through one (two, three...) point, which means obtaining different degrees of similarity between the subsamples;

2) either sequentially: the first $2/3 n_W$ of the points go into the training subsample, the remaining $1/3 n_W$ of the points - into the testing subsample, which results in dissimilar subsamples.

Partitioning by "variance" in terms of ρ^2 -proportional dependence of subsamples can be interpreted as follows. Let the sample be such that all $\bar{x}_i = 0$, then $\tilde{\eta}_j^2 = \eta_j^2|_{\bar{x}=0} = \sum_{i=1}^m (x_{ji})^2$.

Further, it is easy to obtain that

$$\text{tr}(\mathbf{X}^T \mathbf{X}) = \sum_{i=1}^m x_{ii} = \sum_{i=1}^m \sum_{j=1}^{n_W} (x_{ji})^2 = \sum_{j=1}^{n_W} \tilde{\eta}_j^2.$$

Therefore, partition by "variance" through a single point is an attempt to obtain subsamples related by the relation $\text{tr}(\mathbf{X}_A^T \mathbf{X}_A) \approx \text{tr}(\mathbf{X}_B^T \mathbf{X}_B)$, and at same time $n_A \approx n_B$, or $\rho_B^2 = n_B / n_A \approx 1$. In this case, there is reason to expect the best conditions for achieving the minimum criteria using diagonal elements. When splitting ranked points sequentially, the goal is achieved $|\text{tr}(\mathbf{X}_A^T \mathbf{X}_A) - \text{tr}(\mathbf{X}_B^T \mathbf{X}_B)| \rightarrow \max$, which is appropriate when using the unbiased criterion (from GMDH).

It should be noted, however, that the condition $\text{tr}(\mathbf{X}_A^T \mathbf{X}_A) = \text{tr}(\mathbf{X}_B^T \mathbf{X}_B)$ is necessary, but not sufficient to satisfy the condition of quadratic dependence (2.5). Therefore, this method should be considered as one of the variants of partitioning, along with more complex ones.

2.3 Quasi-optimal partitioning in the case of passive sampling

The problem of finding such a partition can be formulated as follows: out of all possible partition of the sample W into subsamples $\mathbf{X}_{A\ell}$ and $\mathbf{X}_{B\ell}$, such that $\text{rank } \mathbf{X}_{A\ell} = \text{rank } \mathbf{X}_{B\ell} = m$ the best variant is chosen according to the condition

$$\min_{\ell=1, L} \min_{\rho_B^2 \neq 0} \left\| \mathbf{X}_{A\ell}^T \mathbf{X}_{A\ell} - \rho_B^2 \mathbf{X}_{B\ell}^T \mathbf{X}_{B\ell} \right\|, \quad (2.6)$$

where L is the number of admissible variants, and is $\|\cdot\|$ a matrix norm.

2.3.1 Criteria for finding a quasi-optimal partition

In solution problem (2.6), three main problems arise: what norm to select for the partition evaluation criterion; how to generate the set of partitioning options; how to look for the best partition on this set.

When using matrix norms

$$E1 = \sum_{j=1}^m \sum_{i=j}^m | \chi_{A_{ij}} - \chi_{B_{ij}} |$$

$$E2 = \max_{i=1, m, j=i, m} | \chi_{A_{ij}} - \chi_{B_{ij}} |$$

their minimization to zero or near-zero values leads to the conditions of a repeated experiment (since the $\rho = 1$), which is a special case of the optimal (ρ^2 -proportional) partitioning.

The following expressions (2.7)-(2.9), (2.11), (2.12) are ρ -independent only when their values are equal to or close to zero.

The norm of averaged absolute values of numerators of all possible differences of the elements of a symmetric matrix (2.10):

$$E3 = \frac{1}{C_R^2} \sum_{i,j,k,l \in [1,m]} | \chi_{A_{ij}} \chi_{B_{kr}} - \chi_{A_{kr}} \chi_{B_{ij}} |, \quad (2.7)$$

where $R = m(m+1)/2$, C_R^2 is the number of combinations.

Maximum absolute value of the numerator of the differences of the matrix elements (2.10):

$$E4 = \max_{i,j,k,r \in [1,m]} | \chi_{A_{ij}} \chi_{B_{kr}} - \chi_{A_{kr}} \chi_{B_{ij}} |; \quad (2.8)$$

Normalized difference between the values of the maximum and minimum elements of the matrix (2.10) [50]:

$$E5 = (\rho_{\max} - \rho_{\min}) / (\rho_{\max} + \rho_{\min}), \quad (2.9)$$

$$\rho_{\max} = \max_{i,j} \rho_{ij}, \quad \rho_{\min} = \min_{i,j} \rho_{i,j}, \quad i = \overline{1, m}, \quad j = \overline{i, m}.$$

where $\rho_{ij} = \chi_{A_{ij}} / \chi_{B_{ij}}$ are the elements of the matrix $\wp$

$$\wp = \begin{pmatrix} \frac{\chi_{A_{11}}}{\chi_{B_{11}}} & \dots & \frac{\chi_{A_{1m}}}{\chi_{B_{1m}}} \\ \vdots & \ddots & \vdots \\ \frac{\chi_{A_{m1}}}{\chi_{B_{m1}}} & \dots & \frac{\chi_{A_{mm}}}{\chi_{B_{mm}}} \end{pmatrix} \quad (2.10)$$

Norms (2.11) and (2.12), which use only diagonal elements, are analogues of norms $E3$ and $E5$.

The average modulus norm obtained from all possible differences of diagonal elements of the symmetric matrix (2.10) has the form

$$E6 = \frac{1}{C_m^2} \sum_i \sum_j \left| \chi_{A_{ii}} \chi_{B_{jj}} - \chi_{A_{jj}} \chi_{B_{ii}} \right|. \quad (2.11)$$

Normalized difference between maximum and minimum values of diagonal elements of the matrix (2.10)

$$E7 = (\rho_{\max_i} - \rho_{\min_i}) / (\rho_{\max_i} + \rho_{\min_i}), \quad (2.12)$$

where:

$$\rho_{\max_i} = \max_{i \in \{1, m\}} \rho_{ii}, \quad \rho_{\min_i} = \min_{i \in \{1, m\}} \rho_{ii}.$$

Obviously, for any of the above norms of information matrices mismatch, the partition with a number

$$\ell^* = \underset{\ell=1, L}{\operatorname{argmin}} EJ_\ell, \quad J = \overline{3, 7} \quad (2.13)$$

should be selected as the best (in the sense of ρ^2 -proportionality) partition from the L variants under consideration.

If the obtained criterion value is sufficiently close to zero, then corresponding partition is accepted to be ρ^2 -proportional, and the relation $\rho_B^2 \approx \rho_{\min} \approx \rho_{\max}$ is automatically satisfied.

Note that the equality of criteria based on norms (2.11) and (2.12) to zero does not imply ρ^2 -proportionality for matrices χ_{A_ℓ} and χ_{B_ℓ} , $\ell = \overline{1, L}$.

The following features were revealed in the study of the ρ -independent norms E3-E7. The least computationally expensive, as can be seen from Table 2.1, is norm (2.12), since it is calculated only for the diagonal elements of the information matrices. Then, in ascending order - (2.9), (2.11), (2.7), and (2.8). A comparison of norms for the number of multiplication and division operations reveals that expression (2.9) is the most preferable for finding the optimal partition. However, it should be noted that the scope of this norm is limited to information matrices χ_B with nonzero elements (otherwise, we encounter the division-by-zero problem). Furthermore, this norm is used under the assumption that all elements of the matrix (2.9) are positive, which is typically not the case in real-world applications.

Table 2.1. – Number of computational operations Q

Q	m				Norm
	2	3	4	5	
$2 \cdot C_R^2$	6	30	90	210	(2.7),(2.8) – E3, E4
$R+1$	4	7	11	16	(2.9) – E5
$2 \cdot C_m^2$	2	6	12	20	(2.11) – E6
$m+1$	3	4	5	6	(2.12) – E7

Criterion (2.7) is free from shortcomings (2.9), but for large matrix sizes it becomes very cumbersome.

2.4 Discussion

When modeling objects and processes using GMDH, it is necessary to take into account the correspondence between the data partitioning criterion and the external complement criterion when selecting models.

Since the main problem with the increase in number of points is exponential growth of the computational complexity of the algorithm, a reasonable compromise is needed between *increasing time of searching for a partition and a decrease in error (increase in accuracy) of the models*, which is observed in connection with the use of quasi-optimal partitioning.

The criterion of quasi-optimality of partition can be used to evaluate affine transformations of geometric figures.

Some transformations of various objects in n -dimensional space can be determined using the considered special criteria for quasi-optimal partitioning of observation points, which is demonstrated for the special case of an affine transformation, when an isometric transformation of the coordinates of figures (homothety) takes place.

Test examples of approximating and extrapolating static objects in the presence of noise in data demonstrate the advantage of quasi-optimal partitioning over other partitioning methods. Such definitive conclusions cannot be drawn when studying various types of partitioning data for real processes, where the true model is unknown. However, in the problem of extrapolating the trend of budget revenue changes in Ukraine, the model outputs obtained using several types of partitioning methods, representing the upper and lower estimates of the real process, reflect the trend of budget revenue changes and are used to increase the accuracy of extrapolation.

Chapter 3

Relaxation Iterative Algorithms of GMDH

Performance results are just as important when looking for a model as they are when looking for a better partition. In both problems, in order to ensure that the best solution is found, if the size of the samples and the complexity of the problems allow, as a rule, a complete sorting of options is carried out in a limited area of definition. When solving NP-complete problems that are computationally difficult, all sorts of heuristics are welcome. And since the problem is the same in them, we need to look for a common solution. The analysis performed in Chapter 1 made it possible to identify a group of relaxation iterative algorithms for complex problems solution.

As mentioned in subsection 1.2.4, to estimate model parameters, in addition to methods for accurately estimating model parameters, iterative methods for estimating parameters and algorithms that implement them have been developed. This chapter examines in more detail two iterative relaxation-type GMDH algorithms: a generalized relaxation iterative algorithm (GRIA) and its prototype - a multi-layered simplified algorithm (MSA). The first of them deserves special attention, since it is the fastest modern GMDH algorithm, and the second, because it is its basis, many of whose ideas are embodied into GRIA.

The goal of developing GRIA was to create an GMDH algorithm for modeling complex objects described by high-dimensional measurement matrices in an acceptable computation time (from minutes to one hour), with acceptable model accuracy.

3.1 Multilayered simplified GMDH algorithm

This subsection will describe: organizing the process of models construction; a method for estimating parameters and calculating the model selection criterion; the procedure for finding the best multiplicative monomial. In conclusion, its advantages and disadvantages will be listed.

3.1.1 Description of a multi-layered simplified algorithm

Let the set of observations W be divided into two – the training A and the validation B : $W = A \cup B$, $A \cap B = \emptyset$, $|A| = n_A$, $|B| = n_B$, $|W| = n_W$; I - is the set of indices of the vector-columns of the matrix $\mathbf{H} = (\mathbf{X} : \mathbf{Z})$. The matrix $\mathbf{Z}$ is obtained using functional transformations during formation of generalized variables.

The process of model construction is presented in Fig. 3.1. For simplicity of description of the method, without loss of generality, let no transformations of the elements of columns of the matrix $\mathbf{X}$ have been applied in the block for generating generalized variables of the MSA, then $\mathbf{H} = \mathbf{X}$. Let $\mathbf{x}_i$ be the column-vector of the matrix $\mathbf{X}$, $i = \overline{1, m}$; $\mathbf{x}_{A,i}$, $\mathbf{x}_{B,i}$ – vector-columns of matrices $\mathbf{X}_A$, $\mathbf{X}_B$; $\mathbf{X}^T = (\mathbf{X}_A^T : \mathbf{X}_B^T)$, $\mathbf{y}^T = (\mathbf{y}_A^T : \mathbf{y}_B^T)$. Let's describe the method and calculate its computational complexity. The number of elementary multiplication and division operations is denoted as $OpNum$.

The method of evaluating the parameters and calculating the selection criteria consists of three stages:

- A. Stage of preparatory calculations;
- B. Iteration stage;
- C. Step of the free term calculation for the selected model.

A. Stage of preparatory calculations.

1.1 Centring the elements of the vector-columns of the matrix X and the output vector y according to the formulas

$$\tilde{x}_{ji} = x_{ji} - \bar{x}_{A,i}, \quad \bar{x}_{A,i} = \frac{1}{n_A} \sum_{j=1}^{n_A} x_{ji}, \quad i = \overline{1, m}, \quad j = \overline{1, n_W} \quad (3.1)$$

$$\tilde{y}_j = y_j - \bar{y}_A, \quad \bar{y}_A = \frac{1}{n_A} \sum_{j=1}^{n_A} y_j, \quad j = \overline{1, n_W}. \quad (3.2)$$

Let be $\tilde{X}_A$ - a matrix containing the centred columns of the matrix X_A , $\tilde{X}_B$ - is a matrix containing the columns of the matrix X_B , where the average is subtracted from the elements of each of its i -th columns $\tilde{x}_{A,i}$; $\tilde{y}_A, \tilde{y}_B$ are (3.2) centered vectors y_A, y_B ; $\tilde{x}_{A,i}, \tilde{x}_i$ are vector-columns of matrices $\tilde{X}_A$, $\tilde{X}^T = (\tilde{X}_A^T : \tilde{X}_B^T)$, $i = \overline{1, m}$.

1.2 Calculation of dot products $\tilde{x}_{A,i}^T \tilde{x}_{A,i}$, $i = \overline{1, m}$, and vector $\tilde{X}_A^T \tilde{y}_A$. For the preparatory stage, $OpNum = 2 + 2mn_A$.

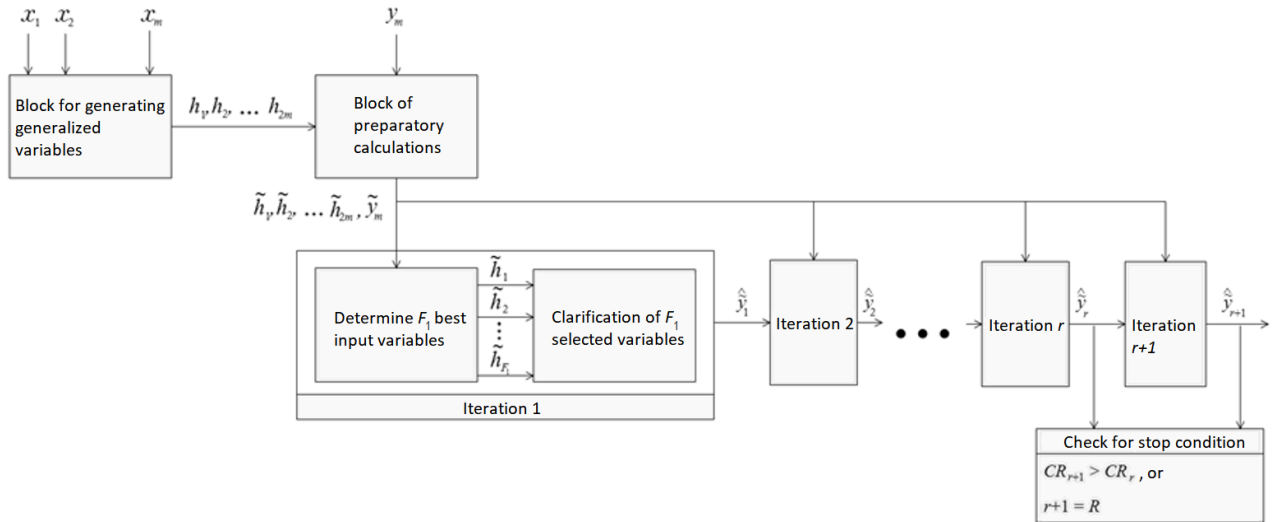


Fig. 3.1 The process of model construction in MSA

B. Iteration phase.

Let $\tilde{x}_{A,l_r}$ be the l_r -th column of the matrix $\tilde{X}_A$ that is added to the model on the r -th iteration; $\tilde{y}_{A,l_r^*}, \tilde{y}_{B,l_r^*}$ - are the output vectors of the best model on the r -th iteration on sets A and B, $l_r^* \in I$; $\tilde{y}_{A,l_r^*,j}, \tilde{y}_{B,l_r^*,i}$ - are elements of vectors $\tilde{y}_{A,l_r^*}, \tilde{y}_{B,l_r^*}$, $j = \overline{1, n_A}, i = \overline{1, n_B}$; $\tilde{y}_{A,j}, \tilde{y}_{B,j}$ are elements of vectors $\tilde{y}_A, \tilde{y}_B$; $\tilde{x}_{A,l_r,j}, \tilde{x}_{B,l_r,i}$ are elements of vectors $\tilde{x}_{A,l_r}, \tilde{x}_{B,l_r}$. The index l_r^* indicates the number of the best argument that was selected on the r -th iteration.

Since it does not matter for the method whether a regression model with multiplicative or linear monomials are built, we will describe it for the last case. The method of estimating coefficients and calculating the model selection criterion consists of two parts:

- 2.1 Construction models at the *first* iteration, $r = 1$;
- 2.2 Construction models at *subsequent* iterations, $r > 1$.

2.1 First iteration, $r = 1$

On the first iteration, you need to find the best model

$$\hat{\mathbf{y}}_{l_1^*}(\tilde{\mathbf{x}}_{1,l_1^*}, \hat{\omega}_{A,1,l_1^*}^1) = \hat{\omega}_{A,1,l_1^*}^1 \tilde{\mathbf{x}}_{1,l_1^*}, \quad l_1^* \in I, \quad (3.3)$$

where the superscript $\hat{\omega}_{A,1,l_1^*}^1$ specifies the ordinal number of the monomial in the current model (in the first iteration, all models contain a single term), and the subindex unit indicates the iteration number.

To define the model (3.3), the following two tasks are solved:

1. Determination of the coefficient estimate $\hat{\omega}_{A,1,l_1}^1$ for each of the models by solving m continuous optimization problems:

$$\hat{\omega}_{A,1,l_1}^1 = \arg \min_{\omega_{A,1,l_1}^1 \in \mathbb{R}} \|\tilde{\mathbf{y}}_A - \omega_{A,1,l_1}^1 \tilde{\mathbf{x}}_{A,l_1}\|^2, \quad \forall l_1 \in I. \quad (3.4)$$

2. Determination of the optimal model of the view $\hat{\mathbf{y}}_{l_1^*}(\tilde{\mathbf{x}}_{1,l_1^*}, \hat{\omega}_{A,1,l_1^*}^1)$ by solving the problem of discrete optimization:

$$l_1^* = \arg \min_{l_1 \in I} \|\tilde{\mathbf{y}}_B - \hat{\omega}_{A,1,l_1}^1 \tilde{\mathbf{x}}_{B,l_1}\|^2. \quad (3.5)$$

When solving problem (3.4), the following algorithm is performed for each $l_1 \in I$:

1. The coefficient estimate $\hat{\omega}_{A,1,l_1}^1$ is calculated as $\hat{\omega}_{A,1,l_1}^1 = \frac{\tilde{\mathbf{x}}_{A,l_1}^T \tilde{\mathbf{y}}_A}{\tilde{\mathbf{x}}_{A,l_1}^T \tilde{\mathbf{x}}_{A,l_1}}$.
2. A solution is found $\tilde{\mathbf{y}}_{A,l_1}$ using the formula $\tilde{\mathbf{y}}_{A,l_1} = \tilde{\mathbf{x}}_{A,l_1} \hat{\omega}_{A,1,l_1}^1$.

To solve problem (3.5), the following algorithm is executed for each $l_1 \in I$:

- 1) The vector is calculated $\tilde{\mathbf{y}}_{B,l_1}$ on sample B : $\tilde{\mathbf{y}}_{B,l_1} = \tilde{\mathbf{x}}_{B,l_1} \hat{\omega}_{A,1,l_1}^1$.
- 2) The regularity criterion is calculated: $AR_{B,l_1} = \sum_{j=1}^{n_B} (\tilde{y}_{B,j} - \tilde{y}_{B,l_1,j})^2$.

Then, among all the values AR_{B,l_1} , $l_1 \in I$, the minimum is chosen, which determines the best model of the first iteration $\hat{\mathbf{y}}_{l_1^*}(\tilde{\mathbf{x}}_{1,l_1^*}, \hat{\omega}_{A,1,l_1^*}^1)$ and the set of F_1 best variables. The number of elementary operations in the first iteration $OpNum = m(1 + n_A) + 2m \cdot n_B$.

2.2 Subsequent iterations, $r > 1$

Let's describe the algorithm for the $(r+1)$ th iteration. For all other iterations, we need to find the best model:

$$\tilde{\mathbf{y}}_{l_{r+1}}^* (\tilde{\mathbf{x}}_{l_{r+1}}^*, \tilde{\mathbf{y}}_{l_r}^*, \hat{\omega}_{A,r+1,l_{r+1}}^{r+1}, \hat{\omega}_{A,r+1,l_r}^*) = \hat{\omega}_{A,r+1,l_{r+1}}^{r+1} \tilde{\mathbf{x}}_{l_{r+1}}^* + \hat{\omega}_{A,r+1,l_r}^* \tilde{\mathbf{y}}_{l_r}^*.$$

For this purpose, the following two tasks are solved:

1. Determination of coefficient estimates $\hat{\omega}_{A,r+1,l_{r+1}}^{r+1}$, $\hat{\omega}_{A,r+1,l_r}^*$ for each model, solving m problems of continuous optimization:

$$(\hat{\omega}_{A,r+1,l_{r+1}}^{r+1}, \hat{\omega}_{A,r+1,l_r}^*)^T = \arg \min_{\omega_{A,r+1,l_{r+1}}^{r+1} \in \mathfrak{R}, \omega_{A,r+1,l_r}^* \in \mathfrak{R}} \|\tilde{\mathbf{y}}_A - \omega_{A,r+1,l_{r+1}}^{r+1} \tilde{\mathbf{y}}_{A,l_r}^* - \omega_{A,r+1,l_r}^{r+1} \tilde{\mathbf{x}}_{A,l_{r+1}}\|^2$$

$$\forall l_{r+1} \in I \quad (3.6)$$

2. Definition of the optimal model $\hat{\mathbf{y}}_{l_{r+1}}^* (\tilde{\mathbf{x}}_{l_{r+1}}^*, \tilde{\mathbf{y}}_{l_r}^*, \hat{\omega}_{A,r+1,l_{r+1}}^{r+1}, \hat{\omega}_{A,r+1,l_r}^*)$, by solving the problem of discrete optimization:

$$l_{r+1}^* = \arg \min_{l_{r+1} \in I} \|\tilde{\mathbf{y}}_B - \hat{\omega}_{A,r+1,l_{r+1}} \tilde{\mathbf{y}}_{B,l_r}^* - \hat{\omega}_{A,r+1,l_r}^{r+1} \tilde{\mathbf{x}}_{B,l_{r+1}}\|^2. \quad (3.7)$$

When solving problem (3.6), the following algorithm is performed for each $l_{r+1} \in I$:

- 1) The value of $\hat{d}_{l_{r+1}} = \frac{\tilde{\mathbf{y}}_{A,l_r}^{*T} \tilde{\mathbf{x}}_{A,l_{r+1}}}{\tilde{\mathbf{y}}_{A,l_r}^{*T} \tilde{\mathbf{y}}_{A,l_r}^*}$.
- 2) The vector is calculated $\hat{\tilde{\mathbf{x}}}_{A,l_{r+1}}$ using the formula $\hat{\tilde{\mathbf{x}}}_{A,l_{r+1}} = \hat{d}_{l_{r+1}} \tilde{\mathbf{y}}_{A,l_r}^*$.
- 3) The coefficient is estimated $\hat{\omega}_{A,r+1,l_{r+1}}^{r+1}$ according to the formula

$$\hat{\omega}_{A,r+1,l_{r+1}}^{r+1} = \frac{(\tilde{\mathbf{x}}_{A,l_{r+1}} - \hat{\tilde{\mathbf{x}}}_{A,l_{r+1}})^T \tilde{\mathbf{y}}_A}{(\tilde{\mathbf{x}}_{A,l_{r+1}} - \hat{\tilde{\mathbf{x}}}_{A,l_{r+1}})^T (\tilde{\mathbf{x}}_{A,l_{r+1}} - \hat{\tilde{\mathbf{x}}}_{A,l_{r+1}})}. \quad (3.8)$$

- 4) The vector of estimating the coefficients of previous iterations by formulas is corrected

$$\hat{\omega}_{A,r+1,l_{r+1}} = (1 - \hat{\omega}_{A,r+1,l_{r+1}}^{r+1}) \hat{d}_{l_{r+1}}, \quad (3.9)$$

$$\hat{\Omega}_{A,r+1,l_{r+1}}^r = \hat{\Omega}_{A,r,l_r}^r \hat{\omega}_{A,r+1,l_{r+1}}, \quad (3.10)$$

where $(\hat{\Omega}_{A,r,l_r}^r)^T = (\hat{\omega}_{A,r,l_1}^1, \hat{\omega}_{A,r,l_2}^2, \dots, \hat{\omega}_{A,r,l_r}^r)$, $l_i^* \in I$, $i = \overline{1, r}$ - the evaluation vector obtained at the r -th iteration corresponding to the vector-columns of the matrix $\tilde{\mathbf{G}}_{A,l_r}^*$; $\tilde{\mathbf{G}}_{A,l_r}^*$ - is a matrix composed of vector-columns of the matrix $\tilde{\mathbf{X}}_A$ in the order in which the regressors are included in the model $\tilde{\mathbf{y}}_{l_r}^* (\tilde{\mathbf{x}}_{l_r}^*, \tilde{\mathbf{y}}_{l_{r-1}}^*, \hat{\Omega}_{A,r,l_r}^r)$, the indices of which correspond to the value l_i^* in $\hat{\omega}_{A,l_i,l_i}^i$, $i = \overline{1, r}$, $\dim \tilde{\mathbf{G}}_{A,l_r}^* = n_A \times r$. The upper index r in $\hat{\Omega}_{A,r,l_r}^r$ determines the number of arguments included in the model, and the lower r - is the iteration number. Then the vector of estimates of the coefficients $(r+1)$ -th iteration for some $l_{r+1} \in I$ take the form:

$$(\hat{\Omega}_{A,r+1,l_{r+1}}^{r+1})^T = (\hat{\omega}_{A,r+1,l_1}^1, \hat{\omega}_{A,r+1,l_2}^2, \dots, \hat{\omega}_{A,r+1,l_{r+1}}^{r+1}) = ((\hat{\Omega}_{A,r+1,l_{r+1}}^r)^T : \hat{\omega}_{A,r+1,l_{r+1}}^{r+1}).$$

The solution vector $(r + 1)$ of the iteration is calculated using the formula

$$\hat{\mathbf{y}}_{A,l_{r+1}} = \hat{\omega}_{A,r+1,l_{r+1}}^{r+1} \tilde{\mathbf{x}}_{A,l_{r+1}} + \hat{\omega}_{A,r+1,l_{r+1}} \hat{\mathbf{y}}_{A,l_r^*}.$$

Note, that

$$\hat{\mathbf{y}}_{A,l_{r+1}} = \tilde{\mathbf{G}}_{A,l_{r+1}} \hat{\Omega}_{A,r+1,l_{r+1}}^{r+1} = (\tilde{\mathbf{G}}_{A,l_r^*} : \tilde{\mathbf{x}}_{A,l_{r+1}}) \cdot \begin{pmatrix} \hat{\Omega}_{A,r+1,l_{r+1}}^r \\ \dots\dots\dots \\ \hat{\omega}_{A,r+1,l_{r+1}}^{r+1} \end{pmatrix}$$

When solving (3.5), the following algorithm is executed for all $l_{r+1} \in I$:

- 1) The vector $\hat{\mathbf{y}}_{B,l_{r+1}}$ of $(r + 1)$ th iteration on sample B is calculated:

$$\hat{\mathbf{y}}_{B,l_{r+1}} = \tilde{\mathbf{G}}_{B,l_{r+1}} \hat{\Omega}_{A,r+1,l_{r+1}}^{r+1},$$

where $\tilde{\mathbf{G}}_{B,l_{r+1}}$ is the matrix composed of vector-columns of the matrix $\tilde{\mathbf{X}}_B$ in the order in which the arguments are included in the model $\hat{\mathbf{y}}_{l_{r+1}}$, whose indices correspond to values l_i^* , $i = \overline{1, r}$ and l_{r+1} in the elements of the vector $\hat{\Omega}_{A,r+1,l_{r+1}}^{r+1}$.

- 2) The regularity criterion is calculated:

$$AR_{B,l_{r+1}} = \sum_{j=1}^{n_B} (\tilde{y}_{B,j} - \hat{y}_{B,l_{r+1},j})^2.$$

Among all the values $AR_{B,l_{r+1}}$, $l_{r+1} \in I$, the minimal is chosen, that determines the best model of $(r+1)$ -th iteration $\hat{\mathbf{y}}_{l_{r+1}^*}^* (\tilde{\mathbf{x}}_{l_{r+1}^*}^*, \hat{\mathbf{y}}_{l_r^*}^*, \hat{\Omega}_{A,r+1,l_{r+1}^*}^{r+1})$ and is used to check the algorithm's stop condition.

The number of operations in the second stage of the iteration process is $OpNum = m(rn_B + 7n_A + 2n_B + 3)$.

Suppose the algorithm stops at iteration with the number R. When the algorithm is finished, for the best model $\hat{\mathbf{y}}_{l_R^*}^* (\tilde{\mathbf{x}}_{l_R^*}^*, \hat{\mathbf{y}}_{l_{R-1}^*}^*, \hat{\omega}_{A,R,l_R^*}^R, \hat{\omega}_{A,R,l_R^*})$ is executed the third stage of the method.

C. Step of the free term calculation for the selected model

3.1 Estimates of the coefficients vector are found $\hat{\Theta}_{A,m}^T = (\hat{\theta}_{A,1}, \hat{\theta}_{A,2}, \dots, \hat{\theta}_{A,m})$ of the model $\hat{y}^*(\tilde{x}_1, \dots, \tilde{x}_m, \hat{\Theta}_{A,m})$ by summing the coefficients $\hat{\omega}_{A,r,l_r^*}^r$, $r = \overline{1, R}$, whose index l_r^* is equal to the index j of the coefficient $\hat{\theta}_{A,j}$, $j = \overline{1, m}$.

3.2 Estimate of the free term of the model $\hat{y}_{A,0}^*(x_1, \dots, x_m, \hat{\theta}_{A,m+1})$ is calculated

$$\hat{\theta}_{A,0} = \bar{y}_A - \sum_{i=1}^m \hat{\theta}_{A,i} \bar{x}_{A,i}.$$

3.1.2 Computational complexity of MSA

In MSA the definition of argument included in the additive-multiplicative model is carried out using truncated combinatorial sorting. Considering that in the general case it is impossible to find the exact number of models-candidates when searching at each iteration, the number of operations of the entire MSA is also calculated for the case of constructing linear models.

Since the freedom of choice of solutions $F = 1$, the number of models-candidates built on each iteration is equal to m . I.e., when constructing linear models, the algorithm performs a complete

combinatorial sorting of all models-candidates from the original variables at each iteration. If we limit the number of iterations of the algorithm to m , then the total number of models that the algorithm builds is equal to m^2 . Table 2.1 shows the number of operations at each stage of the MSA.

The total number of MSA operations is equal to

$$2 + m(3n_A + 2n_B + 1) + m(m-1)(0.5mn_B + 7n_A + 2n_B + 3), \quad (3.11)$$

moreover, the function of calculating the computational complexity of one model at the r -th iteration of the MSA has the form

$$Q(r, n_A, n_B) = n_B r + 7n_A + 2n_B + 3. \quad (3.12)$$

Table 3.1 – Computational complexity of each stage of MSA

Stages	Stages	Number of operations
Preparatory		$2 + 2mn_A$
Iteration stage (for one model)	Iteration 1	$m(1+n_A) + 2mn_B$
	Iterations 2 to m	$\sum_{r=1}^{m-1} m(n_B r + 7n_A + 2n_B + 3)$

If in (3.12) $n_B \geq r$ (n_B is the number of points of the test sample), then during construction of one model on this iteration, the number of operations is a quadratic function of r , and if $r \geq n_B$, then it is a quadratic function of n_B .

From the analysis of the method for estimating coefficients and calculating the selection criterion, it is obvious that its computational complexity depends on the number of observations n_W , and, in particular, has a quadratic complexity in the number of arguments for $n_B \geq r$ and in the number of observations n_B when $r \geq n_B$, which is a disadvantage of the method and the algorithm as a whole.

Let's consider the method of construction additive-multiplicative models in MSA.

3.1.3 A way to find the best multiplicative monomial

When constructing nonlinear models with respect to arguments, at each iteration the algorithm searches for the best argument to add to the model. This is done in three stages:

- 1) Determine the F_1 best models that depend on the initial variables, for which m models of the form are built:

$$\hat{\mathbf{y}}_{l_1}(\tilde{\mathbf{x}}_{1,l_1}, \hat{\omega}_{A,1,l_1}^1) = \hat{\omega}_{A,1,l_1}^1 \tilde{\mathbf{x}}_{1,l_1}, \quad \forall l_1 \in I, \quad r = 1, \quad (3.13)$$

$$\hat{\mathbf{y}}_{l_{r+1}}(\tilde{\mathbf{x}}_{l_{r+1}}, \hat{\mathbf{y}}_{l_r}^*, \hat{\omega}_{A,r+1,l_{r+1}}^{r+1}, \hat{\omega}_{A,r+1,l_{r+1}}) = \hat{\omega}_{A,r+1,l_{r+1}}^{r+1} \tilde{\mathbf{x}}_{l_{r+1}} + \hat{\omega}_{A,r+1,l_{r+1}} \hat{\mathbf{y}}_{l_r}^*, \quad \forall l_{r+1} \in I, \quad r > 1 \quad (3.14)$$

where $\tilde{\mathbf{x}}_{1,l_1}, \tilde{\mathbf{x}}_{l_{r+1}}, \forall l_1, l_{r+1} \in I$ are vector-columns of the matrix $\mathbf{X}$ centred on the training sample; $\hat{\mathbf{y}}_{l_r}^*$ - is solution found at the r -th iteration for vector $\mathbf{y}$ centred on sample A; $\hat{\omega}_{A,r+1,l_{r+1}}^{r+1}, \hat{\omega}_{A,r+1,l_{r+1}}$ - coefficient estimates found by LSM. Among the m constructed models (3.13) or (3.14), are selected the best F_1 according to minimum of CR criterion.

- 2) For each of the selected F_1 models, a multiplicative monomial design procedure (MMDP) is performed.
- 3) Selection of one of the best model among the F_1 constructed.

To describe the MMDP, let's introduce notations: let the vector $\mathbf{x}_{A,l_{r+1}}$, correspond the variable $x_{A,l_{r+1}}$, $l_{r+1} \in I$, and to the vector $\mathbf{x}_{A,k}$ correspond the variable $x_{A,k}$, $k = \overline{1, m}$. The input parameter of MMDP is *multNum* – the maximum number of multiplications, that is allowed in the formation of a multiplicative variable $z_{A,l_{r+1}}$, the multiplication basis of which is variable $x_{A,l_{r+1}}$. Let *it* be the ordinal number of the performed multiplications; CR_{it} – the values of the CR criterion for the best model of the multiplication iterations with the number *it*; $z_{A,l_{r+1},it}$ is a multiplicative monomial obtained on iteration of multiplications with the number *it*. Provided that $it < multNum$, the algorithm constructs models of the form:

$$\hat{\mathbf{y}}_{A,l_{r+1},it}(\tilde{\mathbf{z}}_{A,l_{r+1},it}, \hat{\mathbf{y}}_{A,l_r}^*, \hat{\omega}_{A,r+1,l_{r+1}}^{r+1}, \hat{\omega}_{A,r+1,l_{r+1}}) = \hat{\omega}_{A,r+1,l_{r+1}}^{r+1} \tilde{\mathbf{z}}_{A,l_{r+1},it} + \hat{\omega}_{A,r+1,l_{r+1}} \hat{\mathbf{y}}_{A,l_r}^*, \forall l_{r+1} \in I,$$

where the elements of the vector $\tilde{\mathbf{z}}_{A,l_{r+1},it}$ are centred on sample A by the values of the corresponding elements of the vector $\mathbf{z}_{A,l_{r+1},it}$.

Figure 3.2 shows the complete tree of structures of nonlinear models, which can be formed by the algorithm if the freedom of choice the best multiplicative arguments at each level of the tree is not restricted. In fact, the procedure for finding the best multiplicative argument limits this freedom, namely: only one best child model is selected for each parent, the number of which is F_1 .

The multiplication process stops if:

- the value of criterion for the descendant models is no better, than the value of criterion for the parent model, the multiplication process stops;
- the maximum number of multNum multiplications has been reached.

In addition to the tree itself, Fig. 3.2 illustrates the process of forming variables $z_{A,l_{r+1}}$ by means of multiplying the variables x_i , $i \in I$ - nodes of the tree.

It should be noted that this procedure involves a large number of node repetitions, and therefore, candidate model calculations. Chapter 1 in Figure 1.10 also provides an overview of this procedure, where the repetition rate is 24%. In Figure 3.2, this rate is 40%.

In addition, this procedure, due to its truncated nature, does not guarantee the construction of all possible nonlinear models in a given iteration. In the example in Figure 3.2, four monomials present in the full sorting are excluded: x_2^3 , $x_2^2 x_1$, x_3^3 , $x_3^2 x_1$.

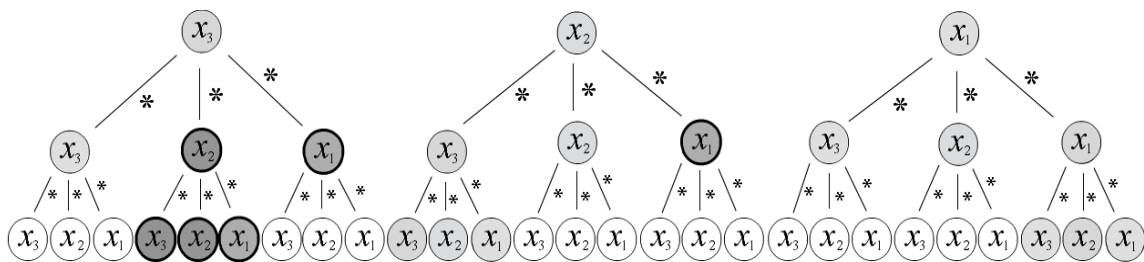


Figure 3.2 – An example of constructing a tree of structures of nonlinear models:
a light grey circle is a constructed model, a white circle is an unconstructed model, and
a dark grey circle is a repetition of a constructed model.

It should be noted that at each non-zero iteration of the multiplication, the selection of the best models is realized by selecting one best descendant model for each parent model, rather than by selecting the best F_1 among all the descendant models. This fact explains the rationale for generating all possible descendant models for each parent model.

Taking into account the analysis performed, MSA has a number of shortcomings.

Let's list them in descending order of weight:

- The computational complexity of the method for estimating coefficients and calculating the criteria for selecting the model depends on the number of observations. Moreover, at $n_B \geq r$ (r is the number of the current iteration that coincides with the number of arguments added to the model), the number of operations to construct one model of this iteration has a quadratic dependence on r , and if $n_B \leq r$, the number of operations is a quadratic function of the number of observations n_B .
- The method of constructing a multiplicative monomial does not guarantee that the global minimum of the CR criterion will be found in the current iteration.
- When constructing a multiplicative monomial, more operations are performed than necessary, since there are repeated PDF generations.
- The network structure of the algorithm on the iterations of summation of monomials implements the principle of inconclusive decisions, the freedom of choice of the algorithm $F = 1$.

The advantages of MSA include saving RAM, since only two matrices $\mathbf{H}$ and $\tilde{\mathbf{H}}$, containing the original and converted input variables are preserved. Memory for multiplicative monomials is allocated and released dynamically at each iteration when they are constructed.

3.2 Generalized relaxation iterative algorithm (GRIA)

The reason for the main disadvantage of MSA – the dependence of the computational complexity of the method of construction models on the number of observations – is the calculation of the elements of the covariance matrix $\tilde{\mathbf{W}}^T \tilde{\mathbf{W}}$, $\tilde{\mathbf{W}} = (\tilde{\mathbf{H}} : \tilde{\mathbf{y}})$ for each calculation of the parameters and criteria for the models-candidates selection. The generalized relaxation iterative algorithm (GRIA) avoids this drawback by constructing models based on the calculation of matrices of normal equations, and other disadvantages listed above with the help of other improvements. In Chapter 1, during the review of sorting-out algorithms, it was mentioned that this class of algorithms also uses the results of calculation of a matrix of normal equations in order to achieve independence of the computational complexity of the model construction stage from the number of observations. However, in general, taking into account the calculation of this matrix, the computational complexity of all modeling algorithms depends on the number of observations.

As a result of the analysis of known GMDH algorithms, the following components were identified in [17]:

- A. class of models;
- B. generator of model structures;
- C. method of parameter evaluation;
- D. algorithm for calculating the selection criterion.

The GRIA is based on the following MSA ideas:

1. adding only one monomial to the model at each iteration;
2. centring of arguments to avoid evaluation of the free term;
3. truncated sorting (but without repetitions) in the search for the best multiplicative monomial.

The following description of the generalized relaxation algorithm allows us to explore the improvements that make GRIA the fastest of the GMDH algorithms and the most accurate in the class of iterative GMDH algorithms.

We will describe GRIA in a similar way to MSA: first, we describe the model construction process, then its architecture, and finally note its computational complexity.

3.2.1 Model construction process

Figure 3.3 shows the process of construction a model, which consists of three stages:

- the stage of transformation of the $\mathbf{X}$ matrix, where a class of models is formed;
- the preparatory stage, where auxiliary calculations are performed, the result of which is necessary at the iteration stage;
- the stage of iterations, at which the construction of models is carried out.

In this case, the freedom of choice $F \geq 1$ the best models, which move on to next iteration is introduced into the model construction process.

GRIA, as well as MSA, in general, constructs both static and dynamic additive-multiplicative models of linear regression of the form:

$$f_{\alpha}(v_1, \dots, v_m, \Theta) = \theta_0 + \sum_{i=1}^m \theta_i f(v_i) + \sum_{i=1}^m \sum_{j=i}^m \theta_{i,j} f(v_i) \cdot f(v_j) + \sum_{i=1}^m \sum_{j=ik=i+j}^m \sum_{k=i+j}^m \theta_{i,j,k} f(v_i) \cdot f(v_j) \cdot f(v_k) + \dots, \quad (3.15)$$

where $f(v) \in \psi$ is some function or superposition of functions from a given set of functions and/or operators (including difference operators), v is any element of the row of matrix $\mathbf{X}$.

Example of a superposition: $f(v_i) = (\nabla^{-1} v_i)^{-1} = (v_{i,-1})^{-1}$, where ∇^{-1} is the shift operator by 1 step relative to the current time of the i -th variable of the matrix $\mathbf{X}$, and $(v_{i,-1})^{-1}$ is the inverse function of lag variable, $\Theta = (\theta_0, \theta_{i,j}, \dots)$ - vector of unknown parameters.

The first stage of GRIA consists of a sequence of two stages:

- 1) implementation of unary transformations of vector-columns of the matrix $\mathbf{X}$;
- 2) implementation of multiplicative transformations of vector-columns.

The basis for performing unary transformations is the set of functional transformations $\mathfrak{F}$, which consists of two functions: $\mathfrak{F} = \{\varphi(x), \nabla^{-j}\}$, where $\varphi(\cdot)$ is a power function. The output of the unary transformation procedure is a matrix $\mathbf{H}$. After performing unary transformations, it is possible to perform k -th transformation (multiplication) over vector-columns of matrix $\mathbf{H}$. Matrix $\mathbf{Z}$ is formed, consisting of all possible products of the elements of vector-rows of matrix $\mathbf{H}$. The output of this procedure is an extended matrix $\mathbf{Q} = (\mathbf{H}:\mathbf{Z})$, which has the dimension $\dim \mathbf{Q} = n_W \times (C_{m+pow}^m)$, where $pow > 0$ is the given maximum power in monomial of the polynomial (3.15).

Without loss of generality, we will demonstrate the stage of preparatory calculations of the matrix $\tilde{\mathbf{W}}^T \tilde{\mathbf{W}}$, ($\tilde{\mathbf{W}} = (\tilde{\mathbf{Q}} : \tilde{\mathbf{y}})$, $\tilde{\mathbf{Q}} = (\tilde{\mathbf{X}} : \tilde{\mathbf{Z}})$) using the example of the initial (not transformed) input variables of the matrix: $\tilde{\mathbf{W}} = (\tilde{\mathbf{X}} : \tilde{\mathbf{y}})$. First, in accordance with [56], at the stage of implementation of preparatory calculations (Fig. 3.3), the centering of the variables is carried out. According to formulas (3.1) and (3.2), all the elements of the columns of the matrix $\mathbf{W}$ are transformed by subtracting the average value of the elements of the vector $\mathbf{y}_A$ and for the matrix $\mathbf{X}_A$ – the average values of corresponding elements of the columns belonging to the training sample A . Elements of the obtained covariance matrices:

$$\tilde{\mathbf{W}}_A^T \tilde{\mathbf{W}}_A = \begin{pmatrix} \tilde{\mathbf{X}}_A^T \tilde{\mathbf{X}}_A & \tilde{\mathbf{X}}_A^T \tilde{\mathbf{y}}_A \\ \tilde{\mathbf{y}}_A^T \tilde{\mathbf{X}}_A & \tilde{\mathbf{y}}_A^T \tilde{\mathbf{y}}_A \end{pmatrix}, \quad \tilde{\mathbf{W}}_B^T \tilde{\mathbf{W}}_B = \begin{pmatrix} \tilde{\mathbf{X}}_B^T \tilde{\mathbf{X}}_B & \tilde{\mathbf{X}}_B^T \tilde{\mathbf{y}}_B \\ \tilde{\mathbf{y}}_B^T \tilde{\mathbf{X}}_B & \tilde{\mathbf{y}}_B^T \tilde{\mathbf{y}}_B \end{pmatrix}, \quad (3.16)$$

are used at the iteration stage.

Let's consider the architecture of this algorithm.

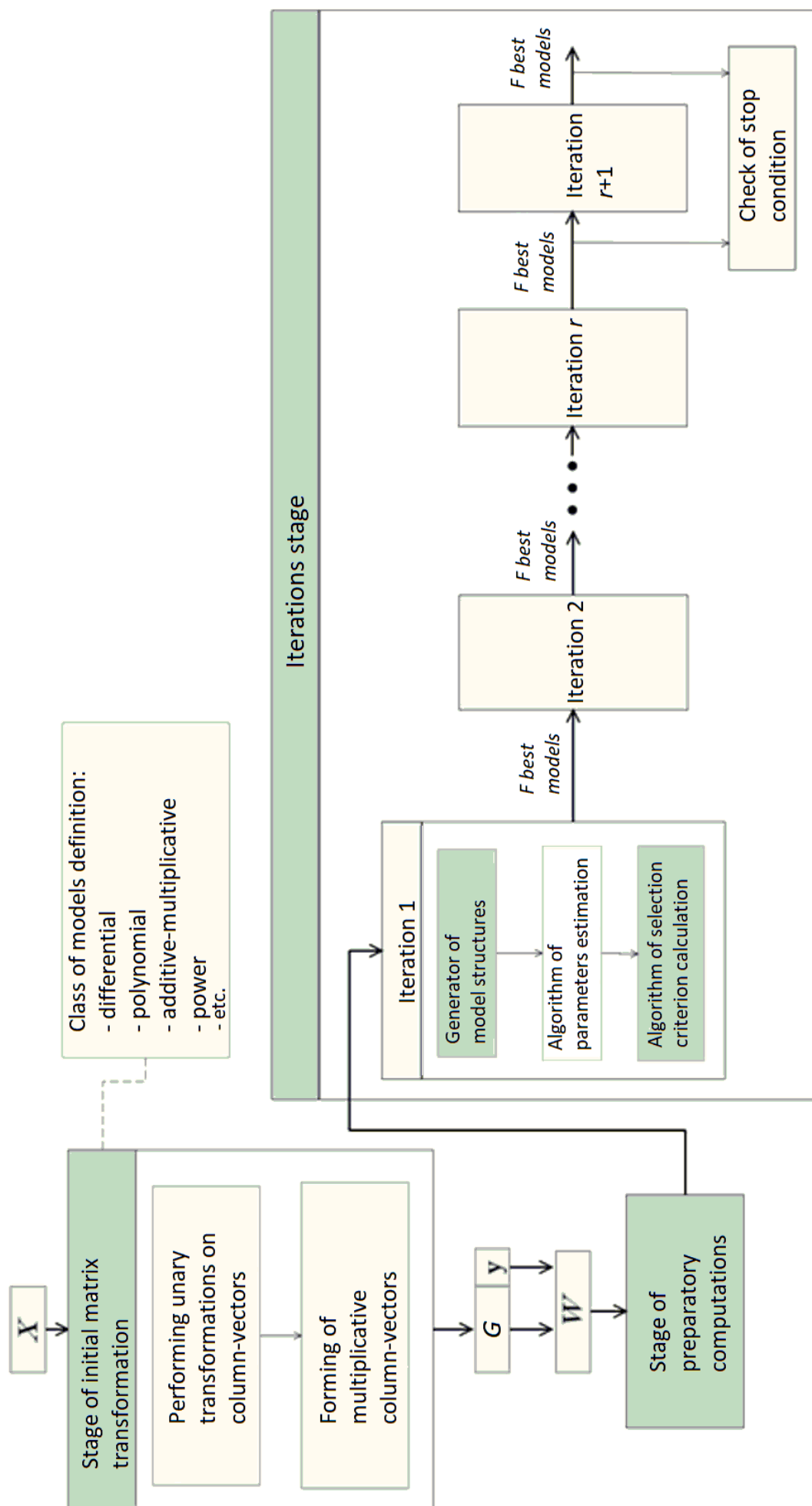


Figure 3.3. The process of construction a model in GRIA

3.2.2 Algorithm architecture

The following three components can be distinguished in the GRIA:

1. *Model class*. The algorithm implements the following models of many variables: power, additive-multiplicative and their difference analogues.
2. *Generators* of partial description structures with full and truncated sorting.
3. *Method for estimating parameters and selection criteria calculation*: a method with non-recurrent calculations; two methods with recurrent calculations.

Each of the three GRIA components is independent of the others, meaning that different RIAs can be constructed by using one variant of each component. Note that for the "model class" component, a combination of variants is possible, since, for example, a difference-polynomial model can be constructed. In the GRIA structure, the "model class" component does not specify a specific relaxation iterative algorithm, since it is essentially a component of the input data transformation block. Given a matrix $\mathbf{X}$ and performing the appropriate transformations on its column vectors, the specified model class can be formed. In this case, the generated matrix $\mathbf{H}$ is considered as the initial one in any possible RIA.

All possible variants of RIA algorithms combined in GRIA are presented in Fig. 3.4.

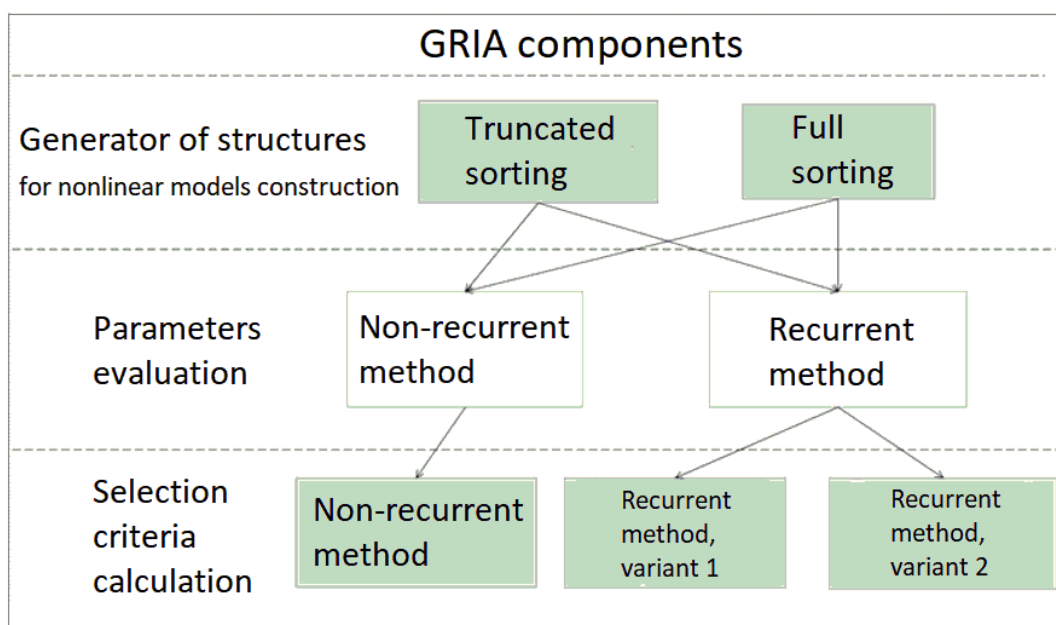


Figure 3.4 – Scheme for obtaining various RIA by combining the components of the GRIA components

In the analysis of the MSA, it was noted that its method of finding the best multiplicative monomial has a number of drawbacks related to the structure of tree on which the sorting is performed. A similar procedure in GRIA, free of these drawbacks is described in the next section.

3.2.3 Method for finding a multiplicative monomial

In GRIA, the sorting for the best multiplicative monomial is performed using a prefix tree [47]. A prefix tree is an abstract data structure that allows storing and searching an associative array whose keys are strings. To construct the tree, an alphabet of symbols $\aleph$ is specified, from which words (strings) are composed. The key value is obtained by scanning all parent nodes, each of which stores one or more symbols of the alphabet, and concatenating the symbols in the order they are scanned.

In our case, the alphabet of symbols $\mathfrak{S}$ is the indices of the vector-columns of the matrix X : to the set of indices $\mathfrak{S} = \{ '0', '1', \dots, 'm-1' \}$ corresponds the set $\{ 'x_1', 'x_2', \dots, 'x_m' \}$. The words of this alphabet are, for example '000', which corresponds to the transformation of vectors x_1, x_1, x_1 in the form of $diag(diag(x_1) \cdot x_1) \cdot x_1$, the word '012' is the transformation of vectors x_1, x_2, x_3 etc.

An illustration of model structure search using a prefix tree is shown in Figure 3.5. The tree shown is minimal (without repetitions). This roughly increases the performance of GRIA compared to MSA by 20% to 40%.

The improved monomial search method, which uses the proposed prefix tree traversal algorithm, is more flexible than MSA. It allows for varying both the number of best $F1$ models, which are selected at the zeroth iteration of the multiplication, and the number of best $F2$ models, which are selected at all other iterations of the multiplication except the zeroth. If all generated models proceed to the next iteration, a complete sorting is performed.

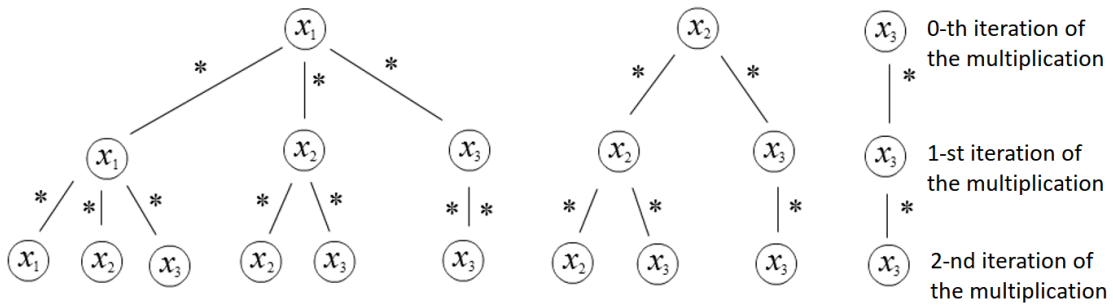


Figure 3.5 – Tree for finding the best multiplicative argument in GRIA

It is this property that allows us to prove the convergence of GRIA when constructing nonlinear models, as will be demonstrated below. If $F2 = 1$, the GRIA tree traversal algorithm is similar to the model structure construction algorithm in MSA. The sets of model structures traversed by these algorithms are the same when constructing linear models. The advantage of this approach is evident when constructing nonlinear models.

It is this property that makes it possible to prove the convergence of GRIA in the case of construction nonlinear models, which will be shown below. If you specify $F2 = 1$, the algorithm for traversing the GRIA tree is similar to the algorithm for construction the model structure in MSA. The advantage of this approach is manifested when construction nonlinear models.

The generalized relaxation iterative algorithm described above is known in two modifications, with non-recurrent and recurrent calculations of model parameter estimates and criterion values. We will discuss the same algorithm, GRIA, whose two modifications are identical in the first two stages and differ only in the third stage, in which models are constructed iteratively using elements of the covariance matrices of samples A and B.

Next, we present non-recurrent and recurrent methods for estimating model parameters and selection criteria, and compare them with the estimation method in the known simplified multi-layered algorithm.

3.2.4 Methods for estimating parameters and calculating selection criteria

Without loss of generality, we will describe the method using the example of matrix X , assuming that the chain of transformations of the initial variables $X \rightarrow H \rightarrow Q \rightarrow \tilde{W}$ does not affect the computational complexity of the iteration stage. For simplicity, the freedom of choice of models

is set to $F = 1$. The methods described below will differ from the parameter estimation method in MSA at iterations with numbers $r > 1$.

Non-recurrent method

On the $(r+1)$ -th iteration, m models of the form are constructed:

$$\tilde{\mathbf{y}}_{A,l_{r+1}} = \hat{\omega}_{A,r+1,l_{r+1}}^{r+1} \tilde{\mathbf{x}}_{A,l_{r+1}} + \hat{\omega}_{A,r+1,l_{r+1}} \tilde{\mathbf{y}}_{A,l_r^*} = \tilde{\mathbf{G}}_{A,l_{r+1}} \hat{\Omega}_{A,r+1,l_{r+1}}^{r+1}, \quad l_{r+1} \in I, \quad (3.17)$$

where I is the set of indices of the vector-columns of the matrix $\mathbf{X}$. The matrix of normal equations in this case looks like this:

$$\begin{pmatrix} \tilde{\mathbf{y}}_{A,l_r^*}^T \tilde{\mathbf{y}}_{A,l_r^*} & \tilde{\mathbf{y}}_{A,l_r^*}^T \tilde{\mathbf{x}}_{A,l_{r+1}} \\ \tilde{\mathbf{x}}_{A,l_{r+1}}^T \tilde{\mathbf{y}}_{A,l_r^*} & \tilde{\mathbf{x}}_{A,l_{r+1}}^T \tilde{\mathbf{x}}_{A,l_{r+1}} \end{pmatrix} \begin{pmatrix} \omega_{A,r+1,l_{r+1}}^{r+1} \\ \omega_{A,r+1,l_{r+1}} \end{pmatrix} = \begin{pmatrix} \tilde{\mathbf{y}}_{A,l_r^*}^T \tilde{\mathbf{y}}_A \\ \tilde{\mathbf{x}}_{A,l_{r+1}}^T \tilde{\mathbf{y}}_A \end{pmatrix}, \quad l_{r+1} \in I, \quad (3.17)$$

Let $a_{A,l_r^*} = \tilde{\mathbf{y}}_{A,l_r^*}^T \tilde{\mathbf{y}}_{A,l_r^*}$, $b_{A,l_{r+1}} = \tilde{\mathbf{y}}_{A,l_r^*}^T \tilde{\mathbf{x}}_{A,l_{r+1}}$, $c_{A,l_r^*} = \tilde{\mathbf{y}}_{A,l_r^*}^T \tilde{\mathbf{y}}_A$. Then the estimates of the coefficients for $l_{r+1} \in I$ are calculated by the formulas

$$\hat{\omega}_{A,r+1,l_{r+1}} = \frac{c_{A,l_r^*}(\tilde{\mathbf{x}}_{A,l_{r+1}}^T \tilde{\mathbf{x}}_{A,l_{r+1}}) - b_{A,l_{r+1}}(\tilde{\mathbf{x}}_{A,r+1,i}^T \tilde{\mathbf{y}}_A)}{a_{A,l_r^*}(\tilde{\mathbf{x}}_{A,l_{r+1}}^T \tilde{\mathbf{x}}_{A,l_{r+1}}) - b_{A,l_{r+1}}^2},$$

$$\hat{\omega}_{A,r+1,l_{r+1}}^{r+1} = \frac{a_{A,l_r^*}(\tilde{\mathbf{x}}_{A,r+1,i}^T \tilde{\mathbf{y}}_A) - b_{A,l_{r+1}} c_{A,l_r^*}}{a_{A,l_r^*}(\tilde{\mathbf{x}}_{A,r+1,i}^T \tilde{\mathbf{x}}_{A,r+1}) - b_{A,l_{r+1}}^2},$$

where the scalars a_{A,l_r^*} , $b_{A,l_{r+1}}$, c_{A,l_r^*} , $l_{r+1} \in I$ are:

$$\begin{aligned} a_{A,l_r^*} &= (\hat{\Omega}_{A,r,l_r^*}^r)^T \tilde{\mathbf{G}}_{A,l_r^*}^T \tilde{\mathbf{G}}_{A,l_r^*} \hat{\Omega}_{A,r,l_r^*}^r = \\ &= \sum_{i=1}^r (\tilde{\mathbf{g}}_{A,l_i^*}^T \tilde{\mathbf{g}}_{A,l_i^*}) (\hat{\omega}_{A,r,l_i^*}^i)^2 + 2 \sum_{i=1}^r \sum_{j=i+1}^r (\tilde{\mathbf{g}}_{A,l_i^*}^T \tilde{\mathbf{g}}_{A,l_j^*}) \cdot \hat{\omega}_{A,r,l_i^*}^i \cdot \hat{\omega}_{A,r,l_j^*}^j, \\ b_{A,l_{r+1}} &= \tilde{\mathbf{y}}_{A,l_r^*}^T \tilde{\mathbf{x}}_{A,l_{r+1}} = \tilde{\mathbf{x}}_{A,l_{r+1}}^T \tilde{\mathbf{G}}_{A,l_r^*}^T (\hat{\Omega}_{A,r,l_r^*}^r)^T = \sum_{j=1}^r (\tilde{\mathbf{x}}_{A,l_{r+1}}^T \tilde{\mathbf{g}}_{A,l_j^*}) \cdot \hat{\omega}_{A,r,l_j^*}^j, \quad (3.18) \end{aligned}$$

$$c_{A,l_r^*} = \tilde{\mathbf{y}}_{A,l_r^*}^T \tilde{\mathbf{y}}_A = \tilde{\mathbf{y}}_A^T \tilde{\mathbf{G}}_{A,l_r^*}^T (\hat{\Omega}_{A,r,l_r^*}^r)^T = \sum_{j=1}^r (\tilde{\mathbf{y}}_A^T \tilde{\mathbf{g}}_{A,l_j^*}) \cdot \hat{\omega}_{A,r,l_j^*}^j,$$

where $\tilde{\mathbf{g}}_{A,l_j^*}$ are the vector-columns of the matrix $\tilde{\mathbf{G}}_{A,l_r^*}^T$, $i = \overline{1, r}$.

Then the estimates of the coefficients of the previous iterations are adjusted according to the formula:

$$\hat{\Omega}_{A,r+1,l_{r+1}}^r = \hat{\Omega}_{A,r,l_r^*}^r \hat{\omega}_{A,r+1,l_{r+1}}$$

For all models (3.17), the selection criterion is calculated, based on the data from the testing sample B

$$AR_{B,l_{r+1}} = \tilde{\mathbf{y}}_B^T \tilde{\mathbf{y}}_B - 2 \tilde{\mathbf{y}}_B^T \tilde{\mathbf{G}}_{B,l_{r+1}} \hat{\Omega}_{A,r+1,l_{r+1}}^{r+1} + (\hat{\Omega}_{A,r+1,l_{r+1}}^{r+1})^T \tilde{\mathbf{G}}_{B,l_{r+1}}^T \tilde{\mathbf{G}}_{B,l_{r+1}} \hat{\Omega}_{A,r+1,l_{r+1}}^{r+1}$$

or

$$RSS_{A,l_{r+1}} = \tilde{\mathbf{y}}_A^T \tilde{\mathbf{y}}_A - 2\tilde{\mathbf{y}}_A^T \tilde{\mathbf{G}}_{A,l_{r+1}} \hat{\Omega}_{A,r+1,l_{r+1}}^{r+1} + (\hat{\Omega}_{A,r+1}^{r+1})^T \tilde{\mathbf{G}}_{A,l_{r+1}}^T \tilde{\mathbf{G}}_{A,l_{r+1}} \hat{\Omega}_{A,r+1,l_{r+1}}^{r+1},$$

by the minimum value of which the best argument and parameters of the model (3.17) are found, i.e. the triple:

$$(\hat{\omega}_{A,r+1,l_{r+1}}^{r+1}, \hat{\omega}_{A,r+1,l_{r+1}}^*, \tilde{\mathbf{x}}_{A,l_{r+1}}^*) = \arg \min_{l_{r+1} \in I} AR_{B,l_{r+1}}.$$

After the process of iterative model construction is completed, in order to represent the model through the original variables, i.e. to move from the centred variables to the original space $\mathbf{y}$ and $\mathbf{x}_i$, it is necessary to calculate only the free term of the model, similar to how it is done in MSA.

Recurrent methods

Let us describe the *first* of the two mentioned recurrent methods for estimating parameters and selection criteria calculation. Basically, the sequence of calculations coincides with the non-recurrent method, except for the calculation of the values $a_{A,l_{r+1}}$, $c_{A,l_{r+1}}$, $AR_{B,l_{r+1}}$, $RSS_{A,l_{r+1}}$. In this method $a_{A,l_{r+1}}$, $c_{A,l_{r+1}}$ are calculated according to the formulas:

$$a_{A,l_{r+1}} = a_{A,l_r}^* (\hat{\omega}_{A,r+1,l_{r+1}}^*)^2 + 2 \cdot \hat{\omega}_{A,r+1,l_{r+1}} \hat{\omega}_{A,r+1,l_{r+1}}^{r+1} b_{A,l_{r+1}} + (\tilde{\mathbf{x}}_{A,l_{r+1}}^T \tilde{\mathbf{x}}_{A,l_{r+1}}) (\hat{\omega}_{A,r+1,l_{r+1}}^{r+1})^2, \quad (3.19)$$

$$c_{A,l_{r+1}} = (\tilde{\mathbf{y}}_{A,l_r}^T \tilde{\mathbf{y}}_A) \hat{\omega}_{A,r+1,l_{r+1}} + (\tilde{\mathbf{x}}_{A,l_r}^T \tilde{\mathbf{y}}_A) \hat{\omega}_{A,r+1,l_{r+1}}^{r+1} = c_{A,l_r}^* \hat{\omega}_{A,r+1,l_{r+1}} + b_{A,l_{r+1}} \hat{\omega}_{A,r+1,l_{r+1}}^{r+1}, \quad (3.20)$$

and $AR_{B,l_{r+1}}$, $RSS_{A,l_{r+1}}$ according to the formulas:

$$\begin{aligned} AR_{B,l_{r+1}} &= (\hat{\omega}_{A,r+1,l_{r+1}}^*)^2 AR_{B,l_r}^* + (1 - (\hat{\omega}_{A,r+1,l_{r+1}}^*)^2) (\tilde{\mathbf{y}}_B^T \tilde{\mathbf{y}}_B) - 2(1 - \hat{\omega}_{A,r+1,l_{r+1}}) (\hat{\Omega}_{A,r+1,l_{r+1}}^r)^T G_{B,l_r}^T \tilde{\mathbf{y}}_B - \\ &- 2 \cdot \hat{\omega}_{A,r+1,l_{r+1}}^{r+1} (\tilde{\mathbf{y}}_B^T \tilde{\mathbf{x}}_{B,l_{r+1}}) + 2 \cdot \hat{\omega}_{A,r+1,l_{r+1}}^{r+1} (\hat{\Omega}_{A,r+1,l_{r+1}}^r)^T \tilde{\mathbf{G}}_{B,l_r}^T \tilde{\mathbf{x}}_{B,l_{r+1}} + (\hat{\omega}_{A,r+1,l_{r+1}}^{r+1})^2 (\tilde{\mathbf{x}}_{B,l_{r+1}}^T \tilde{\mathbf{x}}_{B,l_{r+1}}), \\ RSS_{A,l_{r+1}} &= (\hat{\omega}_{A,r+1,l_{r+1}}^*)^2 RSS_{A,l_r}^* + (1 - (\hat{\omega}_{A,r+1,l_{r+1}}^*)^2) (\tilde{\mathbf{y}}_A^T \tilde{\mathbf{y}}_A) - 2(1 - \hat{\omega}_{A,r+1,l_{r+1}}) (\hat{\Omega}_{A,r+1,l_{r+1}}^r)^T G_{A,l_r}^T \tilde{\mathbf{y}}_A - \\ &- 2 \cdot \hat{\omega}_{A,r+1,l_{r+1}}^{r+1} (\tilde{\mathbf{y}}_A^T \tilde{\mathbf{x}}_{A,l_{r+1}}) + 2 \cdot \hat{\omega}_{A,r+1,l_{r+1}}^{r+1} (\hat{\Omega}_{A,r+1,l_{r+1}}^r)^T \tilde{\mathbf{G}}_{A,l_r}^T \tilde{\mathbf{x}}_{A,l_{r+1}} + (\hat{\omega}_{A,r+1,l_{r+1}}^{r+1})^2 (\tilde{\mathbf{x}}_{A,l_{r+1}}^T \tilde{\mathbf{x}}_{A,l_{r+1}}). \end{aligned}$$

In the second iteration of the GRIA using this method, it is necessary to calculate the initial values of the quantities a_{A,l_1} , c_{A,l_1} , $\forall l_1 \in I$ using the formulas

$$a_{A,l_1} = (\tilde{\mathbf{y}}_{A,l_1}^T \tilde{\mathbf{y}}_{A,l_1}) = (\tilde{\mathbf{x}}_{A,l_1}^T \tilde{\mathbf{x}}_{A,l_1}) (\hat{\omega}_{A,1,l_1}^1)^2, \quad (3.21)$$

$$c_{A,l_1} = (\tilde{\mathbf{y}}_{A,l_1}^T, \tilde{\mathbf{y}}_A) = (\tilde{\mathbf{y}}_A^T \tilde{\mathbf{x}}_{A,l_1}) \hat{\omega}_{A,1,l_1}^1. \quad (3.22)$$

In the *second* recurrent method, $AR_{B,l_{r+1}}$ the $RSS_{A,l_{r+1}}$ calculation of values and is carried out using scalars $a_{A,l_{r+1}}$, $c_{A,l_{r+1}}$, $a_{B,l_{r+1}}$, $c_{B,l_{r+1}}$ as follows:

$$AR_{B,l_{r+1}} = \tilde{\mathbf{y}}_B^T \tilde{\mathbf{y}}_B - 2c_{B,l_{r+1}} + a_{B,l_{r+1}}, \quad RSS_{A,l_{r+1}} = \tilde{\mathbf{y}}_A^T \tilde{\mathbf{y}}_A - 2c_{A,l_{r+1}} + a_{A,l_{r+1}}.$$

However, to do this it is necessary to calculate the formulas (3.18)-(3.22) on the testing sample B .

3.2.5 Computational complexity of GRIA

The computational complexity of the GRIA for various methods of parameter estimation and regularity criterion calculation is presented in Table 3.1. The calculation of the number of elementary operations was carried out for the variant of construction linear models, with freedom of choice $F = 1$ and the number of iterations $R = m$.

In the example below is shown that the generalized relaxation iterative algorithm of GMDH has linear complexity in estimating parameters and calculating the model selection criterion and the operating time of this procedure does not depend on the number of observations.

In Chapter 1, it was noted that among the iterative GMDH algorithms, convergence was proven only for the CML and GRIA algorithms. Recall that we were discussing the convergence of the solution obtained by the algorithm to the solution

$$y = \sum_{i=1}^m \hat{\theta}_i x_i \quad (3.23)$$

in case of construction a linear model, where estimates of the parameters θ_i are obtained from LSM. In the case of convergence to a nonlinear model with respect to input variables, the solution can be represented in the form of (3.23) by simple redesignation of the variables.

Table 3.1 Computational complexity of GRIA when using different methods of model construction:

RIA NC – GRIA with a non-recurrent method of model construction,

RIA RM1, RIA RM2 – GRIA with recurrent methods of model construction.

Stages	Stages	RIA NC	RIA RM1	RIA RM2
Preparatory		$n_w(0.5m^2 + 1.5m + 1)$		
Iteration stage (for one model)	Iteration 1	$5m$		
	Iterations 2 to m	$\sum_{r=1}^{m-1} m(r^2 + 4r + 9)$	$3 + \sum_{r=1}^{m-1} m(4r + 29)$	$6 + \sum_{r=1}^{m-1} m(3r + 26)$

The number of operations required to construct one model on the r -th iteration in the GRIA is shown in Table 3.2.

Table 3.2 – Number of operations for calculating of one model in GRIA

RIA NC	RIA RM1	RIA RM2
$r^2 + 4r + 9$	$4r + 29$	$3r + 26$

The convergence of iterative algorithms is an important property, and is necessary in theoretical and practical problems when it is necessary to discover a "true" model. The next section contains a proof of the so-called "internal" convergence of RIA with a complete sorting of partial descriptions – one of the varieties of the generalized relaxation iterative algorithm.

3.3 Proof of GRIA convergence

The proof answers three questions: “In what cases does the algorithm identify the “true” (unbiased) model?”; “When is there convergence to the solution, i.e. the exit vector?”; “Under what condition does the algorithm converge to some fixed point?”. All these types of convergence, as well as “intrinsic” convergence, which is convergence according to the *RSS* criterion, were discussed in subsection 1.4.

As is well known, using LSM to estimate the coefficients of model (3.23) is not always possible. For example, for $n_A < m$, we have an ill-posed parameter estimation problem. This case was not considered in the proof of convergence of the CML algorithm [14], i.e., it was assumed that $n_A \geq m$. Below is a proof of convergence of the GRIA variant, the generator of model structures of which performs a complete sorting (CS) on each iteration – RIA CS. The proof is based on a geometric interpretation of the process of its convergence, and is more illustrative.

Let be given a matrix $\mathbf{X}$ of true arguments, and a vector $\mathbf{y}$ calculated by a formula of the form:

$$\mathbf{y} = \sum_{i=1}^m \theta_i \mathbf{x}_i, \quad (3.24)$$

where $\boldsymbol{\Theta} = (\theta_1, \theta_2, \dots, \theta_m)^T$ is an unknown parameter vector, $\dim \mathbf{y} = n_A \times 1$, $\dim \mathbf{X} = n_A \times m$. Since both the model parameters and the *RSS* criterion are calculated on sample *A*, we will omit this index further.

Claim 1. RIA CS has internal convergence:

- a) to solution $\mathbf{y}$ expressed by formula (3.24), assuming that there is no noise in the data, and $n \geq m$;
- b) to the fixed point $\mathbf{y}^*$, which is the projection of the vector $\mathbf{y}$ on the hyperplane stretched over the vector-columns of the matrix $\mathbf{X}$, $n \geq m$. In this case is assumed an incomplete set of input variables in the matrix $\mathbf{X}$, or what is essentially the same, the presence of additive uncorrelated (or weakly correlated) noise at the output.
- c) to some fixed point $\mathbf{y}^{pr}$ for $n < m$.

Proof. We divide the proof into three cases:

1) $n = m$; 2) $n > m$; 3) $n < m$. Without loss of generality, we prove a), b), c) for the variant of constructing models linear in the input variables.

At the first iteration of the algorithm, the following model is constructed:

$$\hat{\mathbf{y}}_1 = \hat{\omega}_1 \mathbf{x}_1,$$

where $\mathbf{x}$ is the column-vector of the matrix $\mathbf{X}$. At iterations, the $r = \overline{2, m}$ algorithm constructs a model of the form:

$$\hat{\mathbf{y}}_{r+1} = \hat{\omega}_{r+1} \hat{\mathbf{y}}_r + \hat{\omega}_{r+1} \mathbf{x}_{r+1}, \quad (3.25)$$

where $\hat{\omega}_{r+1}$, $\hat{\omega}_{r+1}$ are estimates of model parameters found by LSM, r is the iteration number.

Let's prove a). Let us divide the proof into two cases: $n = m$ and $n > m$.

Let $n = m = 3$. In this case, using LSM to obtain estimates $\hat{\theta}_1, \hat{\theta}_2, \hat{\theta}_3$ of model

$$\mathbf{y} = \theta_1 \mathbf{x}_1 + \theta_2 \mathbf{x}_2 + \theta_3 \mathbf{x}_3 \quad (3.26)$$

guarantees zero error (residual): $\varepsilon_3 = |\mathbf{y} - \mathbf{X}_3 \hat{\boldsymbol{\theta}}_3|$. Since the RIA CS estimates of the coefficients of the first two iterations coincide with the estimates of the LSM, they guarantee a solution with

a minimum residual $|\overrightarrow{OB}|$ when projecting $\mathbf{y}$ into the plane of vectors $\mathbf{x}_1, \mathbf{x}_2$. The solution $\hat{\mathbf{y}}_2$ is the projection of the vector $\mathbf{y}$ into the plane of these vectors (Fig. 3.7).

The coefficients in the next iteration are calculated by projecting the vector $\mathbf{y}$ into the plane of vectors $\hat{\mathbf{y}}_2, \mathbf{x}_3$. In general, the residual $\overrightarrow{DE} = \hat{\mathbf{y}}_3$ is $|\overrightarrow{EB}|$ not equal to zero and does not coincide with the residual obtained when estimating the coefficients of model (3.26) using LSM.

Consider the case of $(r + 1)$ -th iteration, Figure 3.8. Since the vector $\mathbf{y}$ is expressed in terms of a linear combination of variables x_i , it lies in a hyperplane stretched over these vectors.

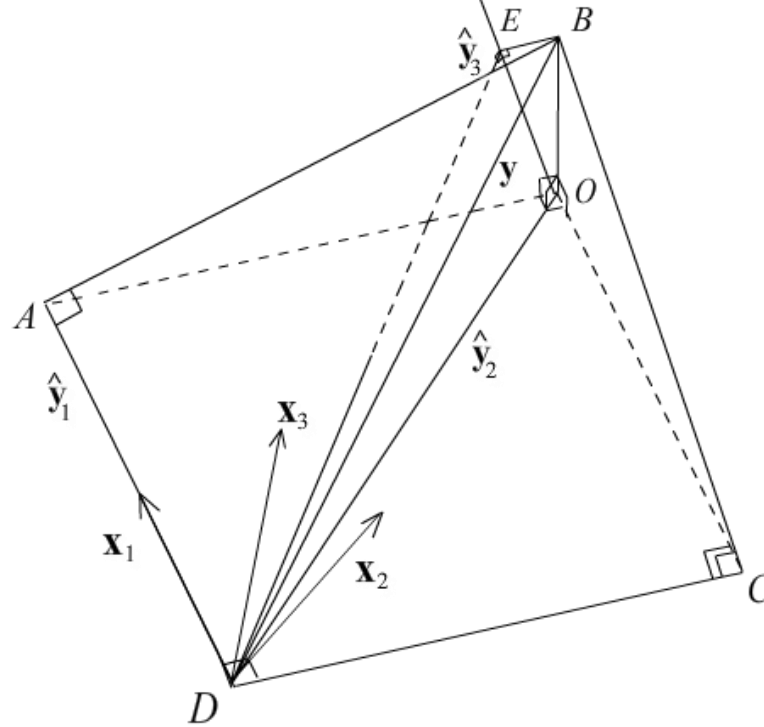


Figure 3.7 – The process of construction a solution for $r = 3$

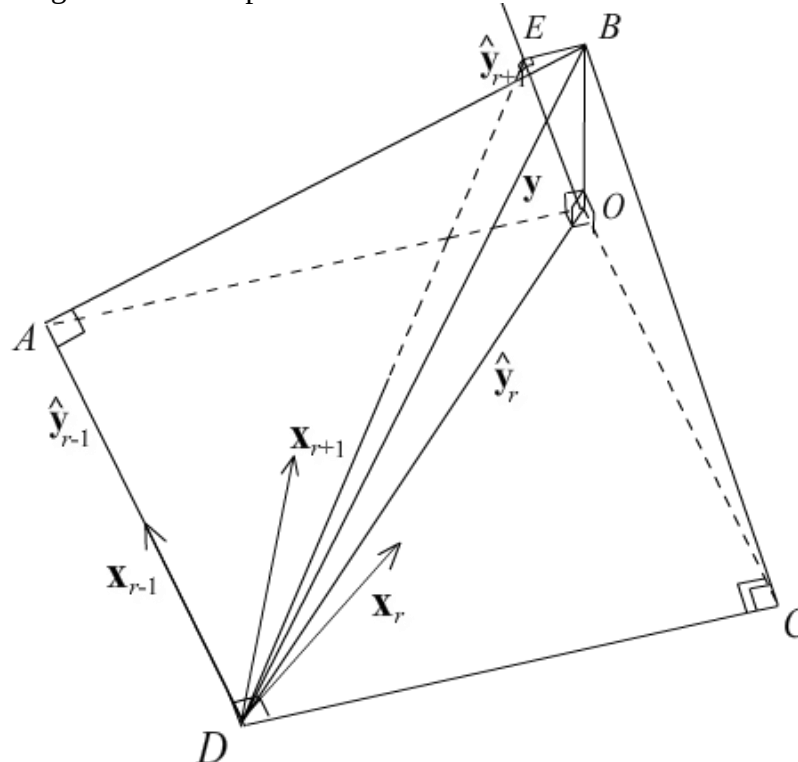


Figure 3.8 – The process of construction a solution for the $(r+1)$ -th iteration

The algorithm constructs a model in accordance with formula (3.25). At each iteration, vectors from the matrix $\mathbf{X}$ are added to the model, gradually decreasing the residual between the vector $\mathbf{y}$ and the current solution $\hat{\mathbf{y}}_{r+1}$, which can be obtained in the hyperplane spanned by the column-vectors of this matrix. This happens as follows. Obviously, the residual of the r -th iteration is $|\overrightarrow{OB}| = |\mathbf{y} - \hat{\mathbf{y}}_r|$. The residual vector of the $(r+1)$ iteration is $|\overrightarrow{EB}| = |\mathbf{y} - \hat{\mathbf{y}}_{r+1}|$. Since in a right triangle $\triangle OEB$, EB is the leg and OB is the hypotenuse, the distance between the vectors $\hat{\mathbf{y}}_{r+1}$ and $\mathbf{y}$ decreases with each iteration. Hence, for the number of iterations $r \rightarrow \infty$, the solution vector approaches the output vector. Since $|\mathbf{y} - \hat{\mathbf{y}}_{r+1}| < |\mathbf{y} - \hat{\mathbf{y}}_r|$, $r = \overline{1, m}$, the algorithm converges monotonically to the solution of the LSM, and the leg $|\overrightarrow{EB}|$, $r = \overline{3, m}$ is a measure of the minimum residual for the model (3.25) at any $n \geq m$. For $n > m$, $m > 2$ is the value $0 < |\overrightarrow{EB}| < \varepsilon_r$.

Let us consider the case $n > m$. Let $\mathbf{W} = (\mathbf{X} | \mathbf{y})$, $\dim \mathbf{W} = m \times m$. When a new observation (row-vector) is added to the matrix $\mathbf{W}$, its column-vectors still remain in the hyperplane of dimension m . Then RIA CS converges to the solution $\mathbf{y}$, if we apply the course of proof of the case $n = m$.

It should be noted that in this case the algorithm has convergence "on structure", i.e. the structure and parameters of the model obtained by the algorithm coincide with equation (3.24).

Let's prove b). In this case, some of the variables included in the $\mathbf{y}$ dependence are absent in the matrix $\mathbf{X}$. Therefore, it is impossible to describe it accurately using the available input variables. In this case, the LSM finds the optimal solution $\hat{\mathbf{y}}^*$ with a minimum distance to vector $\mathbf{y}$. This solution is the projection of the n -dimensional vector $\mathbf{y}$ into the hyperplane defined by m arguments, similar to how the 3-dimensional vector $\mathbf{y}$ is projected into the space of two variables x_1 and x_2 in Figure 3.7. With each iteration, the RIA CS approaches $\mathbf{y}$, adding only the column-vectors of the matrix $\mathbf{X}$ to the model. Consequently, the algorithm converges to the best solution that can be obtained in a given hyperplane, namely the LSM solution - to the projection $\hat{\mathbf{y}}^*$.

Let's prove c). Let $n < m$, i.e. the matrix has incomplete rank, because the system of conditional equations is underdefined. The determinant of the matrix of normal equations for such a system is equal to zero. Let us consider an example where the true model $\mathbf{y}$ is described by three vectors by (3.26). However, in fact we have the data $\mathbf{x}^{pr}$, $\mathbf{y}^{pr}$, $\dim \mathbf{X}^{pr} = 3 \times 2$, $\dim \mathbf{y}^{pr} = 3 \times 1$, Figure 3.9. That is, moving from the matrix $\mathbf{X}$, $\dim \mathbf{X} = 3 \times 3$, and the vector $\dim \mathbf{y} = 3 \times 1$, to the matrix $\mathbf{X}^{pr}$ and the vector $\mathbf{y}^{pr}$, we project the vectors $\mathbf{y}$, $\mathbf{x}_i$, $i = 1, 2, 3$ into a two-dimensional space, which is defined by the two coordinates $O1$ and $O2$ remaining after the "discarding" of the observation by the coordinate $O3$.

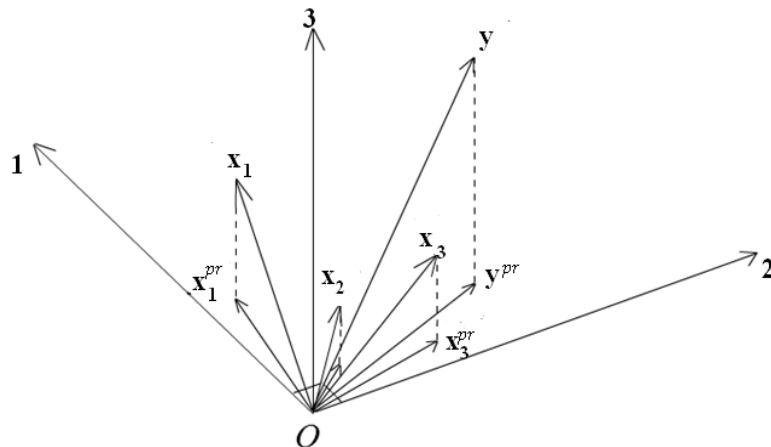


Figure 3.9 – Geometric interpretation of the process of "discarding" observations

In the general case, having $n < m$, we proceed to the projection of all vectors of the matrix $\mathbf{W}$ from an m -dimensional space into a space of dimension n . Thus, the output of the algorithm is not a vector $\mathbf{y}$ of dimension m , but its projection $\mathbf{y}^{pr}$ into n -dimensional space. Therefore, RIA CS converges not to the solution $\mathbf{y}$, but to the limit point $\mathbf{y}^{pr}$, applying the proof for case a) when $n = m$.

3.4 Comparison of GRIA and MSA

Comparison of the performance of these two GMDH algorithms concerns both the theoretical aspect, based on estimates of their computational complexity, and the practical aspect, by comparing their operating speed on the original $\mathbf{X}$ matrices of different dimensions using computer experiments.

3.4.1 Theoretical comparative analysis of the speed of GRIA and MSA

This section describes theoretical estimates of the relative speed of generalized relaxation iterative algorithm (GRIA) in relation to multi-layered simplified algorithm (MSA) for problems of varying complexity.

The complexity of problems depends on the values of parameters m and $n_W = (n_A + n_B)$. In this case, let's take the following conditional gradation:

- small problems – $m \in (1; 25]$;
- medium problems – $m \in (25; 250]$;
- large problems – $m \in (250; 2500]$;
- very large problems – $m \in (2500; 10000]$;
- super-large problems – $m \in (10000; m_{\max}]$.

As a rule, sorting-out algorithms are intended for solving small problems, i.e. $n_W \in (m + 1; n_{\max}]$. Iterative algorithms have an important advantage, since they can construct models with a number of points $n_W \in (3; n_{\max}]$. In practice, the numbers $n_{\max}$ and $m_{\max}$ are limited by the size of RAM for accommodating the initial matrix $\mathbf{W}$ of dimension $n_{\max} \times m_{\max}$ and the matrix $\tilde{\mathbf{W}}^T \tilde{\mathbf{W}}$ of dimension $m_{\max} \times m_{\max}$.

On a computer with 1Gb of RAM, you can place matrices, for example, of the following sizes: $n_{\max} = 500000$, $m_{\max} = 100$, or if $n_{\max} = 3000$, then $m_{\max} = 10000$. At the same time, data storage and calculations are carried out with double precision. To calculate the relative speed of GRIA for the n_W parameter, the following half-intervals were selected: (25; 250], (250; 2500], (2500; 10000], (10000; 50000], (50000; 500000].

The relative rate OS_i , $i = \overline{1,3}$ of each of the GRIA variants is determined by the ratio of its number of Q_i operations to the number of Q operations of the multi-layered simplified algorithm:

$$OS_i = Q_i / Q, \quad i = \overline{1,3}$$

at the same values of n_A , n_B , m . The number Q is calculated according to the formula (3.11), and the numbers Q_i are the sums of the corresponding values in Table 3.1. When calculating the value of Q , the ratio was adopted $n_A:n_B = 2:1$, which is used in practice. Calculation of the values Q , Q_i , $i = 1, 2, 3$ was performed for the average values from the half-intervals of the parameters m , n_W . The limit value was assumed to be $m_{\max} = 50000$.

Table 3.3 shows the results of calculating the reciprocals of OS_i to show how many times the GRIA is faster than the MSA. As we can see from Table 3.3, the speed of the GRIA is significantly higher than the speed of the MSA.

Table 3.3 – Relative speed of GRIA

	n_W	RIA CS	RIA RM1	RIA RM2
Tasks of the medium dimensions, $m = 140$	140	0,58	10,5	13
	1375	5	39	42
	6250	17,6	51	52
	30000	38	55	55
	275000	53	56	56
The tasks are large dimensions, $m = 1375$	1375	0,5	93	116
	6250	2,3	248	280
	30000	11	395	411
	275000	83	458	461
Problems of very large dimensions, $m = 6250$	6250	0,5	418	522
	30000	2,4	1140	1287
	275000	22	1918	1959
Tasks of the super-large dimensions, $m = 30000$	30000	0,5	2001	2501
	275000	4,6	6968	7540

3.4.2 Comparison of the performance of GRIA and MSA software implementations

The numerical experiment compares the performance of the GRIA variant, in which the model structure generator performs a truncated sorting, - let's denote it as MSA^* (which is an analogue of MSA). To show that the computational complexity of the stage of estimating coefficients and calculating the selection criterion in GRIA does not depend on the number of observations, we highlight this stage and denote it by MSA^- .

The freedom of choice of models in both algorithms was set to $F = 1$. The result of the comparison on a logarithmic scale is presented in Figure 3.11. As we can see, with the number of arguments $m = 100$, MSA^* works 100 times faster than its prototype; at $m = 1000$ – a thousand times faster. When the number of arguments reaches $m = 10000$, then the algorithm works as many times faster as the number of observations - proportional to the number of observations: if $n_W = 10$ - then 10 times faster, if $n_W = 100$ - 100 times faster, if $n_W = 1000$ - 1000 times faster, etc. The graphs also show that the computational complexity of MSA^* at the model construction stage (graph MSA^-) does not depend on the number of observations. At $m = 10000$, the graphs for MSA^- and MSA^* coincide.

The freedom of choice of models in both algorithms was set to $F = 1$. The result of the comparison in a logarithmic scale is shown in Figure 3.11. As you can see, with the number of arguments $m = 100$, the MSA^* works 100 times faster than its prototype; with $m = 1000$ – a thousand times faster. When the number of arguments reaches $m = 10000$, then the algorithm works as many times faster as the number of observations – in proportion to the number of observations: when $n_W = 10$ – then 10 times faster, when $n_W = 100$ – 100 times, when $n_W = 1000$ – 1000 times, etc. The graphs also show that the computational complexity of MSA^* at the model construction stage

(graph MSA^-) does not depend on the number of observations. When $m = 10000$, the graphs for MSA^- and MSA^* coincide.

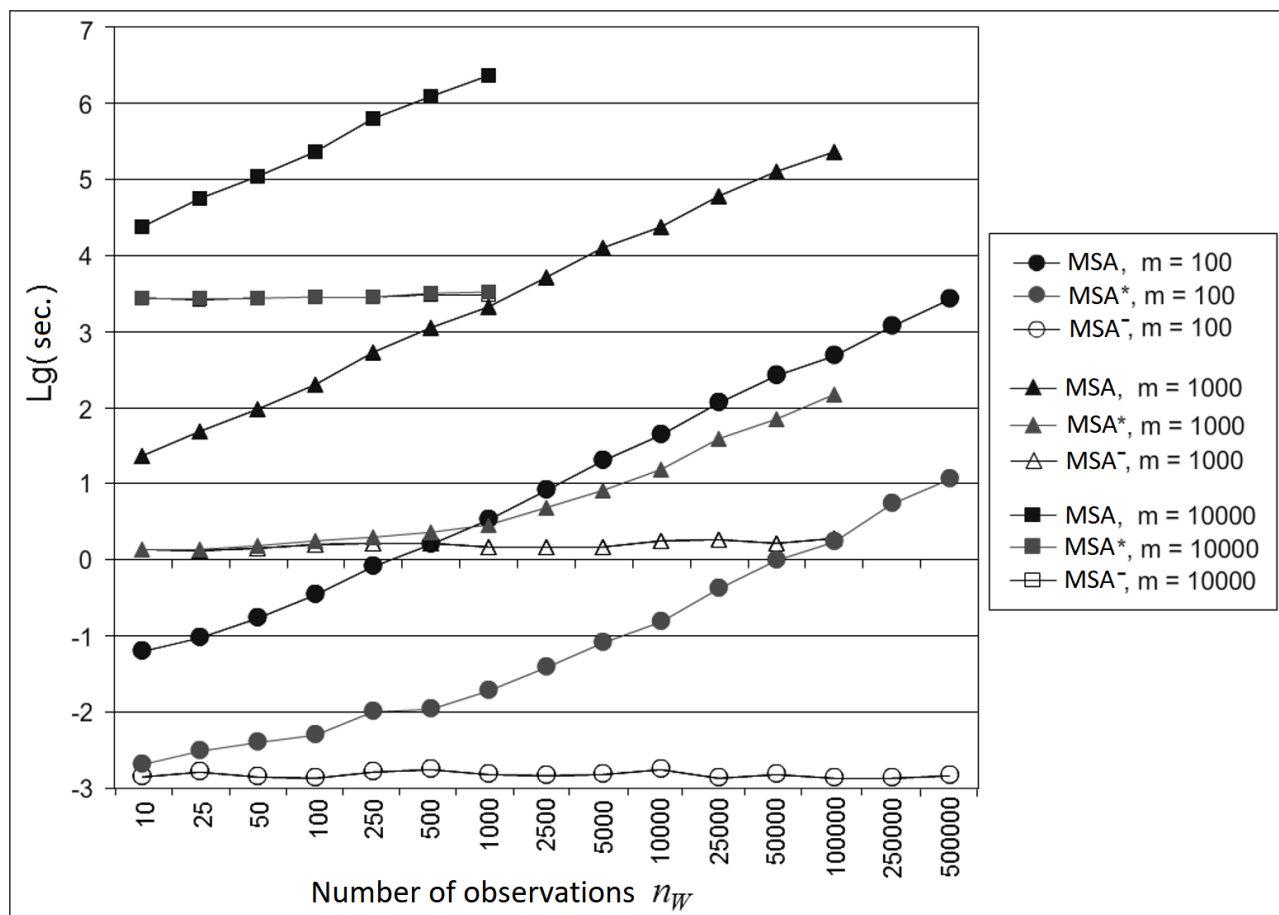


Figure 3.11 – Comparison of the performance of the MSA^* and MSA algorithms

In the range from milliseconds to tens of minutes, there is a linear dependence of the operating time of both algorithms on the number of observations n_W for the number of arguments $m = 100$ and $m = 1000$.

3.5 Discussion

As we can see, the generalized relaxation iterative algorithm has linear complexity in estimating parameters and calculating the model selection criterion, and the operating time of this procedure does not depend on the number of observations, which allows this algorithm to be used to solve practical problems, the data table of which has hundreds of thousands of observations and thousands of variables.

Based on the results of comparing the GRIA with its prototype – a multi-layered simplified algorithm, the advantages of the first one are obvious:

- 1) increase in performance by several orders of magnitude in proportion to the number of observations in the original sample;
- 2) increasing the speed of the algorithm, at least by an order of magnitude, in comparison with any other iterative algorithm of GMDH due to recurrent methods for estimating parameters and calculating the selection criterion;
- 3) the possibility of convergence to a nonlinear model due to the generator of model structures, which performs a complete sorting of multiplicative arguments at each iteration.

It should be noted that this "breakthrough" in the speed of GRIA became possible due to a more complete use of the RAM of a modern computer. When in 1968 A. G. Ivakhnenko proposed the first multi-layered algorithm GMDH, it was designed to solve problems with a small number of observations and could construct quite complex models. Computers at that time (70-80s) had a very limited volume of memory, so the principle of designing algorithms was following: "we store less - we calculate more."

For a number of reasons, interest in improving these algorithms was subsequently lost, so known implementations of iterative algorithms did not fully use the RAM of modern personal computers, most of which remained free. When developing GRIA, the authors proceeded from the opposite principle: "the more we store, the less we calculate". GRIA fully uses RAM of modern computers, makes it possible to solve high-dimensional modeling problems, and also makes it possible to increase the speed of known iterative algorithms by several orders of magnitude.

Regarding the proof of the convergence of GRIA, it should be noted that it does not apply to the entire generalized algorithm, but only to those varieties of it in which a complete sorting of partial descriptions is carried out at each iteration. It is this fact that allows algorithms to have this property. If you use a truncated sorting of partial descriptions, you can always find a counter example when the algorithm will not converge to an exact solution.

Attention should be paid to the case of proving the convergence of GRIA, when the number of observations is less than the number of arguments. It was shown that the GRIA converges to a fixed point – the projection of the output vector on the hyperplane, specified by the column-vectors of the initial matrix. In the chapter 4, in numerical experiments we will show that depending on the values of the GRIA parameters – the initial conditions for the model construction – for example, the freedom of choice F , the limit point can change. This fact is consistent with the result of solution an underdetermined system of normal equations that has many solutions.

The next chapter pays attention to obtaining statistically substantiated results of numerical computer experiments in the study of iterative algorithms using the example of the iterative relaxation algorithm of GMDH.

Chapter 4

Investigation of the Properties of Iterative Algorithms by Computer Experiments

While the speed of problem solving is undoubtedly one of the fundamental properties of an algorithm, it is also important to know: how accurate is the solution obtained by an iterative algorithm? This question is closely related to the issues of algorithm convergence and convergence rate.

Furthermore, identifying properties and the property of noise-immunity of modeling, are important in GMDH theory. The first of them is related to the problem of structural-parametric identification, the goal of which is to find an exact solution ("true" or "adequate" model) in the presence of false or dummy column-vectors in the initial matrix $\mathbf{X}$. Dummy input variables are those variables that are not present in the regression equation for the output variable $\mathbf{y}$. The latter relates to the algorithm's ability to construct models that automatically adapt to an unknown noise level, simplifying the model as the noise level increases.

Thus, this chapter is devoted to the study of three properties of iterative GMDH algorithms that are important for practice using computer experiments: convergence rate, ability to find the "true" model, and noise-immunity. We will show the study of all these properties using the example of the GRIA described in the previous chapter. The goal of the study is to demonstrate the features of the GRIA, to show validity and reliability of its testing results, and draw attention to a methodology for studying iterative algorithms, that allows conducting statistically significant computer experiments.

4.1 Investigation of the Convergence Property of Algorithms

As noted, one of the important properties of any iterative algorithm is its convergence to an exact solution. In the context of GMDH, the question arises: "What is the exact solution?" The author of first iterative algorithm approximated the polynomial

$$f(x_1, \dots, x_m, \Theta) = \theta_0 + \sum_{i=1}^m \theta_i x_i + \sum_{i=1}^m \sum_{j=i}^m \theta_{i,j} x_i x_j + \sum_{i=1}^m \sum_{j=i}^m \sum_{k=i+j}^m \theta_{i,j,k} x_i x_j x_k + \dots \quad (4.1)$$

by superposition of polynomials with a smaller number of variables. Later, an algorithm for a complete combinatorial sorting of the polynomial terms (4.1) – the COMBI algorithm appeared. Its goal is to find the global optimum of criterion in this class of models. Assuming that the true model of an object is described by the polynomial (4.1) with a limited number of terms (up to 30), the COMBI algorithm solves this problem. Given this rather "strong" assumption, as well as the absence of noise in the data, the true model is the exact solution corresponding to the global extremum of the accuracy criterion.

Since the model obtained by sorting on a set of 30 arguments can be not complex enough to solve modern practical problems, iterative algorithms were improved to obtain a solution. The goal of developing iterative algorithms has always been to find a solution that theoretically is obtained by algorithms of complete sorting, but in practice can be obtained under certain constraints only.

As it turns out, not all iterative algorithms, but only some, in particular two relaxation algorithms – CML and RIA CS, have the property of convergence to the "true" model. The rate of convergence of an iterative algorithm is its most important indicator. This chapter discusses the following: "What

factors influence the rate of convergence?" and "How is this influence manifest itself?" However, they were obtained for GRIA through computer experiments, i.e. in practice.

To ensure that such estimates are objective and substantiated, this chapter presents a methodology for conducting computer experiments for studying iterative algorithms, based on statistical testing.

Let us define the concept of the "convergence rate" of an iterative algorithm.

The *convergence rate* of an iterative algorithm is the number of iterations r^* required to achieve a given solution accuracy ε . As we can see, the convergence rate is directly related to the algorithm's execution time.

What factors affect the speed of convergence of an algorithm? Since we are talking about algorithms that process numerical data, the speed depends on the properties of data, and, of course, on the values of algorithm parameters. For example, in GMDH algorithms, it depends on the freedom of choice F . Therefore, first it is necessary to determine the properties or characteristics of the initial data that affect the speed of algorithm convergence. These may be different for each algorithm: for one, it is the condition number of the correlation matrix of variables, while for another, it is the determinant of this matrix. Determining the data characteristic is beyond the scope of computer experiment methodology and requires a researcher to know this characteristic.

The results of computer experiments answer the following question: "At what values of the data characteristic that influences the algorithm's convergence rate does it converge most quickly, and at what values does it converge most slowly?" or "Which values of these properties make the algorithm converge faster, and which slower?" Knowing the answers to these questions, we can determine for a specific data sample which values of the algorithm's parameters – the number of iterations R before the algorithm stops, the freedom of choice F , etc. – should be set in order to obtain a solution with a given accuracy ε . Thus, the strategy for using a converging iterative algorithm has the form shown in Figure 4.1.

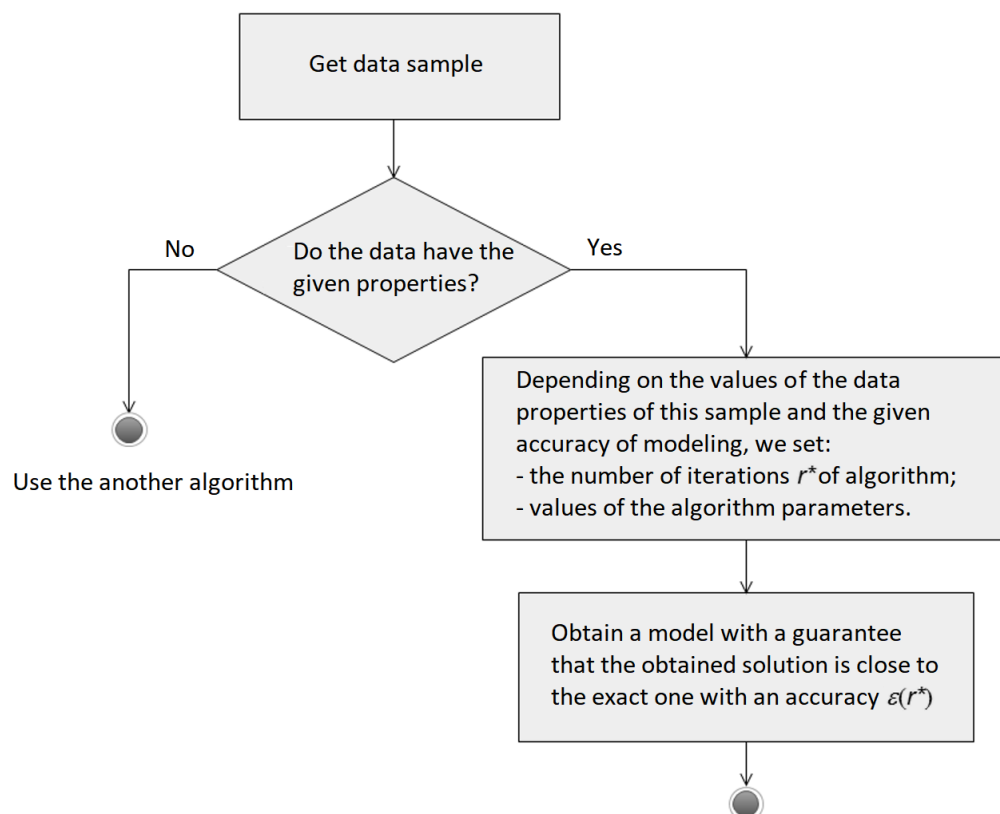


Figure 4.1 – Strategy for applying a converging iterative algorithm in practice

Let the data characteristic influencing the convergence rate be known and, without loss of generality, a single one. Let us denote it by Pr .

To obtain an independent picture of the influence of the Pr property and algorithm parameters on the algorithm's convergence rate, the methodology involves conducting experiments first with fixed values of the Pr value and variable algorithm parameters values, and then, conversely, with fixed algorithm parameters values and a varying data characteristic Pr . The second most important issue is the randomness of the Pr values, since computer experiments use a pseudorandom number generator to generate the initial matrix. Therefore, it is necessary to configure the data matrix generation procedure (MGP) so that the estimated mathematical expectation of the Pr value for all generated matrices is close to the specified value. The randomness of the initial matrices dictates the randomness of the algorithm's convergence rate r^* .

To estimate that quantities, one or two values are insufficient, so the methodology involves conducting a series of input matrix generations and constructing histograms of the r^* parameter distribution for each of the algorithm parameter values and the data characteristic Pr .

We will describe a methodology for conducting computer experiments to study the convergence rate of an iterative algorithm.

4.1.1 Methods of conducting computer experiments

To determine the required number of RN generations (number of realizations) of the input matrices, the experimental error δ is specified as a percentage. The experiment is defined as the constructing a histogram of the r^* parameter distribution for a given RN number. The error δ evaluates the closeness of r^* parameter distribution histograms obtained in two experiments, for example, for $RN = 10^3$ and $RN = 10^4$. The method for calculating the closeness of the histograms may vary.

The technique consists of three stages:

- Stage 1.* Determine the MGP parameters and their values that allow generating matrices with an estimate of the mathematical expectation of the data characteristic Pr , close to the specified value.
- Stage 2.* Study the convergence rate of the algorithm when the value of the parameter Pr varies while the values of its control parameters remain constant.
- Stage 3.* For each of the algorithm's control parameters, study the convergence rate as its value varies while the other parameters (including Pr) remain constant.

Each of the stages of the method consists of two steps:

Step 1. Determine the number of RN^* realizations, satisfying a given δ value for the distribution histograms of the parameter r^* at a given value of the parameter being studied par , $par \in \{Pr, F, \dots\}$.

- 1.1 Generate an initial matrix using MGP.
- 1.2 Build a model using the algorithm under study, feeding generated matrix to its input.
- 1.3 Add the obtained r^* value to the histogram.
- 1.4 Perform steps 1.1-1.3 the specified number of times RN .
- 1.5 Perform step 1.4 for different RN values, increasing it in some increments until the measure of the measure of difference between histograms for two consecutive RN values satisfies the given value δ . The resulting RN value is denoted RN^* .

Step 2. Plot histograms of the r^* parameter distribution for different values of the parameter par using the determined number of realizations RN^* .

At the first stage of the method $par = Pr$. In *Step 2*, it is necessary to find the values of the MGP parameters for which the estimated mathematical expectation of the parameter Pr of the generated matrix is close to the specified value.

Let us illustrate the method by studying the convergence rate of a generalized relaxation iterative algorithm (GRIA) with generator that performs a complete sorting of arguments at each iteration, and call it as RIA CS.

4.1.2 Study of the GRIA convergence rate

The first question to answer is, "What data characteristic influences its convergence rate?" It can be shown that the rate of convergence of the RIA CS depends on such a characteristic of the matrix $\mathbf{X}$ as the measure of orthogonality of its vector-columns system. Let us denote the mean-centred and standardized vector-columns of the matrix $\mathbf{X}$ as the matrix $\tilde{\mathbf{X}}$. Obviously, if the matrix $\tilde{\mathbf{X}}$ vector-columns form an orthogonal system, then the correlation matrix $\Sigma_X = \tilde{\mathbf{X}}^T \tilde{\mathbf{X}}$ is singular with a determinant equal to 1. Therefore, we choose the determinant d of the correlation matrix Σ_X as the orthogonality measure of the matrix $\tilde{\mathbf{X}}$ vector-columns system. Since the sign of the pairwise correlation between the matrix vector-columns vectors is unimportant for the orthogonality measure, let the matrix Σ_X contain the absolute values. Therefore, $d \in [0; 1]$ and if $d = 0$, then the matrix Σ_X is degenerate, and if $d = 1$ it is singular.

The convergence rate will be studied, provided that the matrix $\mathbf{X}$ contains only true arguments. The convergence of the algorithm is tracked by the difference of the $NRSS$ criterion for the best models at two adjacent iterations: $NRSS_r^* - NRSS_{r+1}^* = \Delta_{r+1}$. The algorithm stops according to the condition $\Delta_{r+1} < \varepsilon$, where ε is the specified modeling accuracy. The formula for the calculation $NRSS_r^*$ value is as follows:

$$NRSS_r^* = \sum_{i=1}^{n_A} (\tilde{y}_i - \tilde{y}_{r,i}^*)^2 / \sum_{i=1}^{n_A} \tilde{y}_i^2,$$

where $\tilde{y}_i$ is the centered value of y_i of the vector $\mathbf{y}$; $\tilde{y}_{r,i}^*$ is the corresponding centered value of output of the best model on the r -th iteration.

In the experiments below, we will construct both linear and nonlinear models for input variables x_i , corresponding to the vector-columns $\mathbf{x}_i$ of the matrix $\mathbf{X}$, $i = \overline{1, m}$. To construct nonlinear models, a matrix $\mathbf{Z}$ is preliminarily formed, containing column-vectors corresponding to all possible products of the initial variables x_i , $i = \overline{1, m}$ without repetitions, limited by a given maximum power $pow > 1$, $\dim \mathbf{Z} = n \times l$, $l = C_{m+pow}^{pow} - 1 - m$.

To describe MGP let's introduce notations: s_{lin} is the number of true linear arguments; s_{mult} is the number of true multiplicative arguments; $\mathbf{W}$ is the MGP output matrix, $\mathbf{W} = (\mathbf{H}; \mathbf{y})$, $\dim \mathbf{W} = n \times (M + 1)$; n is the number of observations; M is the number of arguments; $\mathbf{H} = (\mathbf{X} : \mathbf{Z})$. The input parameters of the MGP are: M , n , s_{lin} , s_{mult} . The flow chart of the algorithm is shown in Fig. 4.2.

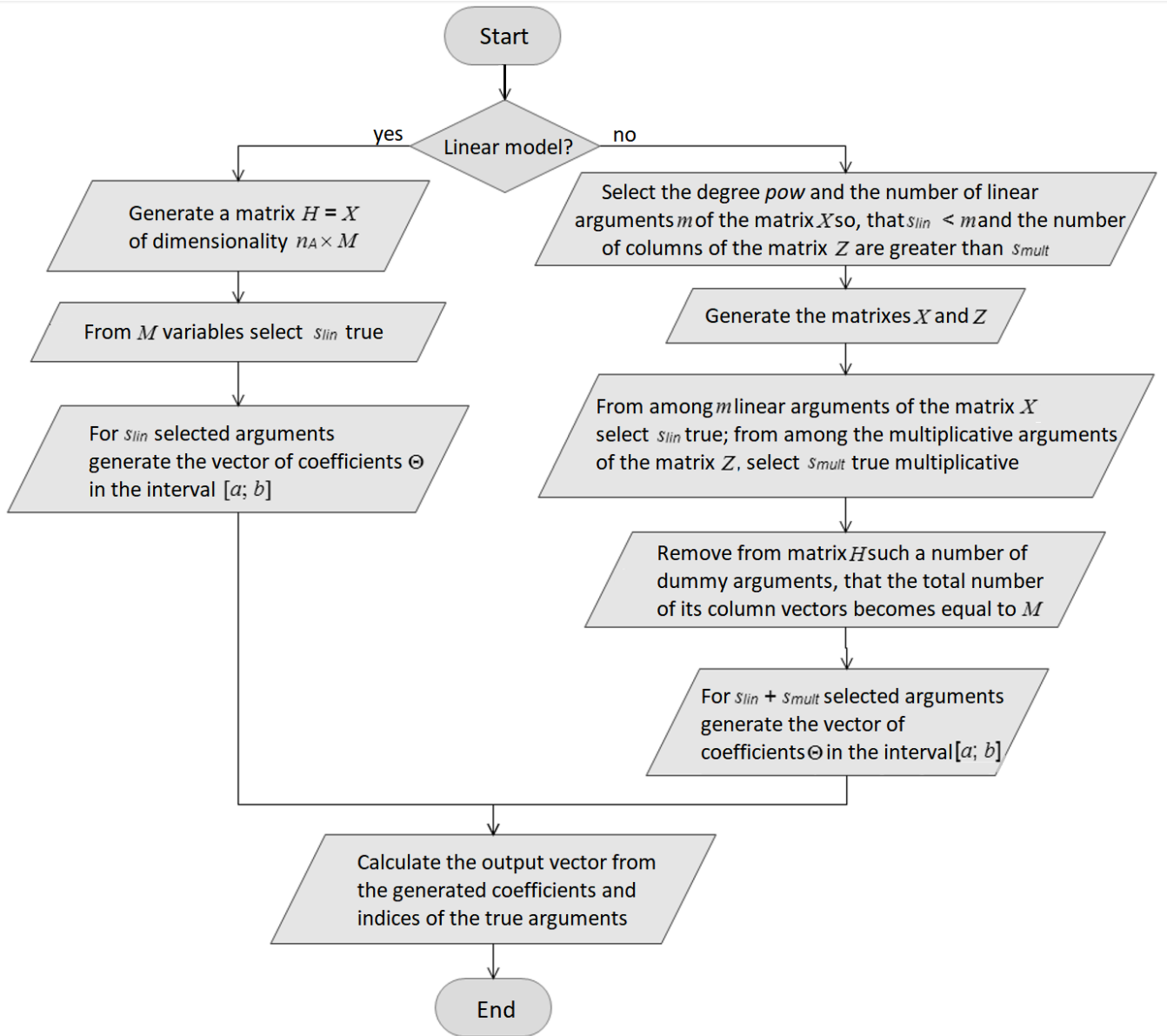


Figure 4.2 – Flow chart of the matrix generation procedure

The PGM uses one of the well-known pseudorandom number generators, Mersenne Twister MT19937 from the open C++ Boost library [97], the period of random sequence repetition of which is 2^{19937} [98]. Pseudorandom numbers are generated using a uniform distribution over the interval $[0; 1]$.

So, let's move on to the first stage of the method.

Stage 1. Determine the parameters of MGP and their values that allow generating matrices with an estimated mathematical expectation of the determinant d , close to a given value.

First of all, it is necessary to determine the number of realizations RN^* for which the results can be considered reliable.

Stage 1. Step 1. Determine the required number of RN^* realizations.

Histograms were plotted for the following series of values $RN \in \Xi$, $\Xi = \{10^3, 10^4, 10^5, 10^6\}$. The frequency (probability) shown in the figures below is the ratio of the number of realizations for which the algorithm converges to the model with a given accuracy ε within the number of iterations from the corresponding interval r^* , to the total number of implementations equal to RN .

As a measure of difference between histograms, we will use the *DM* (Disparity Measure), which is based on the ratio of frequency values, which is calculated for different *RN* values.

To calculate the DM measure of histogram differences.

Let the number of intervals of the r^* histogram parameter be *IntNum*. The *DM* measure for each pair of adjacent values from the set Ξ is calculated as follows:

1. For each interval of r^* histogram values, the lower frequency value in one histogram is divided by the larger one in the other. We obtain *IntNum* values of the accuracy $\gamma_j \in [0; 1]$, $j = \overline{1, \text{IntNum}}$.
2. Calculate the *DM* measure using the formula:

$$DM = \frac{1}{\text{IntNum}} \sum_{j=1}^{\text{IntNum}} \gamma_j .$$

After performing this procedure for all pairs, we get a set of histograms difference measures in percentages: {82.1%, 91.5%, 98%}, which can be interpreted as modeling accuracy. Histograms for the last pair are shown in Fig. 4.3.

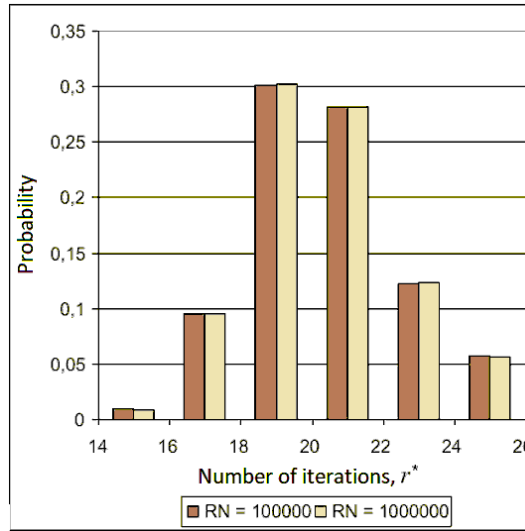


Figure 4.3 – Histograms of the r^* parameter distribution for the last pair of values in the Ξ set.

As can be seen from Figure 4.3, if the value of $RN = 106$ is taken as a "reference", the resulting error $\hat{\delta} = 2\%$ ($DM = 98\%$) satisfies the specified error value $\delta = 5\%$, then $RN^* = 105$. This value is used below, since as was established in numerical experiments for a number of studied parameter values, similar histograms were obtained at the first step of each stage of the methodology.

Stage 1. Step 2. *Determination of the parameters of MGP and their values, allowing the generation of matrices with an estimated mathematical expectation of the determinant d close to a given value.*

The process of generating two independent random column-vectors in the PGM has the following property: with greater number of elements n in the vectors, the lower values of their pairwise correlation. This is explained by the fact that with the higher dimensionality of vectors, the greater the degree of freedom they can be represented, and the greater the possibilities for mutual orthogonalization of the vectors. Therefore, the number of observations n can be used to generate a matrix $\mathbf{X}$ with a given determinant of the matrix $\Sigma_{\mathbf{X}}$, whereas n increases, the diagonal will contain values close to 1 in absolute values, and off-diagonal values will be close to zero. The rationale for this conclusion is demonstrated experimentally in *Appendix A*.

Since the function $d(n)$ is monotonically increasing, binary search over the set of values of the parameter n is used as the algorithm for finding the given values of d .

For the linear model

$$y = 6.29447 - 7.29046 \cdot x_1 + 8.11584 \cdot x_2 + 6.70017 \cdot x_3 - 7.46026 \cdot x_4 + 9.37736 \cdot x_5 \quad (4.2)$$

and nonlinear

$$y = 6.29447 - 7.29046 x_1 + 8.11584 x_2 + 6.70017 x_3 - 7.46026 x_4 + 9.37736 x_5 + 8.26752 x_1^2 - 5.57932 x_1^3 + 2.64719 x_1^2 x_2. \quad (4.3)$$

the values of the parameter n were determined, allowing to generate matrices $\mathbf{X}$ with determinant d of their correlation matrix $\Sigma_{\mathbf{X}}$ for $d \in D$, $D = \{0.5, 0.6, 0.7, 0.8, 0.9, 0.99\}$. The results are presented in Table 4.1, in the cells of which contain the values of n .

Table 4.1 – Number of points for different values of determinant

d	0.5	0.6	0.7	0.8	0.9	0.99
Linear model	12	16	25	41	94	875
Nonlinear model	10	12	16	26	57	500

Let's move on to the second stage of the method, which explore the convergence rate a as a function of the parameter d , keeping the algorithm's parameters fixed: freedom of choice F and the number of arguments in the linear and nonlinear models S_{lin} , S_{nonlin} .

Stage 2. Step 1. Investigating the algorithm's convergence rate as the determinant d varies, while the other parameters remain constant.

We investigate the algorithm's convergence rate to the linear model (4.2) as the determinant d varies, $d \in D$. As we can see from the results presented in Figure 4.4, the value of the determinant influences the algorithm's convergence rate.

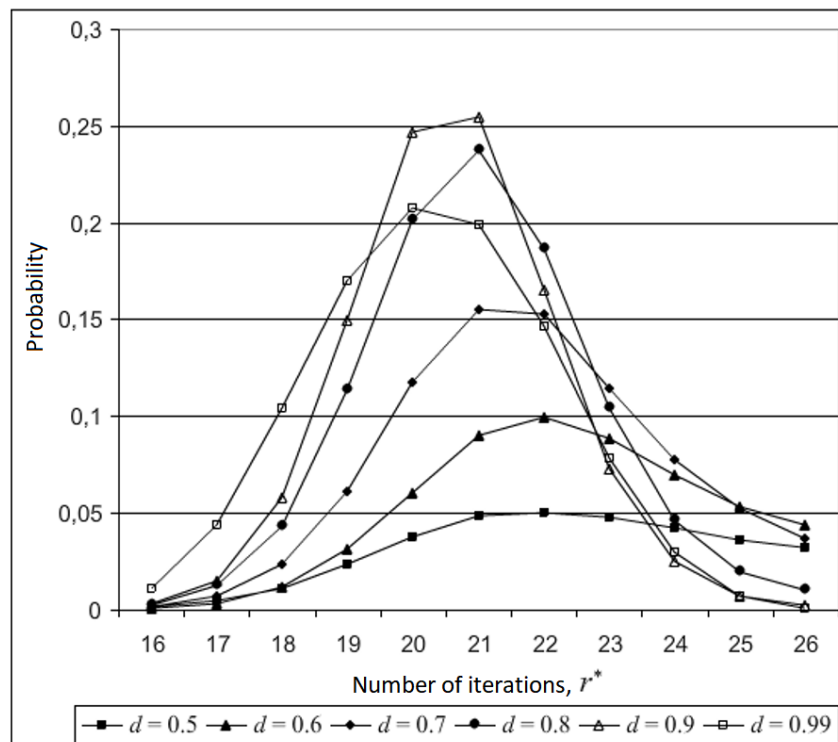


Figure 4.4 – Histograms of the r^* distribution with varying determinant d

Let us examine how the magnitude of the coefficients in the model arguments affects the rate of convergence of r^* .

Study of sensitivity of the algorithm convergence rate to changes in the model coefficients. The experiment analysed the rate of convergence to two models: model (4.2) and model

$$y = -0.600433 - 0.29681 \cdot x_1 - 0.791249 \cdot x_2 - 4.93607 \cdot x_3 + 1.51528 \cdot x_4 - 5.49378 \cdot x_5. \quad (4.4)$$

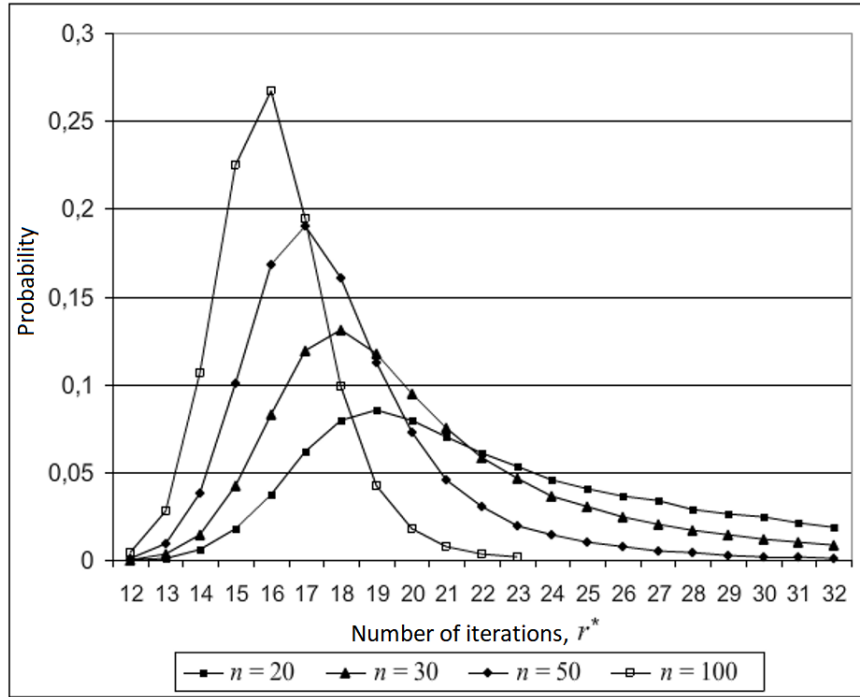


Figure 4.5 – Histograms of the r^* distribution for model (4.4)

The algorithm parameters were as follows: freedom of choice $F = 5$ and modeling accuracy was chosen to be as $\varepsilon = 10^{-12}$, based on considerations of achieving the highest accuracy possible in the study of recurrent algorithms. For $\varepsilon < 10^{-12}$ due to rounding, the value of the model criterion may turn out to be negative. The number of observations n took values from the set $n \in \{20, 30, 50, 100\}$.

During models construction, the same data samples were used, they differed only in the output column. As we can see, some coefficients in (4.4) are an order of magnitude smaller than in (4.2). The result for model (4.2) is shown above in Figure 4.4, and for model (4.4) – in Figure 4.5.

As can be seen from the figures, the rate of convergence to model (4.4) is higher than to model (4.2). This is explained by the following. Since the values of the vectors x_i , $i=1, \dots, 5$ are generated in the same interval, the contribution of each argument to the model is determined by a coefficient. In other words, the coefficient determines the extent to which the argument is present in the model (if $\theta=0$ it is absent). Since in model (4.4), unlike (4.2), the coefficients for arguments x_1 and x_2 are smaller, the algorithm reaches the desired accuracy more quickly.

If an argument is generated or measured in different intervals, then its contribution to the model depends both on the coefficient and on the size of interval of its values. If the argument values are uniformly distributed, the contribution of the i -th argument to the model can be calculated using the formula:

$$w_i = \frac{|\theta_i| (x_{i,\max} - x_{i,\min})}{\sum_{j=1}^{s_t} |\theta_j| (x_{j,\max} - x_{j,\min}) + \theta_0}, \quad i = \overline{1, s_t},$$

where θ_i is the coefficient of the i -th argument, $x_{i,max}$, $x_{i,min}$ is the maximum and minimum value of the i -th argument; s - number of arguments in the model; θ_0 - intercept. The weight of the intercept is calculated as

$$w_0 = \frac{|\theta_0|}{\sum_{j=1}^{s_t} |\theta_0| (x_{j,max} - x_{j,min}) + |\theta_0|},$$

while $\sum_{i=0}^s w_i = 1$. Obviously, if all weights are equal

$$w_i = \bar{w} = \frac{1}{s+1}, \quad i = \overline{0, s},$$

then the algorithm's convergence rate to the true model with s arguments is the slowest.

To measure the degree of deviation of the weights of w_i arguments from the mean, we introduce the following measure: $\psi = \sum_{j=0}^s |w_j - \bar{w}|$. It is directly checked that $\psi \in [0; 1]$. Thus, if with a given number of s arguments $\psi = 0$, then the rate of convergence is the slowest and the number r^* is the maximum. If $\psi = 1$ then the rate of convergence is maximum – the model contains only one argument and the number r^* is minimal. The values ψ for models (4.4) and (4.2) is equal to 0.173909 and 0.003537, respectively.

Let's expand the scope of our research and conduct experiments by also changing the number of model arguments s_{lin} , to see how the parameter d influences models of different lengths.

Investigation of the algorithm convergence rate when changing both the determinant d and the number of linear model arguments s_{lin}

The parameters took the values $d \in D$, $s_{lin} \in S_{Lin}$, $S_{Lin} = \{5, 8, 11, 14, 17\}$. The absolute values of the rounded model coefficients for $s_{lin} \in S_{Lin}$ and the measure of coefficients contribution ψ are presented in Table 4.2.

Table 4.2 – Linear models coefficients and measure ψ

S_{lin}	$ \theta_0 $	$ \theta_1 $	$ \theta_2 $	$ \theta_3 $	$ \theta_4 $	$ \theta_5 $	$ \theta_6 $	$ \theta_7 $	$ \theta_8 $	ψ
5	6.294	7.29	8.12	6.70	7.46	9.38				0.0035374
8										0.008908
11							8.26	5.58	2.65	
14										
17										
(continuation of the table to the right)										
S_{lin}	$ \theta_9 $	$ \theta_{10} $	$ \theta_{11} $	$ \theta_{12} $	$ \theta_{13} $	$ \theta_{14} $	$ \theta_{15} $	$ \theta_{16} $	$ \theta_{17} $	ψ
11	3.83	8.04	0.94							0.012718
14				4.43	6.23	0.93				0.013257
17							9.85	9.15	9.92	0.01033

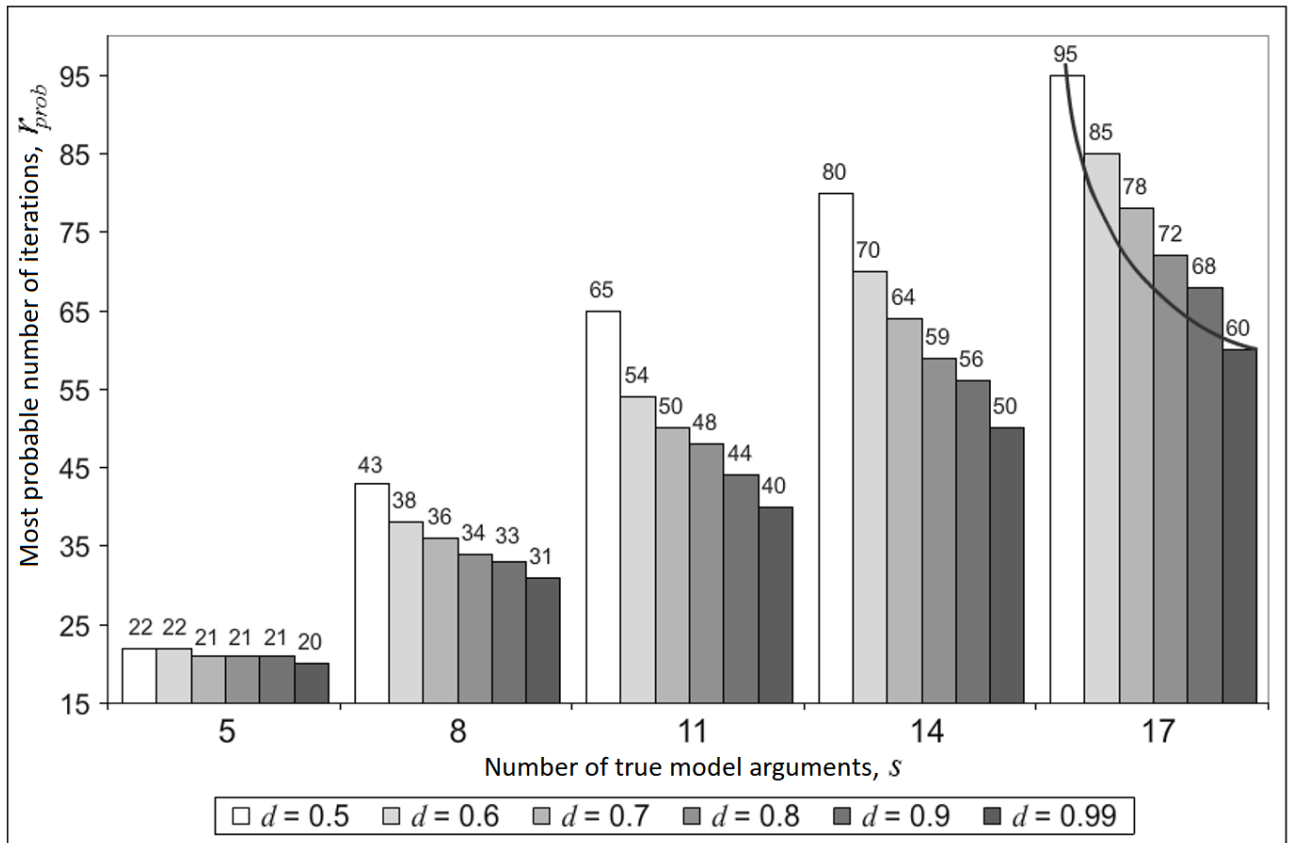


Figure 4.6 – Values of the number of iterations r_{prob} for each pair (s_{lin}, d)

Histograms were plotted for each pair (s_{lin}, d) , and for each s_{lin} value the value n was found at which the highest peak of the histogram corresponds to the specified value d . Constant are freedom of choice $F = 5$ and modeling accuracy $\varepsilon = 10^{-12}$. From results presented in Fig. 4.6 the following conclusions about convergence to the linear model can be drawn:

1. The value of parameter d affects the convergence rate in direct proportion, i.e. the larger d , the higher rate of convergence (and lower the r_{prob} value).
2. Determinant d affects the rate of convergence r^* nonlinearly for any number of model arguments (shown by the curve in the figure).
3. For an orthogonal system of column vectors in matrix $\mathbf{X}$, the algorithm requires $3s_{lin}$ iterations to obtain a solution with an accuracy of $\varepsilon = 10^{-12}$.

And now let's explore the convergence rate to a nonlinear model.

Study of the algorithm's convergence rate with varying determinant d , nonlinear model

In this case, the column-vectors $\mathbf{x}_i$ and $\mathbf{z}_j$ of the matrix $\mathbf{H} = (\mathbf{X} : \mathbf{Z})$ have a pairwise correlation lying in the range $[0.6; 0.99]$, indicating a high degree of their linear dependence. Therefore, if the determinant d_H of the correlation matrix is used as a measure of orthogonality $\Sigma_H = \tilde{\tilde{\mathbf{H}}}^T \tilde{\tilde{\mathbf{H}}}$, it will be close to zero. Most importantly, even an arbitrarily large value of the parameter n will not be able to increase the value of d_H by even 0.1. Therefore, when studying nonlinear models, the determinant d of the submatrix $\Sigma_X = \tilde{\tilde{\mathbf{X}}}_A^T \tilde{\tilde{\mathbf{X}}}_A$ of the correlation matrix Σ_H is calculated as a measure of orthogonality.

It should be emphasized that the rate of convergence of the algorithm is not affected by the formal nature of the model (linear or nonlinear), but by the degree to which the combinations of the column vectors of the matrix $\mathbf{H}$ are pairwise linearly independent. For example, if we take the model:

$$y = \theta_1 x_1^3 + \theta_2 x_2^2 + \theta_3 x_3^3, \quad (4.5)$$

which is nonlinear in arguments, and the vectors $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3$ are randomly generated (their pairwise correlation is close to zero), then the rate of convergence of the algorithm to it will be comparable to the rate of convergence to a linear model: $y = \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3$.

To study the rate of convergence of the algorithm when constructing nonlinear models, the model (4.3) was chosen, which contains both linearly "independent" variables x_1, x_2, x_3, x_4, x_5 and linearly "dependent" with $x_i, i = \overline{1,5}$, variables $x_1^2, x_1^3, x_1^2 x_2$.

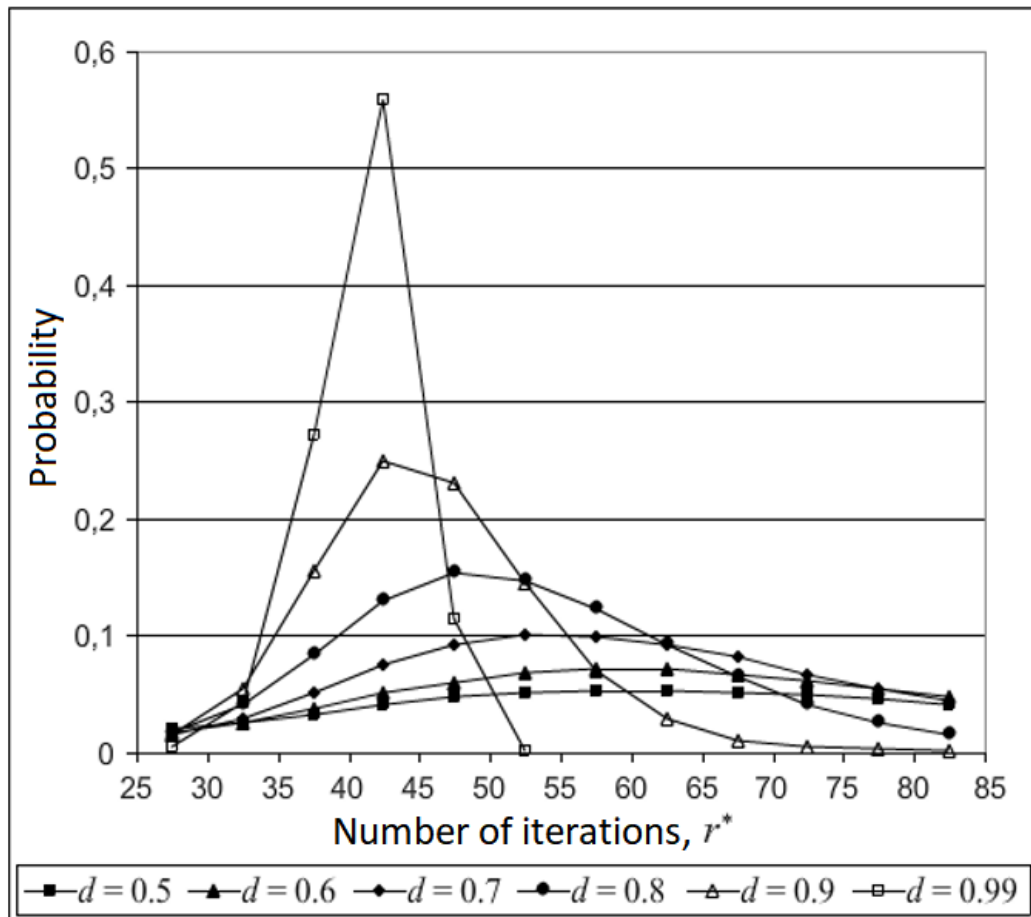


Figure 4.7 – Histograms of the r^* distribution, nonlinear model (4.3)

We examine the convergence rate of algorithm to a nonlinear model (4.3) for $d \in D$. Freedom of choice $F = 5$, modeling accuracy $\varepsilon = 10^{-6}$. The latter value was chosen because when $\varepsilon < 10^{-6}$, during histograms constructing, the parameter r^* often exceeds the maximum limit of iterations $R(F)$, which exhausts the available RAM (1.5 GB). For example, for $F = 5$, $R(F) = 4000$ was obtained. The results are presented in Fig. 4.7.

So, we can see the rate of convergence to this model is significantly lower than to the linear model. It should be noted that the rate of convergence to nonlinear models depends significantly on the model structure. For example, for the model

$$y = -4.82186 + 9.10802 \cdot x_1 - 7.82742 x_2 - 5.4407 \cdot x_3 - 2.3515 \cdot x_4 + 3.55896 \cdot x_5 - 1.37364 \cdot x_1^3 + 7.78903 \cdot x_1 x_2 x_4 + 8.92675 \cdot x_2^3 \quad (4.6)$$

It is significantly lower (see Fig. 4.8) than to the model (4.3) because it is simpler in terms of the number of relationships between nonlinear and linear terms than model (4.6).

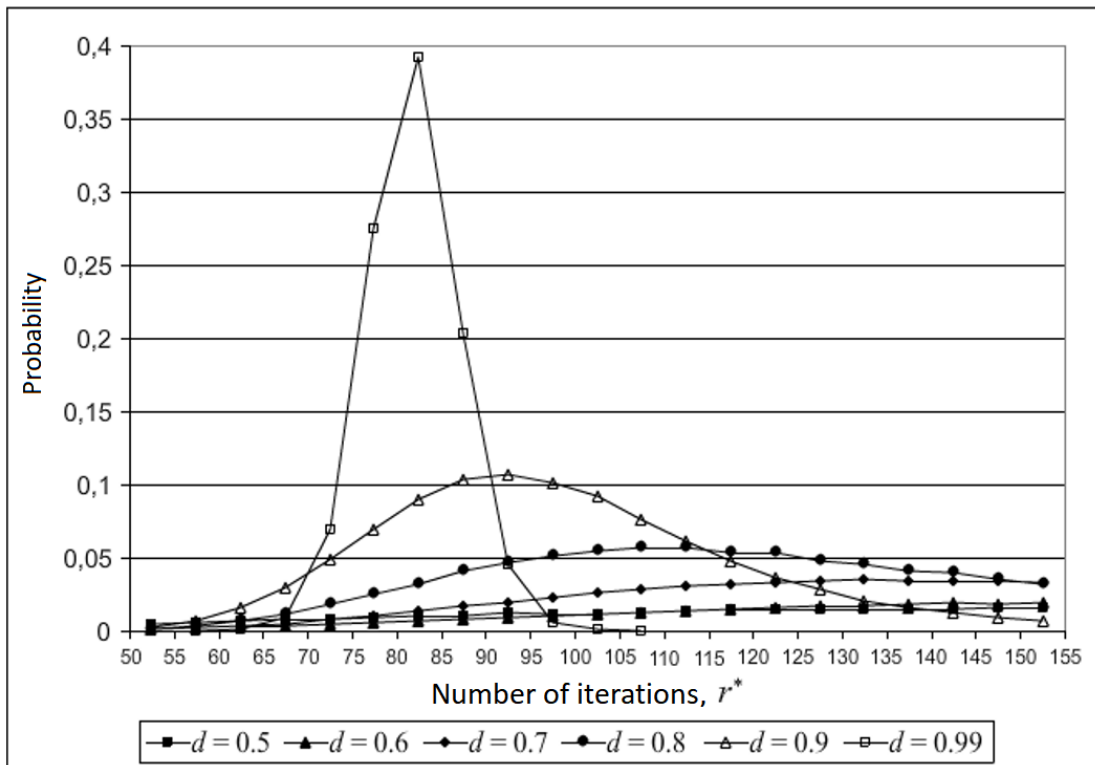


Figure 4.8 – Histograms of the r^* distribution, model (4.6)

Let us carry out experiments similar to the study of convergence to the linear model, where both the value of the determinant d and the number of model arguments varied, for the nonlinear model (4.3). The arguments of this nonlinear model are divided into two sets: linear: $x_1, x_2, \dots$ and multiplicative: $x_1^3, x_1 x_2 x_4, \dots$.

We will study the dependence of the algorithm's convergence rate on varying parameters d and s_{lin} and the constant number of multiplicative s_{mult} arguments. Model (4.3) was studied, $d \in D$, $s_{lin} \in S_{lin}$, $S_{lin} = \{5, 6, 7, 8, 9, 10\}$, $s_{mult} = 3$. Absolute values of the rounded model coefficients for s_{lin} and the measure of variance of the coefficients' contributions ψ are presented in Table 4.3.

Table 4.3 – Coefficients of nonlinear models and measure ψ

Saliva	$ \theta_0 $	$ \theta_1 $	$ \theta_2 $	$ \theta_3 $	$ \theta_4 $	$ \theta_5 $	$ \theta_6 $	$ \theta_7 $	$ \theta_8 $	$ \theta_9 $	$ \theta_{10} $	ψ
5	6.294	7.29	8.12	6.70	7.46	9.38	5.30	4.00	7.78	1.04	3.15	0,0035374
6												0,004711
7												0,007436
8												0,005845
9												0,014198
10												0,015051

Histograms were plotted for each pair (s_{lin}, d) , and the r_{prob} value corresponding to the maximum peak of the histogram was determined. Freedom of choice $F = 5$. The results are shown in Fig. 4.9.

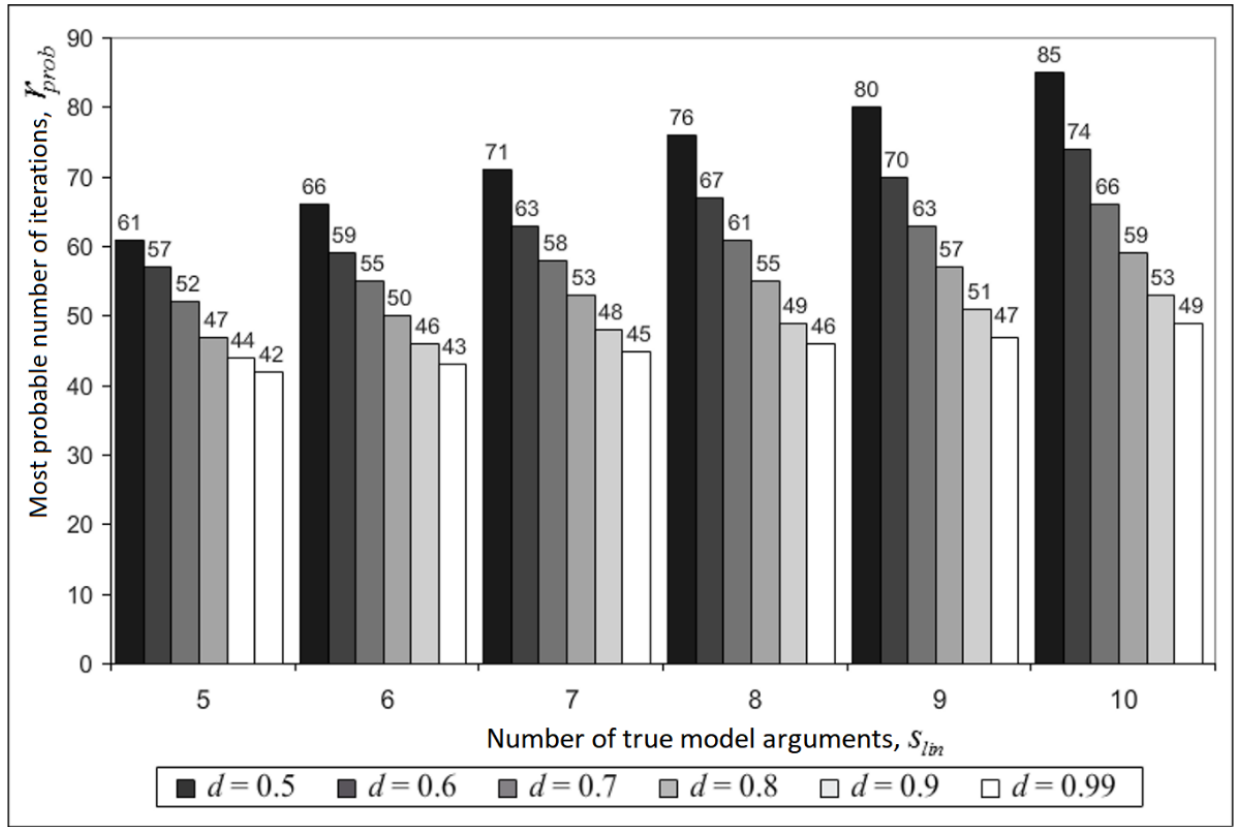


Figure 4.9 – Values of the number of rows r_{prob} for each pair (s_{lin}, d)

Let us complete the *third stage* of the methodology, which requires to fix the parameter d and examining the convergence rate as each algorithm parameter varies, while the other parameters remain constant. The studies below expand on this stage by adding experiments examining the change in the parameter r^* while varying both the freedom of choice F and the number of model arguments.

Stage 3. Step 2. For each algorithm parameter, examine the convergence rate as its value varies while the other parameters (including d) remain constant.

In the experiments, $d = 0.7$. We examine the linear model (4.2).

At the first stage the values of n for all values $d \in D$ were determined (Table 4.1), therefore $n = 25$. In enable comparison of results, the algorithm is examined with the same initial matrices as in the previous experiments. Freedom of choice $F \in \mathfrak{F}_{lin}$, $\mathfrak{F}_{lin} = \{5, 10, 25, 50, 100, 250, 500\}$. The results are presented in Fig. 4.10.

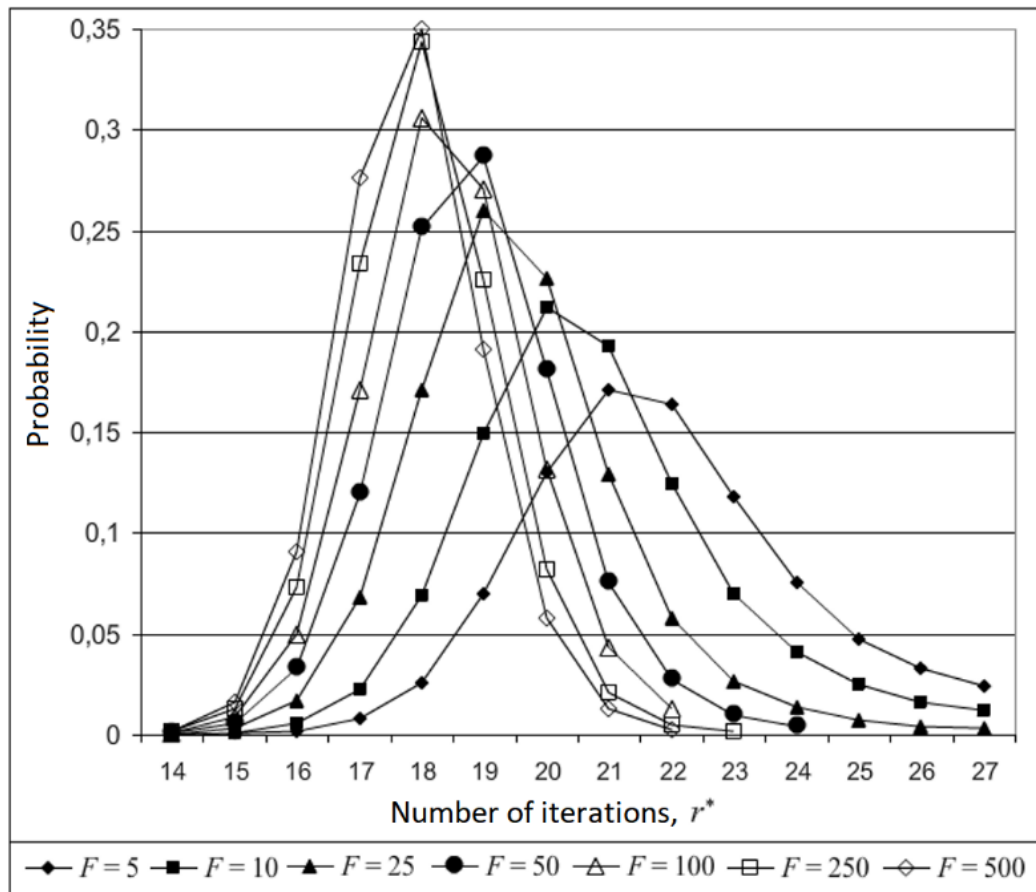


Figure 4.10 – Histograms of the r^* distribution with varying parameter F

As we can see, the freedom of choice F , like the determinant d directly affects the algorithm's convergence rate: the higher the value of F , the higher the rate of convergence (i.e., the lower the value of parameter r^*).

To illustrate the convergence process in more detail, Table 4.4 illustrates the convergence rate of the best averaged model for the peak probability of the histogram of the variant where $F = 50$ in Figure 4.10. The model is averaged over coefficients and criterion value.

Table 4.4 – Results of convergence to the linear model

No iteration	NRSS	θ_0	θ_1	θ_2	θ_3	θ_4	θ_5
5	$5.47 \cdot 10^{-3}$	6.35397	-7.16523	7.99056	6.57297	-7.46026	9.37736
7	$1.84 \cdot 10^{-4}$	6.29582	-7.28527	8.11217	6.69561	-7.33251	9.25849
9	$5.82 \cdot 10^{-6}$	6.29464	-7.2905	8.11572	6.69998	-7.45551	9.37293
11	$1.8 \cdot 10^{-7}$	6.2945	-7.29044	8.11584	6.70012	-7.46026	9.37737
13	$5.61 \cdot 10^{-9}$	6.29448	-7.29046	8.11583	6.70017	-7.46028	9.37735
15	$1.69 \cdot 10^{-10}$	6.29447	-7.29046	8.11584	6.70017	-7.46026	9.37736
True Model:		6.29447	-7.29046	8.11584	6.70017	-7.46026	9.37736

Let us now examine how the convergence rate of r^* changes with variations in two parameters: the freedom of choice F and the number of model arguments s . We will determine the nature of dependence of r^* on F , regardless of the number of arguments.

Investigation of the algorithm's convergence rate with variations in both F and s_{lin}

The parameters took values of $F \in \mathfrak{F}_{lin}$, $s_{lin} \in S_{Lin}$. Histograms were plotted for each pair (s_{lin}, F) , $d = 0.7$. Figure 4.11 shows the r_{prob} values for each pair of histograms.

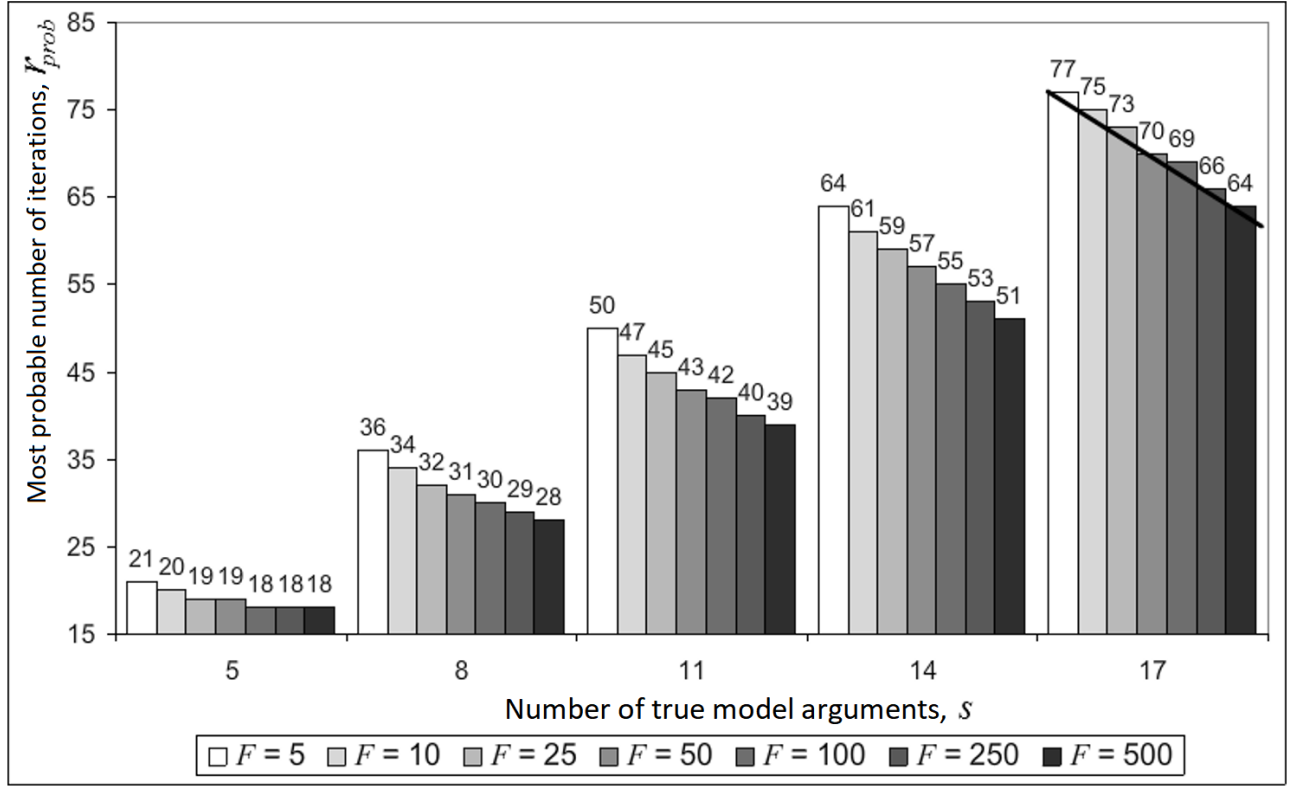


Figure 4.11 – Values of r_{prob} for each pair (s_{lin}, F)

As we can see, Figure 4.11 shows a linear dependence of the convergence rate on the freedom of choice F .

Now let's see what happens in the case of convergence to a nonlinear model.

Study of the convergence rate of the algorithm when varying freedom of choice F , nonlinear model.

We investigate the convergence rate to the nonlinear model (4.3) at $d = 0.7$, $n = 17$. Freedom of choice $F \in \mathfrak{F}_{nonlin}$, $\mathfrak{F}_{nonlin} = \{5, 10, 25, 50, 100, 250\}$. Histograms for $F = 5$ and $F = 10$ were obtained with $RN = 106$ in order to satisfy the specified accuracy $\delta = 5\%$.

As we can see from the results presented in Figure 4.12, freedom of choice again has a direct proportional effect on the convergence rate of the algorithm, however, the nature of histograms is significantly different from the histograms for the linear model: when F changes within $[25-100]$, histograms have two extrema.

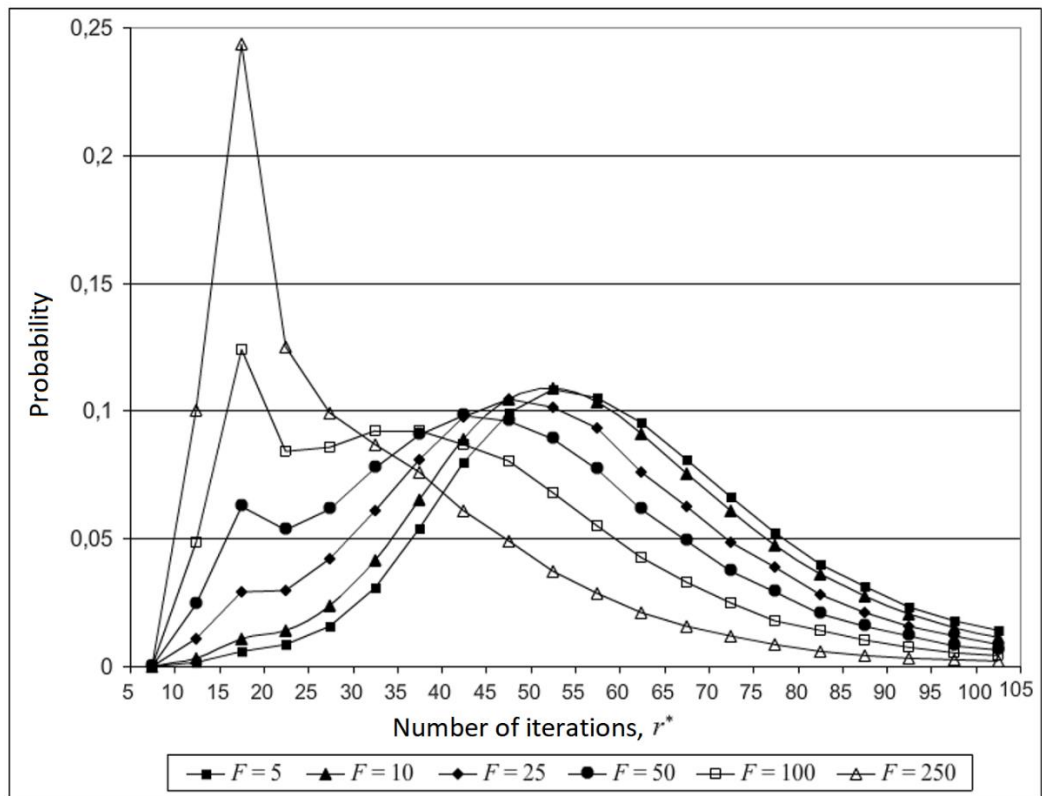


Figure 4.12 – Histograms of the r^* distribution when constructing a nonlinear model with varying parameter F

It should be said that when constructing histograms for any nonlinear model with increasing $F = 1, 5, 10, \dots, 250$ the left extremum becomes global, while with decreasing F the right extremum becomes global. Therefore, increasing F from 50 to 100, increases the convergence rate r^* stepwise by about 3 times (from $r^* = 48$ to $r^* = 17$ iterations).

Table 4.5 illustrates the convergence rate of the best averaged nonlinear model for the histogram in Figure 4.12, where $F = 50$. The model is averaged by coefficient values and criterion.

Table 4.5 – Results of convergence to the nonlinear model (4.3)

N ₀ Iterations	NRSS	θ_0	θ_1	θ_2	θ_3	θ_4	θ_5	θ_6	θ_7	θ_8
8	$1.4 \cdot 10^{-3}$	5.55	-3.06	8.65	6.66	-7.42	9.37	0	0	1.01
12	$6.4 \cdot 10^{-4}$	5.70	-3.36	8.47	6.68	-7.44	9.37	-0.04	0	1.62
15	$4.7 \cdot 10^{-4}$	5.75	-3.47	8.40	6.69	-7.44	9.37	-0.05	0	1.84
25	$2.9 \cdot 10^{-4}$	5.84	-3.65	8.27	6.69	-7.45	9.37	0.003	-0.088	2.21
35	$2.4 \cdot 10^{-4}$	5.89	-3.73	8.21	6.69	-7.45	9.37	0.017	-0.11	2.39
44	$2.3 \cdot 10^{-4}$	5.91	-3.77	8.18	6.69	-7.45	9.37	0.033	-0.13	2.48
True Model:		6.22	-7.29	8.11	6.7	-7.46	9.37	8.26	-5.57	2.64

As we can see from Table 4.5, due to the experiment's specified accuracy $\varepsilon = 10^{-6}$ the algorithm could not converge to model (4.3) with sufficient accuracy. However, when constructing the model

$$y = 6.29447 - 7.29046 \cdot x_1 + 8.11584 \cdot x_2 + 6.70017 x_1^2 - 7.46026 \cdot x_1 x_2 + 9.37736 x_2^2 \quad (4.7)$$

with algorithm parameters $n = 50$, $RN = 104$, $F = 50$, the amount of RAM was sufficient for the algorithm to converge with an accuracy $\varepsilon = 10^{-12}$ (see Table 4.6).

Table 4.6 – Results of convergence to nonlinear model (4.7)

No iterations	NRSS	θ_0	θ_1	θ_2	θ_3	θ_4	θ_5
5	$1.34 \cdot 10^{-2}$	5.44639	0	7.55181	0	-7.11667	9.67987
10	$5.11 \cdot 10^{-3}$	6.62952	-6.59114	6.74778	4.97901	-5.74463	9.79954
20	$1.67 \cdot 10^{-3}$	6.35726	-6.76131	7.61762	5.85519	-6.91197	9.56507
43	$3.2 \cdot 10^{-4}$	6.26284	-6.93145	7.98576	6.32843	-7.42516	9.47388
100	$7.78 \cdot 10^{-6}$	6.28432	-7.22814	8.1131	6.64537	-7.47706	9.38612
200	$5.55 \cdot 10^{-9}$	6.29436	-7.28942	8.11563	6.69918	-7.46045	9.37758
289	$1.6 \cdot 10^{-11}$	6.29448	-7.29046	8.11582	6.70016	-7.46025	9.37736
True Model:		6.29447	-7.29046	8.11584	6.70017	-7.46026	9.37736

This concludes the study of the convergence rate of the RIA CS using the described methodology. Recall that its goal was to study the parameters influencing the convergence rate of the iterative algorithm in order to obtain both a quantitative and qualitative picture of this influence, as well as an analytical form of these dependencies. By measuring these parameters based on the obtained data sample and substituting them into the obtained dependencies, we determine the number of iterations $R = r^*$ necessary to achieve a solution with a given accuracy ε .

We believe we have successfully answered many of these questions using computer modeling using RIA CS:

- we were able to identify parameters that influence the rate of convergence: data characteristics - the determinant d of the correlation matrix of variables, the number of model arguments s ; algorithm parameters – freedom of choice F , modeling accuracy ε ;
- parameters F and d have an inversely proportional effect on the number r^* , i.e. the larger F , the smaller r^* , and directly proportional to the rate of convergence, because with smaller number of iterations r^* , is higher the convergence rate;
- parameters s and ε have a direct proportional effect on the number r^* and inversely proportional on the convergence rate;
- the parameters F and s have a linear effect on the convergence rate, while parameter d has a nonlinear effect;
- an analytical formula was obtained for the dependence of the number of iterations r^* on the number of model arguments s for $d = 1$ (the matrix $\mathbf{X}$ is orthogonal) and the freedom of choice $F = 5$, namely $r^* = 3s$ for a linear model and $r^* = 40s$ for a nonlinear model.

An interesting question is: "To what solution does the algorithm converge if the number of observations n is less than the number of vector-columns of the matrix $\mathbf{X}$ or the model arguments s ?" In the previous chapter, it was theoretically proven that RIA CS converges to a certain solution $\mathbf{y}^{\text{pr}}$ – the limit point. Let us investigate this question experimentally.

Study of the speed of convergence of the algorithm when the number of observations is less than the number of model arguments.

The following linear model is being studied:

$$y = 6.29447 - 7.29046 \cdot x_1 + 8.11584 \cdot x_2 + 6.70017 \cdot x_3 - 7.46026 \cdot x_4 + 9.37736 \cdot x_5 + \\ + 8.26752 \cdot x_6 - 5.57932 \cdot x_7 + 2.64719 \cdot x_8 - 3.83666 \cdot x_9 - 8.04919 \cdot x_{10}$$

The following parameters are constant: $n = 8$, $\varepsilon = 10^{-12}$. The variable parameter is the freedom of choice F . Two experiments were conducted with $F = 25$ and $F = 50$. The results of convergence are presented in Tables 4.7, 4.8. As can be seen from these tables, the convergence of algorithm to a fixed point for $n < s$ is confirmed by experimental results.

It should be noted that for $F = 26$ and $F = 25$, the limit point remains the same. Note also that in the latter case, the limit point is formed by excluding two arguments with indices 7 and 10 from the model, thereby determining the system of linear equations.

Table 4.7 – Algorithm convergence process for an undefined problem, $F = 25$

Number of iterations	$NRSS$	θ_0	θ_1	θ_2	θ_3	θ_4	θ_5	θ_6	θ_7	θ_8	θ_9	θ_{10}
10	0.000877	-2.7716	-14.2445	10.3339	11.6999	0	0	2.757	0	2.64018	4.98151	0
15	0.000219	-2.7787	-14.1506	9.74994	11.9245	-0.6864	0	3.16385	0	2.73475	4.9197	0
20	0.000163	-2.8198	-14.1522	9.66811	11.9259	-0.7604	-0.1399	3.32365	0	2.85623	4.92025	0
50	$6.86 \cdot 10^{-5}$	-2.3664	-14.2494	9.57102	11.8061	-1.29257	-0.7775	3.58756	0	3.02755	4.87084	0
100	$1.65 \cdot 10^{-5}$	-1.8754	-14.3661	9.49095	11.674	-1.80591	-1.3843	3.80524	0	3.15816	4.83009	0
200	$1.01 \cdot 10^{-6}$	-1.5395	-14.4568	9.43429	11.5945	-2.15582	-1.8661	3.93882	0	3.29628	4.83186	0
280	$1.08 \cdot 10^{-7}$	-1.4697	-14.474	9.42145	11.5753	-2.23111	-1.9695	3.97015	0	3.32969	4.83164	0
363	$1.05 \cdot 10^{-8}$	-1.4448	-14.4802	9.41747	11.5688	-2.25789	-2.0042	3.98108	0	3.33893	4.83157	0
446	$1.04 \cdot 10^{-9}$	-1.4378	-14.4824	9.41624	11.567	-2.26513	-2.0154	3.98442	0	3.34242	4.83189	0
528	$1.04 \cdot 10^{-10}$	-1.4357	-14.483	9.41572	11.5664	-2.26747	-2.0187	3.98545	0	3.34351	4.83197	0
561	$4.16 \cdot 10^{-11}$	-1.4352	-14.4832	9.41568	11.5663	-2.26788	-2.0193	3.98562	0	3.34369	4.83197	0
True model		6.2944	-7.29046	8.11584	6.70017	-7.46026	9.37736	8.26752	-5.5793	2.64719	-3.8366	-8.0491

Table 4.8 – Algorithm convergence process for an indeterminate problem, $F = 50$

No. of iteration	$NRSS$	θ_0	θ_1	θ_2	θ_3	θ_4	θ_5	θ_6	θ_7	θ_8	θ_9	θ_{10}
10	$8.94 \cdot 10^{-6}$	1.9170	-13.419	8.99505	11.029	-6.744	0	5.05671	0	0	1.78293	0
12	$3.75 \cdot 10^{-6}$	1.7756	-13.387	8.94388	11.092	-6.69262	0.10697	5.08558	0	0	1.7931	0
17	$2.20 \cdot 10^{-6}$	1.7555	-13.384	8.94188	11.0895	-6.65845	0.15936	5.06982	0	-0.0107	1.7927	0
90	$2.20 \cdot 10^{-7}$	1.6793	-13.360	8.97046	11.1032	-6.55963	0.29076	5.01579	0	-0.053	1.79107	0
172	$2.21 \cdot 10^{-8}$	1.6461	-13.352	8.97642	11.1114	-6.52327	0.3408	5.00079	0	-0.0681	1.79056	0
254	$2.22 \cdot 10^{-9}$	1.6356	-13.349	8.97856	11.1143	-6.51194	0.35618	4.9963	0	-0.0732	1.79039	0
337	$2.19 \cdot 10^{-10}$	1.6322	-13.348	8.97912	11.1152	-6.50822	0.36116	4.9947	0	-0.0746	1.79035	0
392	$4.75 \cdot 10^{-11}$	1.6313	-13.348	8.97926	11.1154	-6.50738	0.36243	4.99436	0	-0.075	1.79033	0
True model		6.2944	-7.2904	8.11584	6.70017	-7.46026	9.37736	8.26752	-5.5793	2.64719	-3.8366	-8.0491

The next important property of the GMDH algorithms is the identification property. As noted earlier, it is related to the task of identifying an exact model. Let us formulate it.

4.2 Study of the Identifying Property of Algorithms

he problem of identifying the true model consists of determining:

- true arguments $\tilde{x}_1, \dots, \tilde{x}_{s_t}$, from a set of arguments $\tilde{x}_1, \dots, \tilde{x}_{s_t}, \tilde{x}_1, \dots, \tilde{x}_{s_f}$ where $\tilde{x}_1, \dots, \tilde{x}_{s_f}$ is a set of dummy (false) arguments,
- true values of the parameters $\tilde{\Theta}$ of the function Θ

$$\tilde{f}(\tilde{x}_1, \dots, \tilde{x}_{s_t} | \tilde{\Theta}) = \sum_{i=1}^{s_t} \tilde{\theta}_i \tilde{x}_i,$$

for a given matrix $W = (X : y)$, $\dim X = n \times m$, $X = (\tilde{X} : \bar{X})$, $\dim \tilde{X} = n \times s_t$, $\dim \bar{X} = n \times s_f$, $s_t + s_f = m$.

The identifying property of the algorithm is characterized by the probability p_1 that the model contains only true arguments. We also introduce the probability p_2 that the constructed model contains all true arguments, but may also contain false ones, in order to show the possibility of filtering out the latter by the coefficient value. We examine the identifying properties of the version of the GRIA whose generator performs a complete sorting of arguments at each iteration, previously designated as the RIA CS, when:

- various input data: the determinant of the correlation matrix of variables $d \in D$, $D = \{0.5, 0.6, 0.7, 0.8, 0.9, 0.99\}$;
- different freedom of choice of the algorithm $F \in \mathfrak{F}$, $\mathfrak{F} = \{5, 10, 25, 50, 100\}$.

4.2.1 Study of the Impact of Data Characteristics

A linear model (4.2) was identified, i.e. the number of true arguments $s_t = 5$. The number of false arguments $s_f \in S_f$, $S_f = \{5, 10, 25, 50, 100, 250\}$. The algorithm parameters were as follows: $RN = 105$, $\varepsilon = 10^{-7}$. Probabilities p_1 and p_2 were calculated for each pair of values (d, s_f) , $d \in D$, $s_f \in S_f$. The results are presented in Figures 4.13 and 4.14.

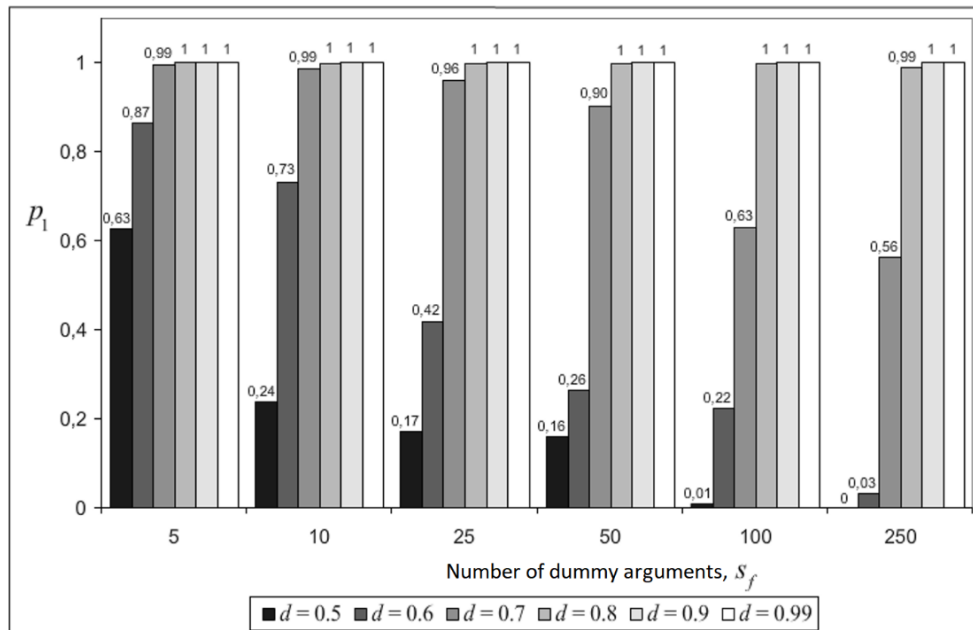


Figure 4.13 – Probability of identifying a model that contains only true arguments

The following conclusions can be drawn from Figure 4.13:

- For ill-conditioned matrices ($d \in [0.5; 0.8]$), an increase in the number of dummy arguments reduces the probability of model identification.
- The probability of model identification remains unchanged with the number of dummy arguments is proportional to the number of true arguments.
- For well-conditioned matrices ($d \geq 0.8$), the algorithm identifies the model with probability "1" regardless of the number of dummy arguments.

To characterize the probability of rejection of a dummy argument in models with coefficient c_f , a histogram of the distribution of absolute value of this coefficient, $|c_f|$ was constructed. The histogram is shown in Figure 4.15 for the variant $s_f = 25$, $d = 0.6$ in Figure 4.14.

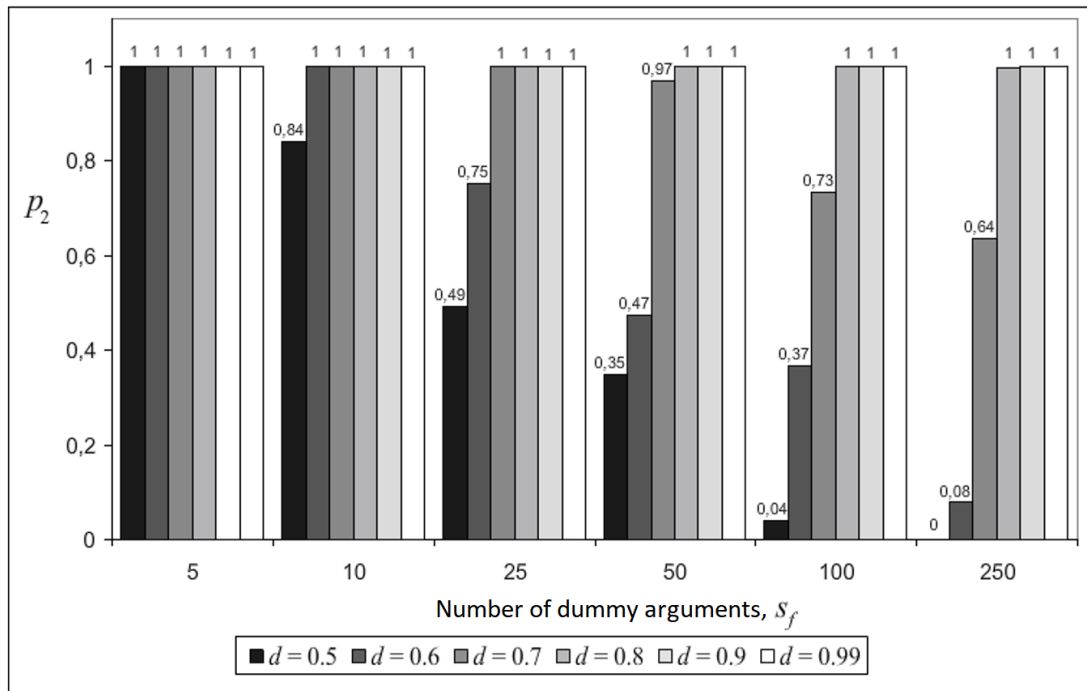


Figure 4.14 – Probability of identifying a model containing both true and false arguments

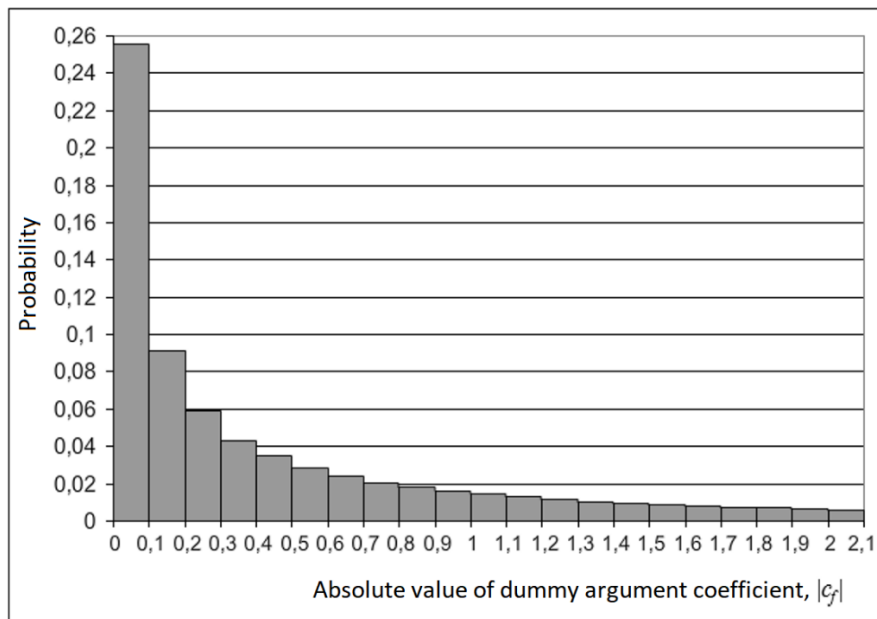


Figure 4.15 – Histogram of the distribution of the maximum coefficient of a dummy argument for the possibility to eliminate it

As we can see in Fig. 4.15, the most probable value of the coefficient c_f lies within the range $[-0.1; 0.1]$. Taking into account that the modulus of the value of the true coefficients of the model (4.2) lies within the range $[7; 9]$, dummy arguments are easily identified and eliminated.

The identification process for model (4.2) is shown in Table 4.9 for variant $s_f = 25$, $d = 0.6$ in Figure 4.13.

Table 4.9 – Linear model identification results

No Iterations	NRSS	$\tilde{\theta}_0$	$\tilde{\theta}_0$	$\tilde{\theta}_2$	$\tilde{\theta}_3$	$\tilde{\theta}_4$	$\tilde{\theta}_5$
5	$2.55 \cdot 10^{-2}$	6.54115	-6.63917	7.54099	6.05947	-6.82743	8.83562
7	$2.34 \cdot 10^{-3}$	6.31449	-7.18139	8.04159	6.56916	-7.35993	9.33722
9	$1.80 \cdot 10^{-4}$	6.29524	-7.27238	8.10511	6.67651	-7.44416	9.37639
11	$9.96 \cdot 10^{-6}$	6.2938	-7.28752	8.11433	6.69689	-7.45822	9.37867
13	$4.67 \cdot 10^{-7}$	6.29436	-7.29003	8.11576	6.69963	-7.46003	9.37753
15	$2.37 \cdot 10^{-8}$	6.29446	-7.29039	8.1158	6.70007	-7.46021	9.37741
True Model:		6.29447	-7.29046	8.11584	6.70017	-7.46026	9.37736

The probability of such model appearing, according to the results presented in Fig.4.13, is $p_1 = 0.952$. The average model calculation time is 0.0024 seconds on an Intel Core i3 2.27MHz processor.

Let us now see how the freedom of choice F affects the probability of identifying p_1 of the model.

4.2.2 Study of the Influence of Algorithm Parameters

In the experiment, a linear model was built (4.2), with $d = 0.7$, $RN = 10^5$, $\varepsilon = 10^{-7}$. In the same way, for each variant (F, s_f) , $F \in \mathfrak{F}$, $s_f \in S_f$, the probability p_1 was calculated. The result is shown in Figure 4.16.

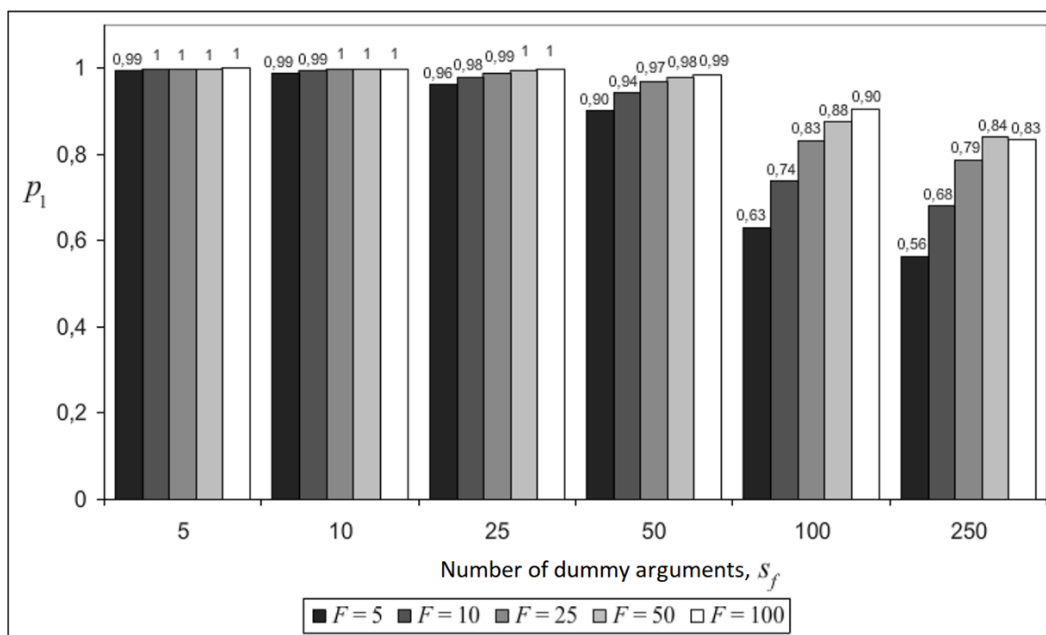


Figure 4.16 – Probability of p_1 when changing the parameters s_f and F

From analysis of the results of Figure 4.16, we can conclude that:

- increasing the parameter F leads to an increase in the probability of p_1 for any number of dummy arguments s_f ;
- there is a certain threshold value F^* at which the value of p_1 does not increase and even falls, for example at $s_f = 250$ and increasing F^* in the range [50-100]. Therefore, in practice setting the number F greater than the threshold F^* makes no sense.

One very important question remains: what happens in case of identification of nonlinear models? Unfortunately for GRIA, the probability of identifying the model p_1 is zero, no matter what the freedom of choice F is chosen. The reason for this is the high correlation between true and false arguments. Recall that it lies within the range of [0.6; 0.9]. As a result, inaccurate parameter estimation occurs with true arguments, and as a result, the algorithm begins introducing false arguments into the model. In other words, the algorithm replaces true arguments with false ones due to their high mutual correlation..

Table 4.10 summarizes the experimental results of the RIA CS identification and convergence properties.

Table 4.10 – Summary of the RIA CS study

	Determinant $d \in [0.5; 0.75]$	Determinant $d \in [0.75; 1]$	Determinant $d \approx 0$
<i>Model type</i>	Linear model	Linear model	Nonlinear model
<i>Convergence rate</i>	Medium	High	Very low
<i>Number of iterations r^*</i>	$r^* = 5s$	$r^* = 3s$	$r^* > 40s$
<i>Probability of identification</i>	Decreases in proportion to the increase in the number of false arguments	Equal to "1"	Equal to "0"

4.3 Study of the Noise-Immunity Property of Modeling

According to the GMDH theory of noise-immunity of modeling [9], an algorithm operates correctly if, under conditions of increasing additive uncorrelated noise level in the output vector $\mathbf{y}$, it simplifies the model structure (reduces the number of model monomials). In other words, as the noise level increases, minimum of the "external" CR criterion shifts toward simpler model structures.

This property qualitatively depends both on the parameter estimation method and on the specific type of selection criterion that uses the residual sum of squares (RSS) in calculations. As an example for a numerical experiment, we consider RIA CS - a version of GRIA with a complete sorting of model structures at each iteration, since its internal convergence has been theoretically proven.

In the experiment, let's assume that noise is generated using a pseudorandom number generator according to the formula:

$$y = \tilde{y} + \xi, \quad \xi = \alpha(2\zeta - 1) \frac{\tilde{y}_{\max} - \tilde{y}_{\min}}{200},$$

where α is the noise level in percentage, %; ζ - a pseudo-random variable distributed uniformly over the interval [0; 1].

The algorithm stops when the specified modeling accuracy $\varepsilon = 10^{-9}$ is reached. The selection criterion is the normalized regularity criterion NAR_B , calculated on the test sample B :

$$NAR_B = \sum_{i=1}^{n_B} (\tilde{y}_{B,i} - \hat{\tilde{y}}_{B,i})^2 \bigg/ \sum_{i=1}^{n_B} \tilde{y}_{B,i}^2 ,$$

where $\tilde{y}_{B,i}$ is the value of $y_{B,i}$ of the vector $\mathbf{y}_B$ centred on the training sample A ; $\hat{\tilde{y}}_{B,i}$ – the corresponding model output value.

The $\mathbf{X}$ matrix in the experiment contained both true and dummy column-vectors. The noise-immunity of modeling was studied when constructing both a linear

$$\begin{aligned} \tilde{y} = & 6.29447 - 7.29046 \cdot x_1 + 8.11584 \cdot x_2 + 6.70017 \cdot x_3 - 7.46026 \cdot x_4 + 9.37736 \cdot x_5 + \\ & + 8.26752 \cdot x_6 - 5.57932 \cdot x_7 + 2.64719 \cdot x_8 - 3.83666 \cdot x_9 - 8.04919 \cdot x_{10}, \end{aligned} \quad (4.8)$$

and nonlinear model

$$\begin{aligned} \tilde{y} = & 6.29447 - 7.29046 \cdot x_1 + 8.11584 \cdot x_2 + 6.70017 \cdot x_3 - 7.46026 \cdot x_4 + 9.37736 \cdot x_5 + \\ & + 8.26752 \cdot x_1^2 - 5.57932 \cdot x_1 x_2 + 2.64719 \cdot x_1 x_3 - 3.8366 \cdot x_1 x_4 - 8.0492 \cdot x_1 x_5. \end{aligned} \quad (4.9)$$

The freedom of choice the algorithm $F = 50$. Data matrices $\mathbf{X}$ for different values of parameter α were the same in both experiments.

When constructing a linear model, the values $n_A = 40$, $s_f = 50$ were specified, while for nonlinear model $n_A = 110$. The ratio $n_A : n_B = 2:1$ was equal to the number of linear dummy arguments and was equal to the number of nonlinear ones and was 25.

Tables 4.11 and 4.12 show the identification results for models (4.8) and (4.9) at different noise levels in the output variable $\mathbf{y}$.

As can be seen from Tables 4.11 and 4.12, the algorithm exhibits property of noise-immunity: as the noise level – the parameter α increases, the algorithm simplifies the model by discarding monomials with small coefficients. An interesting feature is that the algorithm reveals the "true" structure of the model at a noise level 10-20%.

It is important to note that, despite the presence of fifty dummy arguments in matrix $\mathbf{X}$, the constructed models do not contain a single such argument.

The ability to recover a model in the presence of noise depends significantly on the data characteristics, in our case, on the parameter d . For example, in the case of a nonlinear model, if the number of points $n_A = 10000$ ($d = 0.998$), then the algorithm reveals the true structure of the model even at 70% of the noise level. In the given examples, the parameter $d_{lin} = 0.41$ (Table 4.11), and the determinant of the matrix containing the nonlinear transformations of the input variables $d_{nonlin} = 0.993$ (Table 4.12).

This concludes the study of the three main properties of iterative GMDH algorithms..

Table 4.11 – Results of noise-immunity of the linear model modeling

α	k	NAR_g	$N_{\text{iterat.}}$	$\check{\theta}_0$	$\check{\theta}_1$	$\check{\theta}_2$	$\check{\theta}_3$	$\check{\theta}_4$	$\check{\theta}_5$	$\check{\theta}_6$	$\check{\theta}_7$	$\check{\theta}_8$	$\check{\theta}_9$	$\check{\theta}_{10}$
0	10	$7.5 \cdot 10^{-10}$	37	6.2944	-7.2901	8.1158	6.7004	-7.4603	9.3772	8.2676	-5.5797	2.6469	-3.8363	-8.0494
5	10	$4.6 \cdot 10^{-4}$	62	6.5888	-6.9505	8.1627	6.5190	-7.2623	9.6069	7.9152	-5.9482	2.2217	-3.5620	-7.6909
10	10	$2.6 \cdot 10^{-3}$	61	6.8120	-6.3543	7.8509	6.6677	-7.0162	9.8966	7.8674	-6.0584	2.3652	-3.6313	-7.6486
20	9	0.01069	67	7.4472	-5.2685	7.8406	6.1688	-7.0823	9.50713	8.2898	-6.0215	0	-3.4673	-7.0844
30	9	0.02623	63	5.7095	-4.0877	7.8582	6.3132	-6.8623	9.6633	8.9417	-6.1445	0	-3.6698	-5.8810
40	7	0.04198	47	6.0133	-6.6	6.6990	6.7982	0	10.3763	0	-6.8650	0	-4.2594	-2.2245
50	7	0.01424	49	2.5224	-4.1614	6.2481	7.8985	0	11.5922	0	-4.9241	0	-3.3325	-3.2814
60	7	0.03809	60	2.1845	-4.6706	5.4338	8.7070	0	9.52767	0	-1.4751	0	-6.2766	-2.0527
70	5	0.08132	48	4.4819	-3.2823	0	9.1989	0	9.90538	0	0	0	-6.2435	-0.4079
80	5	0.09756	36	12.886	-1.5004	0	5.6809	0	11.1849	0	-0.8887	0	0	-2.0660
90	5	0.07627	55	9.3349	-2.3567	5.2908	5.8686	0	10.6238	0	0	0	-5.4252	0
True model				6.2944	-7.2904	8.1158	6.7001	-7.4602	9.37736	8.2675	-5.5793	2.6471	-3.8366	-8.0491

Table 4.12 – Results of noise-immunity of the nonlinear model modeling

α	k	NAR_B	$N_{\text{iterat.}}$	$\bar{\theta}_0$	$\bar{\theta}_1$	$\bar{\theta}_2$	$\bar{\theta}_3$	$\bar{\theta}_4$	$\bar{\theta}_5$	$\bar{\theta}_6$	$\bar{\theta}_7$	$\bar{\theta}_8$	$\bar{\theta}_9$	$\bar{\theta}_{10}$
0	10	$5.78 \cdot 10^{-8}$	329	6.3	-7.3080	8.1134	6.7010	-7.4628	9.3738	8.2770	-5.5744	2.6456	-3.8318	-8.0423
5	10	0.00740	219	6.6671	-8.5611	8.0818	6.8611	-7.6034	8.76584	8.3497	-5.4853	2.5207	-3.5899	-6.6560
10	10	0.02861	155	6.8820	-9.1552	7.8312	7.1394	-7.7921	8.48477	8.1759	-5.0757	2.2394	-3.2633	-6.0729
20	10	0.11153	126	6.8627	-7.6834	7.7178	7.9522	-8.6552	6.6046	5.9917	-4.9422	1.0632	-1.6912	-2.1237
25	9	0.16724	96	7.1794	-7.1386	7.1052	8.0783	-8.6873	5.7033	4.7524	-4.1652	0.8790	-1.5946	0
30	8	0.222	84	7.2417	-6.9170	6.8162	8.7045	-8.7789	5.4180	4.2225	-3.5940	0	-1.5074	0
40	8	0.33561	71	7.6911	-6.8685	5.8751	8.7167	-9.5601	5.5813	2.7267	-2.2309	0.0537	0	0
50	7	0.42821	45	5.3337	0	7.7313	8.7783	-7.4317	5.3505	1.0634	-5.381	0	-4.2865	0
60	6	0.52146	42	7.0619	0	7.4461	8.3246	-9.3855	5.1291	0	-5.0265	0.5393	0	0
70	6	0.59462	34	6.7691	0	7.4466	8.3520	-9.3663	5.0046	0	-4.8537	0.5942	0	0
80	6	0.65443	33	6.7139	0	7.4866	8.4464	-9.39	4.9221	0	-4.8397	0.6078	0	0
90	5	0.70469	27	6.9869	0	7.7947	8.9252	-9.296	5.0556	0	-5.6385	0	0	0
True model:				6.2944	-7.2904	8.1158	6.7001	-7.4602	9.37736	8.2675	-5.5793	2.6471	-3.8366	-8.0491

In conclusion of this chapter, we will focus on examining the algorithm's parameters.

In our experiments, we focused on two parameters: the modeling accuracy ε and the freedom of choice F . The first is related to the algorithm's stopping rule. In practical applications, ε value is set to 10^{-4} or 10^{-5} .

The second parameter, the freedom of choice F is always present in the algorithm. In the next series of experiments, we would like to shed some light on this parameter, so that the user has some experience with further use of the algorithm in practice, and an understanding of how it influences the resulting solution.

This study focuses on determining the influence of this parameter on the minimum value of the *NRSS* model selection criterion. As an example, we will examine the RIA CS.

4.4 Study of the Parameter of Freedom of Choice

The accuracy of structural-parametric identification of the linear model (4.2) is studied for various values of the algorithm's freedom of choice. Figure 4.10 displays histograms of the distribution of the number of iterations r^* for different freedom of choice $F \in \mathfrak{F}_{lin}$, $\mathfrak{F}_{lin} = \{5, 10, 25, 50, 100, 250, 500\}$. In this experiment, each of the model coefficients corresponding to the peaks of the histograms of Figure 4.10 and the resulting values of the selection criterion were averaged. Families of averaged best models were constructed at iterations numbered 5, 7, 9, 11, 13, 17 for each $F \in \mathfrak{F}_{lin}$. The results are presented in Fig. 4.17.

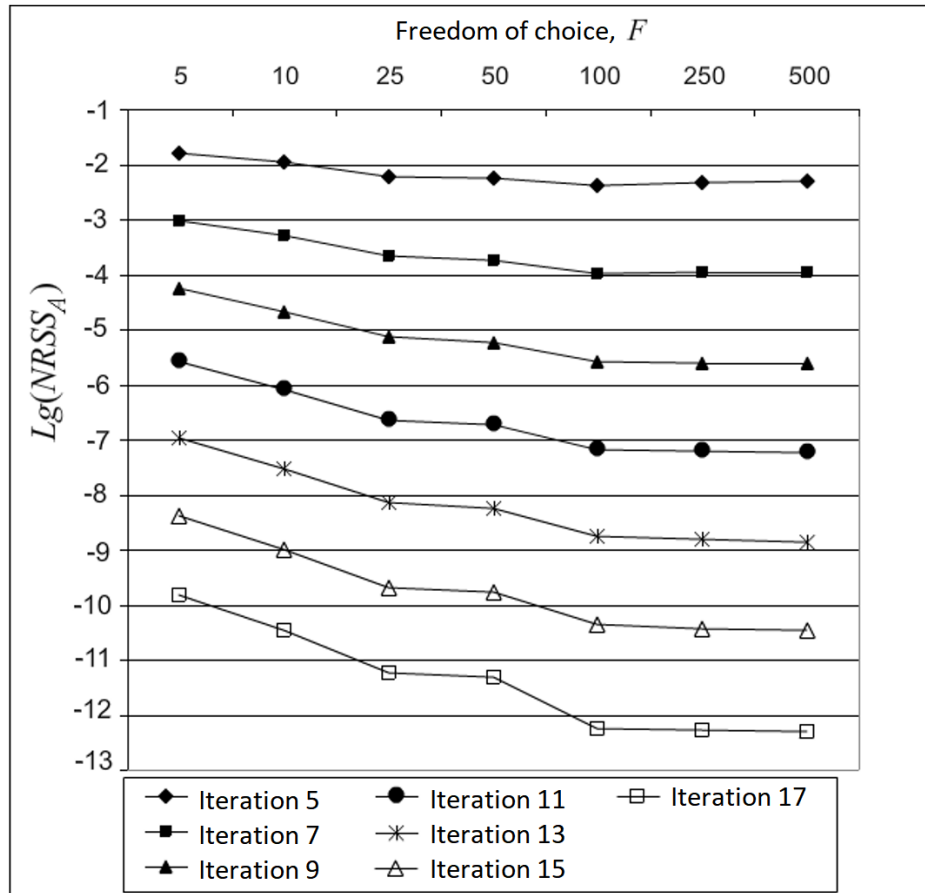


Figure 4.17 – Changes of *NRSS* criterion minima with variations in the *F* parameter at different iterations

- Analysis of the results presented in Fig. 4.17, allows us to draw the following conclusions:
- freedom of choice F significantly influences the change in the NRSS criterion minimum, with the higher the iteration number, the stronger this influence;
 - there is a value F^* above which the value of the criterion stabilizes. In this example, we can set $F^* = 100$.

We now explore the nonlinear model (4.3). The result is shown in Figure 4.18.

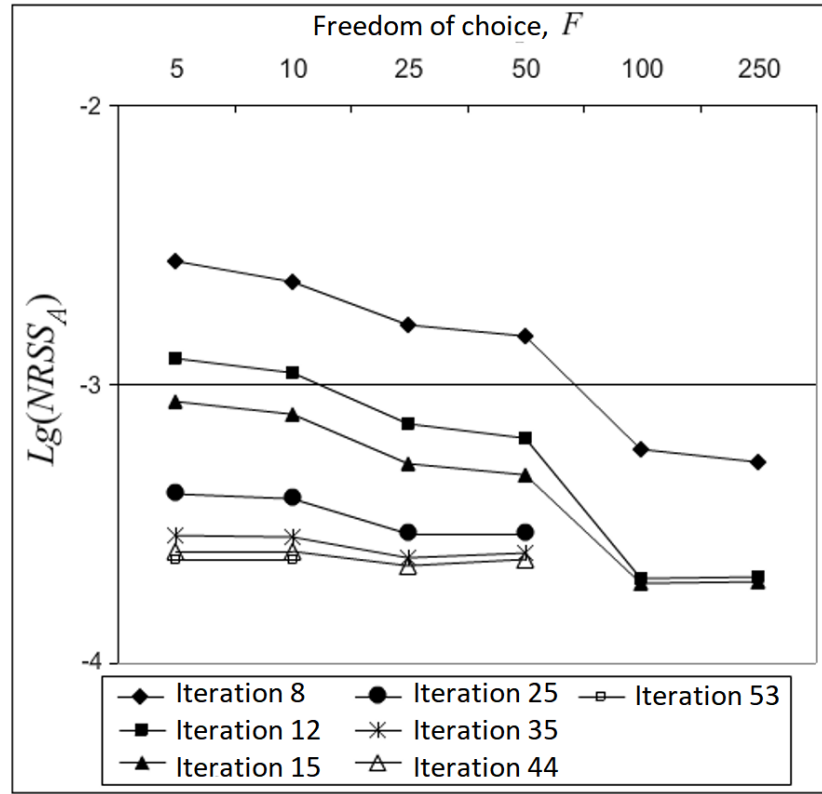


Figure 4.18 – Change in NRSS criterion minimum with changing of parameter F , model (4.3)

The absence of points on the graphs for some values of F indicates that the algorithm converged to a solution in fewer iterations. The figure clearly shows jumps in the number of r_{prob} iterations corresponding to the highest probability as the maximum transitions from the right peak of the histograms in Figure 4.12 to the left: this maximum occurred on the interval $F \in [50; 100]$. The following conclusions can be drawn from the results of this experiment:

- When studying nonlinear models, as with linear models, there is a value of F^* for the right maximum of the histograms, above which the NRSS criterion stabilizes. In the example with the nonlinear model F^* is equal to 100, just like with the linear model.
- In the case of the nonlinear model, the impact of the same F values on the minimum NRSS criterion is less (the range of NRSS variation is smaller) than for the linear model.

From an analysis of the graphs in Figure 4.18 we can conclude that with increasing iteration number, the impact of the F parameter on the minimum NRSS criterion decreases; however, this example is the exception rather than the rule. For example, when studying the nonlinear model (4.7), the graphs of the minimum NRSS versus F are similar to those for the linear model (see Figure 4.19).

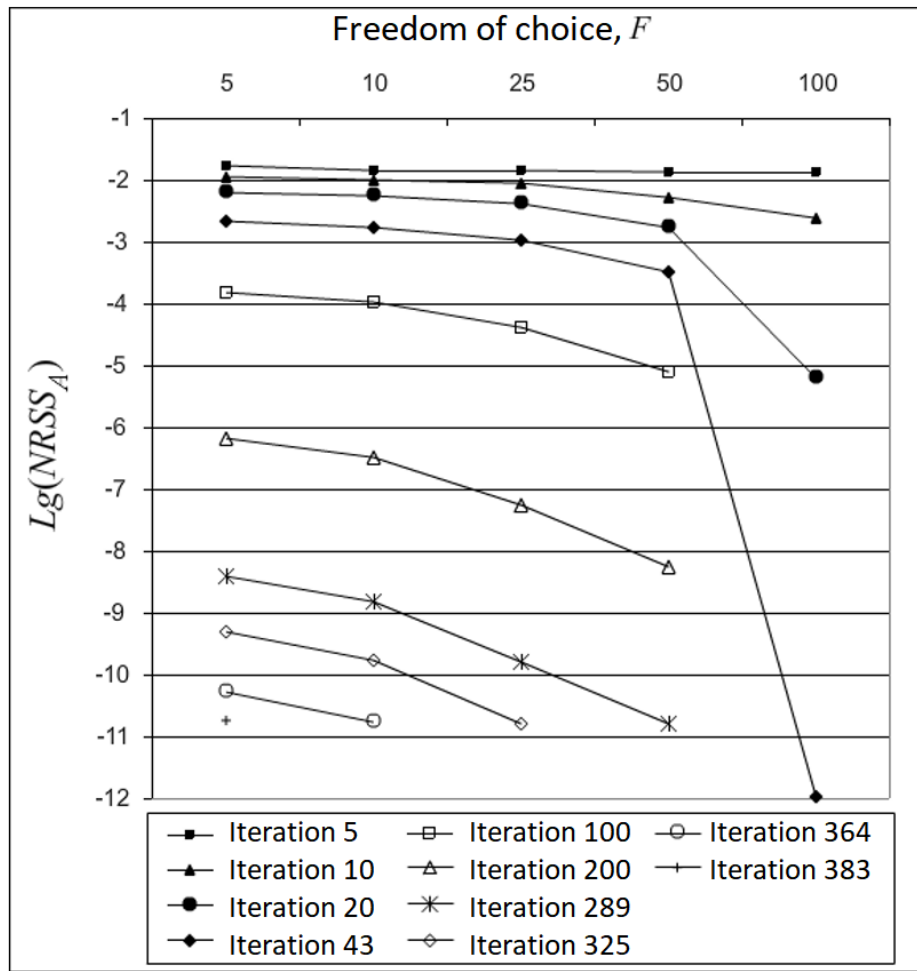


Figure 4.19 – Change in NRSS minimum with varying F parameter, model (4.7)

Figure 4.19 shows that the influence of freedom in choice F increases with increasing iteration number. This figure also clearly shows jumps in r_{prob} values as the maximum moves from the right to the left extremum of the histograms.

Therefore, when using the algorithm in practice, one should not set the parameter F too high, as there is always a threshold value F^* , exceeding which does not improve the accuracy of the model, requires more RAM and increases calculation time.

4.5 Discussion

Regarding the study of the convergence properties of the iterative algorithm, its practical value should be noted. Undoubtedly, they are very important for answering the questions: "How does the algorithm work?" and "What decisions does it obtain depending on various factors?" However, all computational experiments are somewhat artificial. While the determinant of the correlation matrix of variables can be calculated in practice, the number of arguments of the true model is generally unknown. Furthermore, given the noise immunity of modeling and the presence of unknown noise in the data, there is no 100% guarantee that the algorithm will converge to the "true" (physical) model, rather than a simplified one. We also recall that the group method of data handling, like other regression methods, is based on a very strong simplification: the object model is linear in parameters (it may be nonlinear in input variables), which limits its practical application. As we can see, there are too many "ifs" for the obtained experimental results to be applicable in practice.

However, prior to the development of this methodology, even this was not the case. Typically, algorithm research was selective: a couple of models were constructed, and conclusions about the algorithm's identifying properties were drawn based on these results, not to mention the fact that questions about convergence rate were not even considered. The results of such studies were difficult to trust.

With the advent of such computer technologies and research methods, the method for obtaining results such as the convergence rate of any iterative algorithm (not just the GMDH algorithm) and the identifying properties of any GMDH algorithm has become more substantiated and statistically reliable.

We hope that this chapter has helped researchers and scientists become familiar with the details and highlight the specifics of studying algorithm properties using computer experiments, and has helped users understand the properties of the generalized relaxation iterative algorithm. The next chapter of the book is devoted to the practical applications of GMDH algorithms for solving real-world forecasting and recognition problems.

Chapter 5

Solving practical modeling problems

In this chapter, we present three examples of solving applied problems in modeling processes of various natures. The first example is the problem of forecasting "space weather," which is characterized by a large number of observations, measured in hundreds of thousands. The next two problems relate to the field of medical diagnostics and are characterized by a small volume of initial data. The first, assessing the effectiveness of medications, is an extrapolation problem, but the resulting model essentially describes the dynamics of the human immune system over time; the second is the task of classification problem, specifically the differential diagnosis of haemostatic pathologies. We will briefly describe the relevance of each of them..

"Space weather" plays an important role in our lives and is determined by the state of the Earth's magnetic field under the influence of the solar wind. It is known that disturbances in the magnetic field have a negative impact on both electrical systems and human health [75]. An example of such a disturbance was the magnetic storm that occurred in Canada on March 13, 1989. Power was knocked out in Quebec, the largest province by population and area, leaving 6 million people without power for nine hours.

To prevent mass panic, the United States, Canada, and European countries are spending significant resources on developing magnetic storm forecasting methods. This problem remains unresolved and remains relevant. Moreover, it is a typical example of a modern, high-dimensional problem solved under conditions of significant uncertainty and incomplete information: hundreds of thousands of hourly measurements of several dozen cosmic and orbital parameters are available, for which the physical mechanism by which solar activity influences the Earth's ionosphere is unknown.

The goal of assessing the effectiveness of medications is to save money on their purchase and laboratory testing. The effectiveness of a medication for treating a cancer patient is determined by an indicator specified by an expert. This indicator is calculated based on the thiol-disulfide ratio (TDR) measured in a blood sample from the cancer patient after exposure to the medication. TDR is an indirect characteristic of the human immune system's response to medication. TDR values are measured five times per day. Monitoring TDR values is quite expensive. An individualized approach, combined with the diversity of therapeutic agents, tends to significantly increase the cost of testing when assessing the effectiveness of these interventions. Therefore, the task of reducing the number of TDR measurements and the time required to conduct medication efficacy studies is urgent.

Haemostasis pathology is a range of diseases associated with blood clotting disorders. For diagnosis, physicians rely on a number of clinical signs and their experience. However, a physician is not always able to correctly diagnose a patient due to the large number of symptoms, as well as the overlapping of symptoms for different diagnoses. Patients who cannot be confidently diagnosed by a physician may be referred for expensive laboratory chemical and biological tests, the results of which provide a definitive diagnosis. However, for many patients, such examinations are unavailable for a variety of reasons. Therefore, the task of classifying (diagnosing) patients based on clinical characteristics is urgently needed to save on laboratory testing costs and reduce the risk of misdiagnosis.

5.1 Space weather forecasting

The appearance of a magnetic storm can be identified by several geomagnetic indices, one of which is the *Dst* (disturbance storm-time) index. The *Dst* index describes changes in the northern horizontal component of the Earth's geomagnetic field and is associated with the annular flow around the Earth, which creates its own magnetic field caused by solar wind protons and electrons. If the difference between solar wind protons and electrons increases, the Earth's magnetic field weakens.

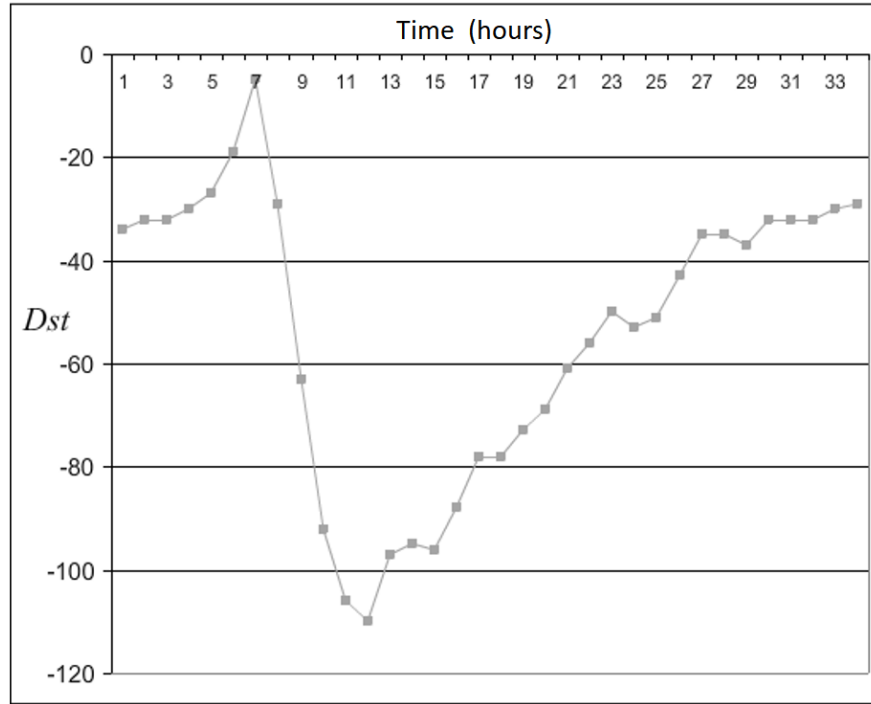


Figure 5.1 – Example of a magnetic storm based on *Dst* index values

Negative *Dst* index values indicate that the Earth's magnetic field is weakened, in other words, a magnetic storm occurring. An example of a magnetic storm is shown in Figure 5.1. The *Dst* index is measured and calculated at four stations located near the Earth's equator.

5.1.1 Description of initial data

The NASA OMNI2 Database (DB) was used for the modeling. This database is a statistical data table containing approximately 400,000 observations and 54 features. This DB has been accumulated since 1960 and contains data on solar wind parameters, the Earth's magnetosphere, and geomagnetic activity indices [91]. The OMNI2 database contains information on the following variables:

1. The set of Interplanetary Magnetic Field (IMF) parameters, $H = \{N_I, F, |B|, \alpha_B, \beta_B, x_{B,GSE(GSM)}, y_{B,GSE}, z_{B,GSE}, y_{B,GSM}, z_{B,GSM}, \sigma(|B|), \sigma(B), \sigma(x_{B,GSE}), \sigma(y_{B,GSE}), \sigma(z_{B,GSE})\}$, where:
 - N_I is the number of points for calculating the average parameters of the MMP;
 - F is the average value of the Earth's magnetic field, nT;
 - $|Q|$ is the value of the average magnetic field vector, nT;
 - α is the latitude angle of the mean magnetic field vector, degrees (GSE coordinate system);
 - β is the longitude angle of the mean magnetic field vector, degrees (GSE coordinate system);
 - $x_{B,GSE(GSM)}$ – coordinate x of the magnetic field vector, GSE and GSM systems, nT;
 - $y_{B,GSE}, z_{B,GSE}$ – y and z coordinates of the magnetic field vector, GSE system, nT;
 - $y_{B,GSM}, z_{B,GSM}$ – y and z coordinates of the magnetic field vector, GSM system, nT;
 - $\sigma(|B|)$ is the standard deviation of the mean magnetic field vector, nT;

- $\sigma(B)$ is the standard deviation of the magnetic field vector;
- $\sigma(x_{B,GSE}), \sigma(y_{B,GSE}), \sigma(z_{B,GSE})$ are standard deviations of the x, y, z coordinates of the magnetic field vector B , GSE system, nT.
- 2. Set of solar wind parameters, $\Psi = \{N_p, T, N, V, \theta, \varphi, N_a/N_p, P, \sigma(T), \sigma(N), \sigma(V), \sigma(\theta), \sigma(\varphi), \sigma(N_a/N_p), PN_1, PN_2, PN_4, PN_{10}, PN_{30}, PN_{60}\}$, where:
 - N_p is the number of points for calculating the average parameters of solar wind plasma flux;
 - T is the temperature of plasma protons, Kelvin;
 - N is the density of plasma protons;
 - V is the plasma flow velocity, km/s;
 - θ – angle along the longitude of the plasma flux, degrees (GSE systems);
 - φ – angle along the plasma flux latitude, degrees (GSE systems);
 - N_a/N_p - ratio of the number of alpha particles to the number of protons in the plasma flow;
 - P - plasma flow pressure on the Earth's magnetosphere, nPa;
 - $\sigma(T), \sigma(N), \sigma(V), \sigma(\theta), \sigma(\varphi), \sigma(N_a/N_p)$ are the standard deviations of the above parameters;
 - PN_n is the number of proton flows with energies greater than n MeV, $n \in \{1, 2, 4, 10, 30, 60\}$.
- 3. Set of transformed variables, $\Lambda = \{EF, Beta, Ma\}$, where:
 - EF – electric field, $EF = (V \cdot B_z) \cdot 10^{-3}$, mV/m;
 - $Beta$ – beta plasma, $Beta = (T \cdot 4,16 \cdot 10^{-5} + 5,34) \cdot \frac{N_p}{B^2}$;
 - Ma is the Alfvén Mach number, $Ma = \frac{V \cdot \sqrt{N_p} \cdot B}{20}$.
- 4. Set of geomagnetic indices, $\Gamma = \{Kp, Dst, AE, AU, AL, Ap, PC(N), f10.7\}$, where:
 - Kp is the planetary index or the Bartels index;
 - Dst index (Disturbance storm time index) – storm-time variation of geomagnetic activity, nT;
 - AE, AU, AL – indices developed to obtain the intensity of the auroral electric jet, nT;
- 5. Set of additional variables, $\Theta = \{Year, Day, Hour, BN, ID_1, ID_2, flag\}$, where:
 - $Year$ – the current year;
 - Day – the current day;
 - $Hour$ – the current hour;
 - BN - Barthel rotation number (identifies the number of the current 27-day solar rotation cycle);
 - ID_1, ID_2 – satellite identifiers;
 - $flag$ – a flag that determines which of the proton fluxes $PN_n, n \in \{1, 2, 4, 10, 30, 60\}$ are measured.
- 6. Solar activity index R - the number of spots on the solar limb.

While geomagnetic and solar activity indices are measured on Earth, solar wind and IMF parameters are measured by satellites in space. The OMNI2 database receives data every hour.

5.1.2 Data pre-processing

The original data table contained a significant number of missing values. This is due to two factors: firstly, satellite electronics are turned off during magnetic storms to prevent damage, and secondly, technological advances have led to the launch of more modern satellites into space, capturing a wider range of parameters. Table 5.1 presents the parameters with the fewest missing values. Before modeling, the following factors were excluded from Table 5.1:

- $Year, Day, BN, R$ – as slowly changing quantities;
- $ID_1, ID_2, N_1, N_p, flag$ – as non-informative factors;

- $y_{B, GSM}, z_{B, GSM}$ – as redundant factors, since the set of features contains the same values measured in a different coordinate system;
- $EF, Beta, Ma$ – as values calculated through other measured variables (entered into the database);
- $Kp, Ap, PC(N), f10.7$ – as values correlated with Dst .

Table 5.1 – Factors containing the least amount of missing data

Nature of factor measurement	Factors	Col.
Earth's Magnetic Field	$N_1, F, B , \alpha_B, \beta_B, x_{B,GSE(GSM)}, y_{B,GSE}, z_{B,GSE}, y_{B,GSM}, z_{B,GSM}, \sigma(B), \sigma(B), \sigma(x_{B,GSE}), \sigma(y_{B,GSE}), \sigma(z_{B,GSE})$	15/15
Solar wind	$N_p, T, N, V, \theta, \varphi, P, \sigma(T), \sigma(N), \sigma(V), \sigma(\theta), \sigma(\varphi)$	12/20
Derived Parameters	$EF, Beta, Ma$	3/3
Solar Activity Index	R	1/1
Geomagnetic indices	$Kp, Dst, Ap, PC(N), f10.7$	5/8
Additional Parameters	$Year, Day, Hour, BN, ID_1, ID_2, flag$	7/7

The variables present in the data sample fed to the input of the modeling algorithm are presented in Table 5.3.

Table 5.3 – Factors, data samples

Nature of factors	Factors	Quantity
Earth's Magnetic Field	$F, B , \alpha_B, \beta_B, x_{B,GSE(GSM)}, y_{B,GSE}, z_{B,GSE}, \sigma(B), \sigma(B), \sigma(x_{B,GSE}), \sigma(y_{B,GSE}), \sigma(z_{B,GSE})$	12/15
Solar wind	$T, N, V, \theta, \varphi, P\sigma, (T\sigma), (N\sigma), (V\sigma), \theta(\sigma), \varphi()$	11/20
Geomagnetic Indices	Dst	1/8
Additional parameters	$Hour$	1/7

5.1.3 Statement of the modeling problem

The matrix $\mathbf{X}$ is specified, $\dim \mathbf{X} = n \times m$ values of the variables $H \cup \Psi \cup \Lambda \cup \Gamma \cup \vartheta \cup \{R\} \setminus \{Dst\}$ of the set at n points of the current time with a resolution of one hour, and the vector $\mathbf{y}$ of the values of the output variable Dst index is given. It is necessary to construct the function:

$$Dst_{+1}^* = f^*(\mathbf{v}, \hat{\Theta}^*), \quad \mathbf{v} \in \mathbb{R}^{l_y + 1 + L}$$

where $\mathbf{v}$ is a vector-row of a matrix $\mathbf{H} = (\mathbf{Y} : \mathbf{X}_1 : \dots : \mathbf{X}_m)$ consisting of submatrices $\mathbf{Y} = (\mathbf{y}_0 : \mathbf{y}_{-1} : \dots : \mathbf{y}_{-l_y})$, $\mathbf{X}_i = (\mathbf{x}_{i,0} : \mathbf{x}_{i,-1} : \dots : \mathbf{x}_{i,-l_i})$; $i = \overline{1, m}$; l_y - the number of lagging values of the y variable; l_i - the number of lagged features values of the initial matrix $\mathbf{X}$, $i = \overline{1, m}$;

$$L = m + \sum_{i=1}^m l_i.$$

5.1.4 Problem Solution

The initial set of observations W was divided into working set U ($U = A \cup B, A \cap B = \emptyset$) and examination C as follows: the period of observation of the data from 01.01.1976 to 31.12.2000 formed the set U , and the period from 01.01.2001 to 22.06.2011 formed set C .

The number of lags for the Dst index and other parameters of Table 5.3 was chosen based on the Fisher test [93] and the RAM limitation: the maximum lag for Dst was 500, the maximum lag for other parameters was 24. The RAM limit was 1.3 GB. The total number of inputs (vector-columns of the H matrix) was $m = 1101$. The number of observations in the samples was $n_U = 51246$, $n_C = 86505$.

The number of magnetic storms in each of the sets of observations was counted. The magnetosphere was considered disturbed if the Dst index fell below -80nT . Set U contained 48 samples of storms, and set C contained 56 samples.

To construct a forecasting model, the *SWF* (Space Weather Forecaster) algorithm was developed, the flow diagram of which is presented in Fig. 5.10.

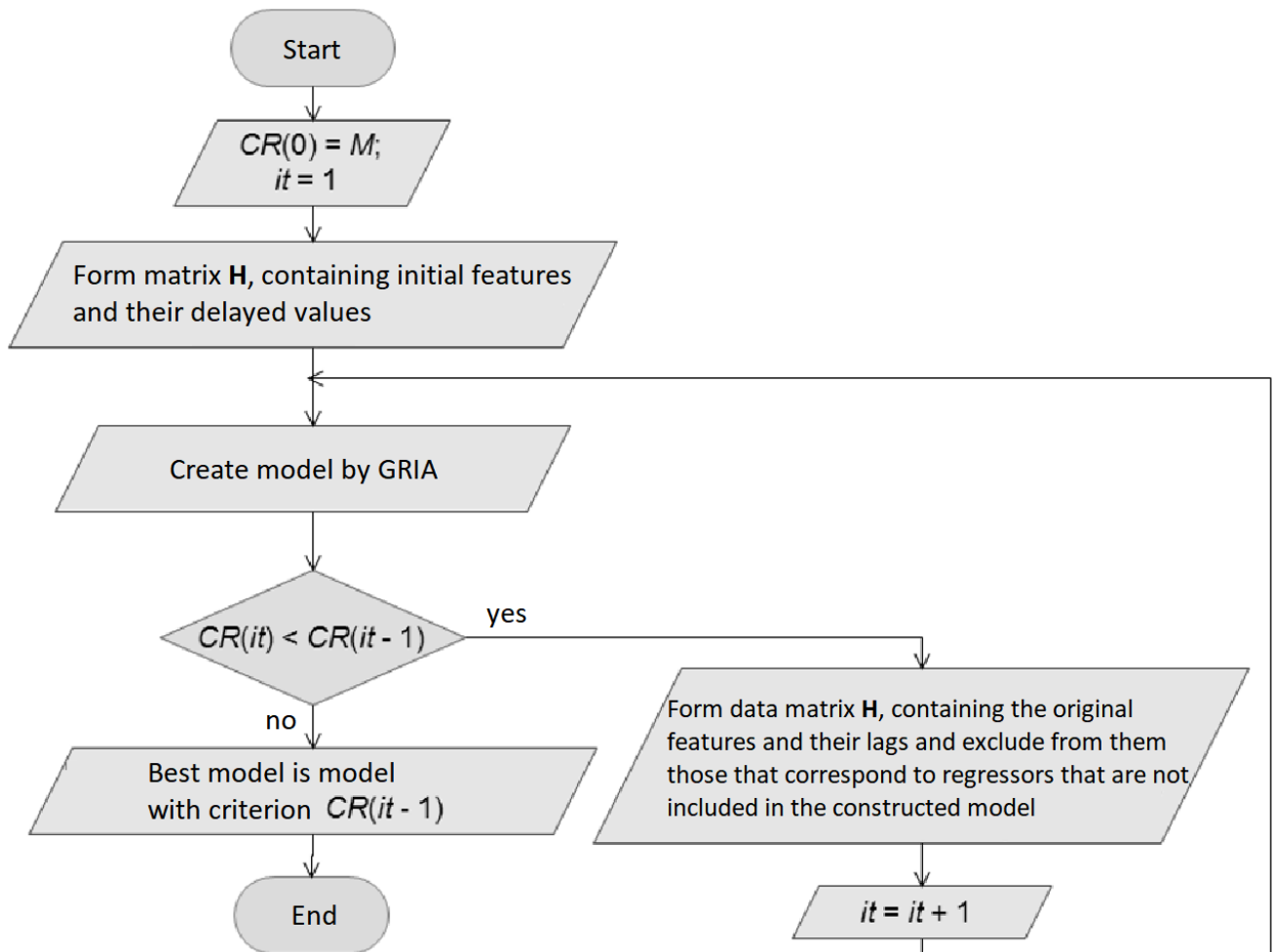


Figure 5.2 – Block diagram of the *SWF* algorithm: M - a sufficiently large positive number; $CR(it)$ - value of selection criterion for the best model

The predictive model was built using GRIA, the version of which had a structure generator with a complete sorting of models at each iteration and recurrent methods for estimating parameters and calculating the model selection criterion (RIA CS). Freedom of choice of the $F = 5$ algorithm.

Two models were built that met different rules for stopping RIA CS, namely, the shutdown was carried out on the condition that:

1. The normalised regularity criterion used as a selection criterion satisfied the inequality $NAR_{B,r+1}^* > NAR_{B,r}^*$,

$$NAR_{B,r}^* = \frac{\sum_{i=1}^{n_B} (\tilde{y}_i - \tilde{y}_{r,i}^*)^2}{\sum_{i=1}^{n_B} \tilde{y}_i^2},$$

where $\tilde{y}_{r,i}^*$ are the elements of the exit vector $\tilde{\mathbf{y}}_{r,i}^*$ of the best model of the r -th iteration.

2. An inequality was fulfilled for the difference in the values of the criteria: $NAR_{B,r}^* - NAR_{B,r+1}^* < \varepsilon$, where ε is the specified modeling accuracy, $\varepsilon = 10^{-4}$.

5.1.5 Modeling results

The linear model was obtained at the second iteration of the *SWF* algorithm according to the first rule of the algorithm and contained 186 regressors. The variables included in the model and the maximum number of lags specified in advance for them, of which only a certain number of lag variables were included in the model, are presented in Table 5.4.

Table 5.4 – Variables, their maximum lags and the number of lag variables for the first model.

Features	Dst	$Hour$	F	$ B $	α_B	β_B	$x_{B,GSE}$	$y_{B,GSE}$
Max. lag	460	15	9	1	24	24	24	20
Num. var.	39	6	2	2	9	6	10	7
Features	$z_{B,GSE}$	$\sigma(B)$	$\sigma(B)$	$\sigma(x_{B,GSE})$	$\sigma(y_{B,GSE})$	$\sigma(z_{B,GSE})$	T	N
Max. lag	13	24	21	24	23	21	6	15
Num. var.	10	8	2	13	7	5	6	7
Features	V	θ	φ	P	$\sigma(T)$	$\sigma(N)$	$\sigma(V)$	$\sigma(\theta)$
Max. lag	2	11	21	6	21	16	24	23
Num. var.	1	5	6	2	7	7	6	11
Features	$\sigma(\varphi)$							
Max. lag	16							
Num. var.	2							

The model corresponding to the second stopping rule of the *SWF* algorithm was obtained at the second iteration and looks like this:

$$Dst_{+1} = 1,52813 + 1,11965Dst_0 - 0,182017Dst_{-1} + 0,017157Dst_{-29} + 0,933523x_{1,0} - 0,26622x_{1,-1} - 0,378x_{1,-2} - 0,1585x_{2,0} - 0,161x_{3,0} + 0,06976x_{4,0} - 0,6606x_{5,0},$$

where x_1 - the z coordinate of the magnetic field vector, GSE system, $z_{B,GSE}$; x_2 - the standard deviation of the coordinate y of magnetic field vector $\sigma(y_{B,GSE})$; x_3 - the average value of the Earth's magnetic field, F ; x_4 - the density of plasma protons, N ; x_5 - the plasma flow pressure on the Earth's magnetosphere, P .

As we can see, the model contains both the parameters of the Earth's magnetic field and the parameters of the solar wind.

5.1.6 Analysis of modeling results

Statistical characteristics of the models are presented in Table 5.5, and histograms of the distribution of model errors on the working and examination sets are presented in Figures 5.6, 5.7.

Table 5.5 – Statistical characteristics of the constructed models

Characteristics		Working sample		Examination sample	
		First Model	Second Model	First Model	Second Model
Mathematical expectation, E , nT		0,0372	0,0203	-0,093	-0,125
Standard deviation, σ , nT		3,59	3,68	3,47	3,52
Average error modulus, nT		2,41	2,47	2,25	2,29
Percentage of observations whose model error fell within the interval	$[-2\sigma; 2\sigma]$	99,91	99,92	99,87	99,87
	$[-3\sigma; 3\sigma]$	99,97	99,97	99,96	99,96
	$[-5\sigma; 5\sigma]$	89,6	89,1	91	90,8

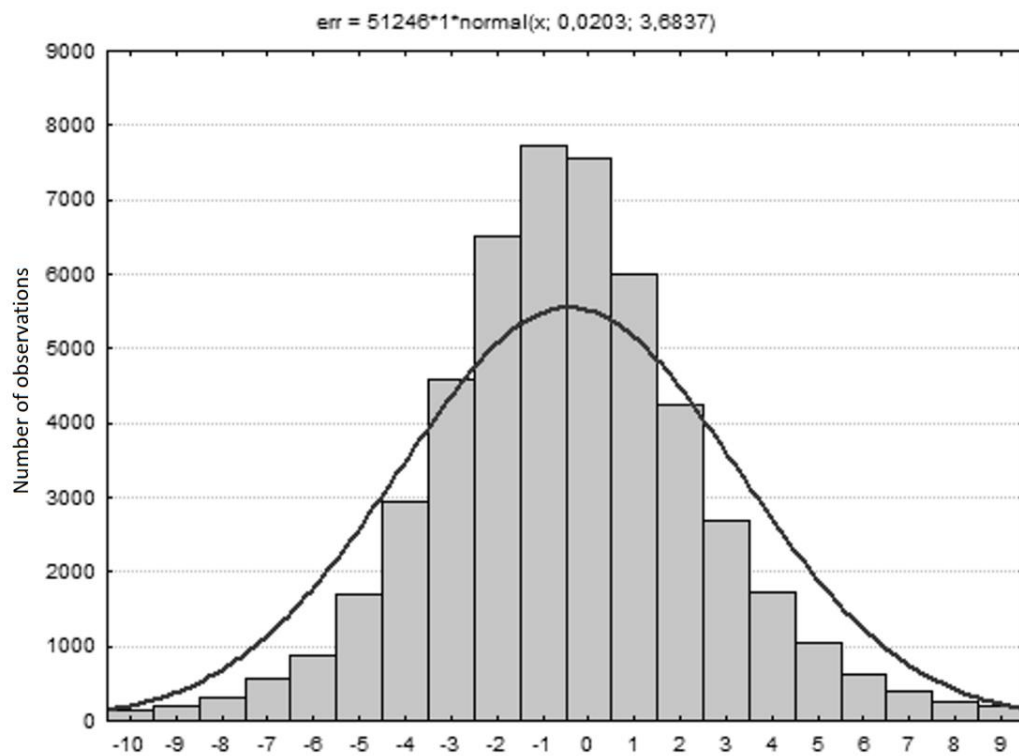


Figure 5.6 – Histogram of the model error distribution, working sample

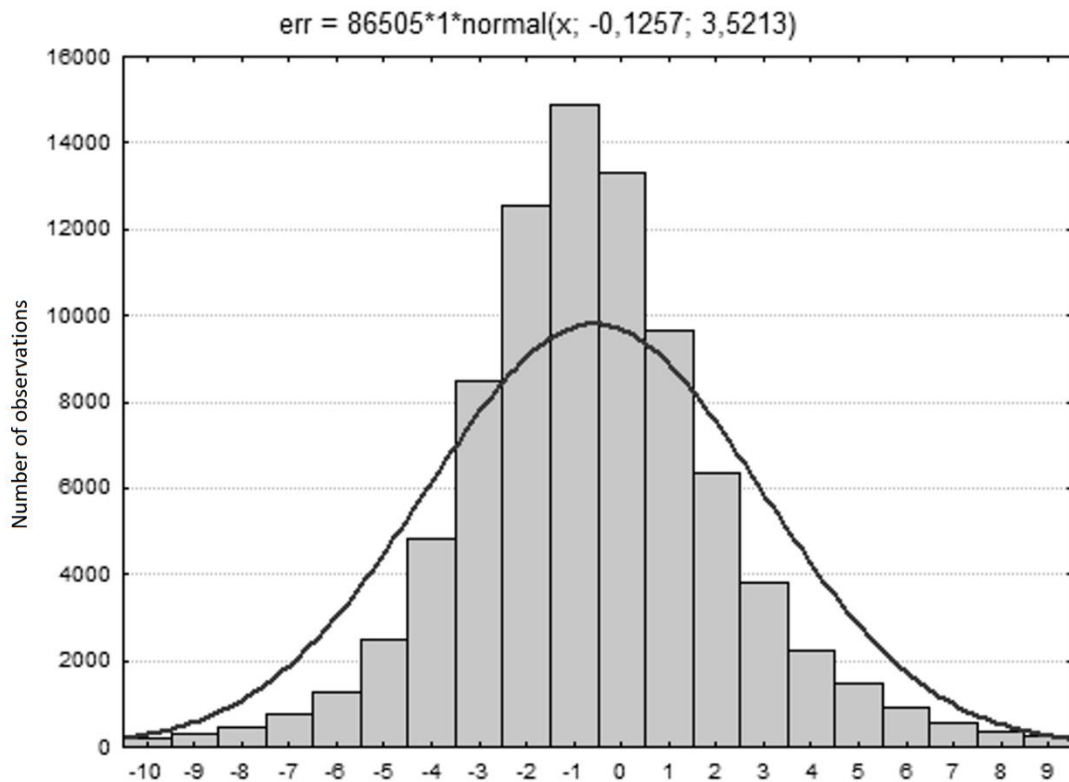


Figure 5.7 – Histogram of the model error distribution, exam sample

As can be seen from the figures, the error density functions of the model are close to the theoretical normal curve. The statistical characteristics of the model guarantee that when forecasting, the forecast error will fall within the interval $[-2\sigma; 2\sigma]$ with a probability of 0.9987, in the interval $[-3\sigma; 3\sigma]$ with a probability of 0.9996, and in the interval $[-5\sigma; 5\sigma]$ with a probability of 0.9.

The standard errors of the models on each of the samples are presented in Table 5.6, and the performance of the RIA CS in construction models is presented in Table 5.7.

Table 5.6 – Errors of the first model on each sample

Error	The first model	The second model
MSE_U, nT	3,59	3,68
MSE_C, nT	3,47	3,52
MSE_A, nT	3,31	3,38
MSE_B, nT	4,11	4,23
NAR_B, nT	0,026	0,028

Table 5.7 – Performance of RIA CS

Characteristics	First model	Second model
Algorithm running time (sec.)	620	0,05
Time to create a covariance matrix (sec.)	5	0,045
Model constructing time (sec.)	615	0,005
Number of models built	4685650	733
Number of iterations performed	4691	14

For both models, the number of good storm forecasts was calculated. We will consider good forecasts to be those for which the calculated values of the *SS* and *AP* criteria are equal to one. Otherwise, the prognosis was considered unsatisfactory.

SS criterion that determines the start of a storm:

$$SS = \begin{cases} 1, & \text{if } \frac{|Dst_{ss}^* - Dst_{bs}^*|}{|Dst_{ss} - Dst_{bs}|} \cdot 100\% > \nu, \\ 0, & \text{in opposite case} \end{cases},$$

where ν is the given threshold in percentage, which determines whether the storm is predicted by the model or not.

The *AP* criterion, which determines the correctness of the amplitude prediction:

$$AP = \begin{cases} 1, & \text{if } \frac{||Dst_{min}^{\square} - Dst_{bs}^{\square}| - |Dst_{min}^* - Dst_{bs}^*||}{|Dst_{ss} - Dst_{bs}|} \cdot 100\% > \gamma, \\ 0, & \text{in opposite case} \end{cases}$$

where Dst_{min}^* , Dst_{min} - the minimum value of the model output and the real value of *Dst* in the selected section of the time series containing the storm; γ - amplitude prediction error threshold in percentage.

An illustration of the entered criteria and their thresholds is shown in Fig. 5.8.

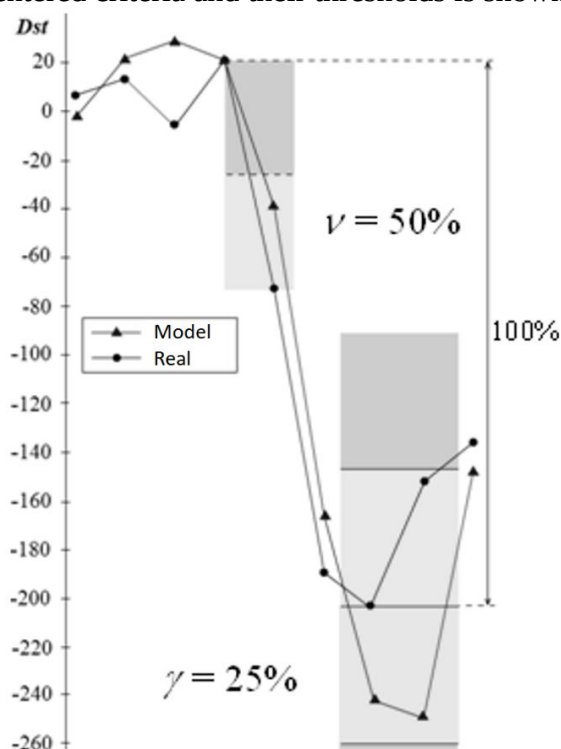


Figure 5.8 – Criteria thresholds for determining whether a storm is predicted by the model

When calculating the *SS* and *AP* criteria, the thresholds were equal, respectively: $\nu = 50\%$, $\gamma = 25\%$. The results for both models were the same and are shown in Table 5.8. The "Percentage" column of the table shows the percentage (%) of good and poor storm forecasts relative to the total number of storm samples represented in the corresponding samples.

Table 5.8 – Forecast quality

	Working sample		Examination sample	
	Quantity	Percentage	Quantity	Percentage
Good forecasts	32	66,7%	34	60,7%
Unsatisfactory forecasts	16	33,3%	22	39,3%

As we can see from Table 5.8, the model provides good forecast quality. Some of the storms predicted by the model in the exam sample are shown in Figures 5.9 to 5.11.

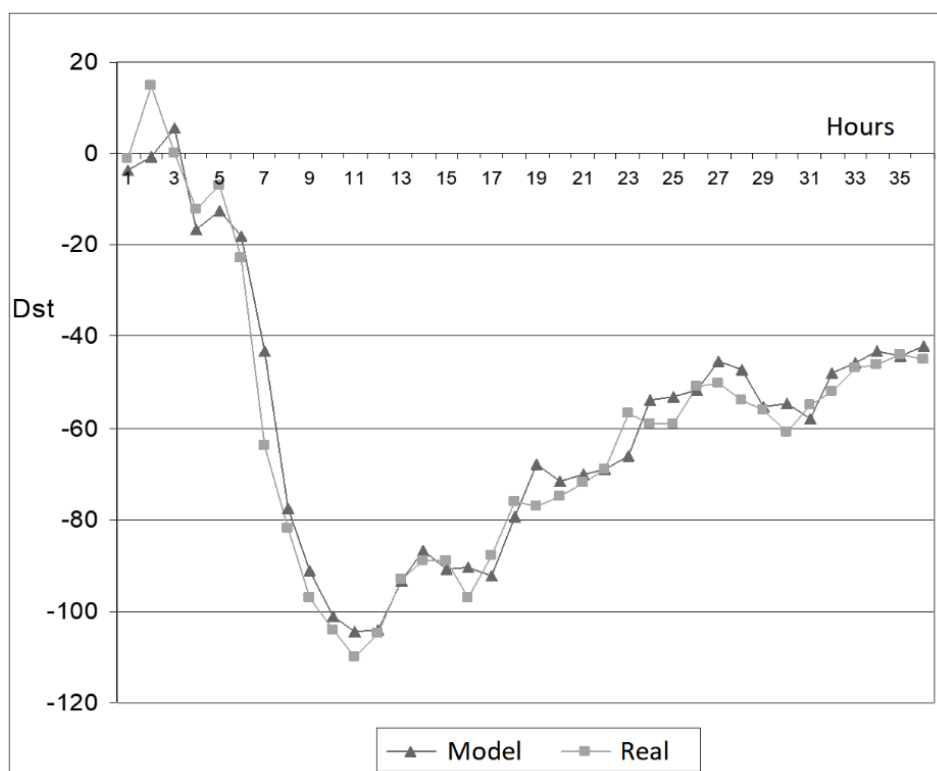


Figure 5.9 – Change in the values of the *Dst* index (modelled and real) from 11.05.2002, 09:00 to 12.05.2002, 20:00

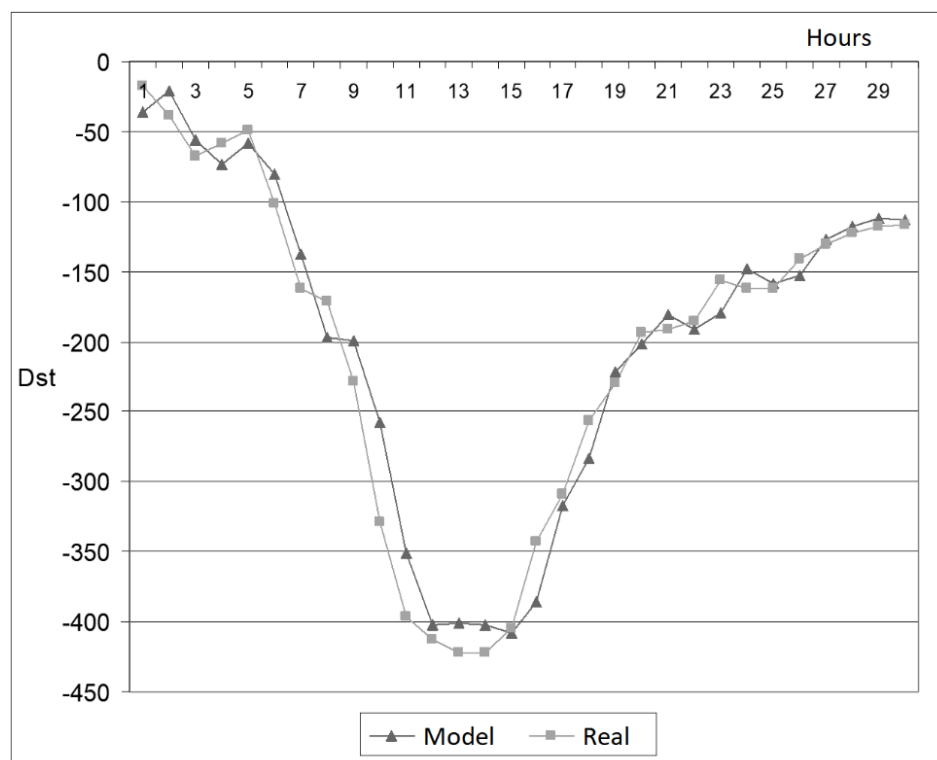


Figure 5.10 – Change in the values of the *Dst* index (modelled and real) from 15.11.2003, 08:00 to 16.11.2003, 13:00

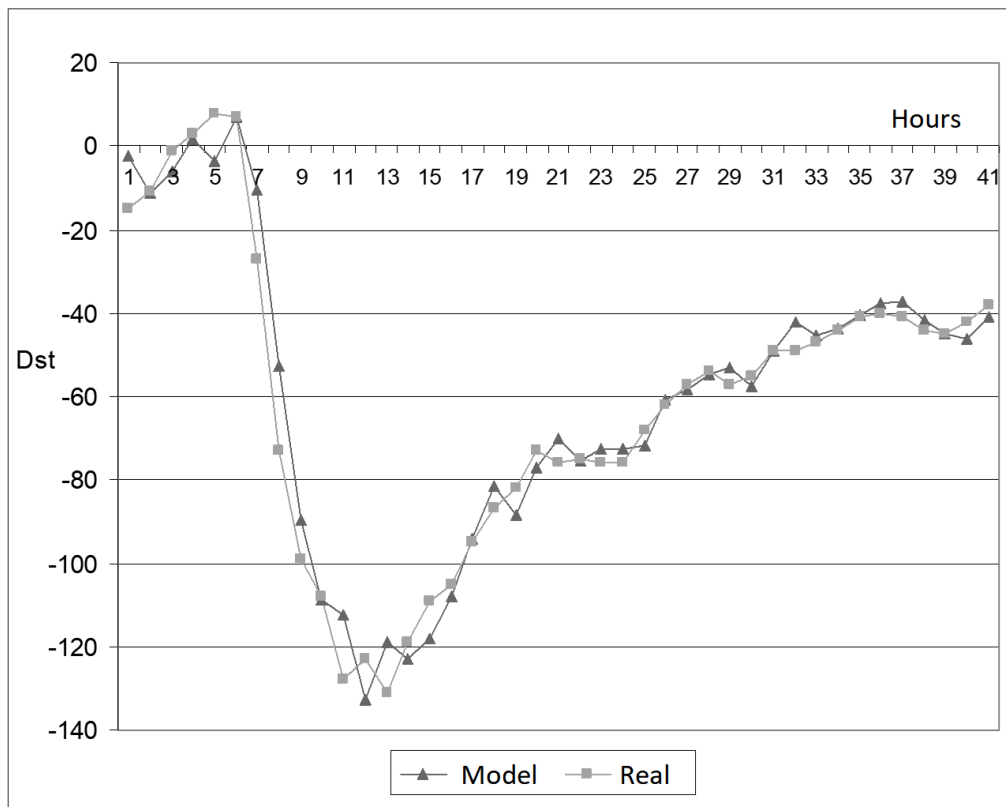


Figure 5.11 – Change in the values of the *Dst* index (modelled and real) from 31.08.2005, 07:00 to 01.09.2005, 23:00

It should be noted that storms with an amplitude greater than -300nT were not encountered in the *U* sample, but the model predicted a storm from 11/15/2003, 08:00 to 11/16/2003, 13:00 (Fig. 5.10) with an amplitude of -400nT .

5.1.7 Conclusions on the task of space weather forecasting

Taking into account the values of the statistical characteristics of the histograms of the distribution of residuals and the quality of the model forecast, we can conclude that it can be successfully used for short-term (one step) forecasting of magnetic storms.

5.2 Evaluation of the effectiveness of medications

Currently, blood samples from each cancer patient are examined when exposed to a number of drugs of varying dosages in order to find effective therapy [84]. The effectiveness of drugs is calculated based on an indirect characteristic of the resistance of the body's immune system – the thiol-disulfide ratio (TDR) [85]. TDS are determined using blood samples from a patient as a result of exposure of the sample to a specific drug at a specific dosage. TDR values are measured with a special device 5 times a day: 0 hours, 0.5 hours, 1 hour, 3 hours and 24 hours after exposure of a blood sample to a medical product (Fig. 5.12).

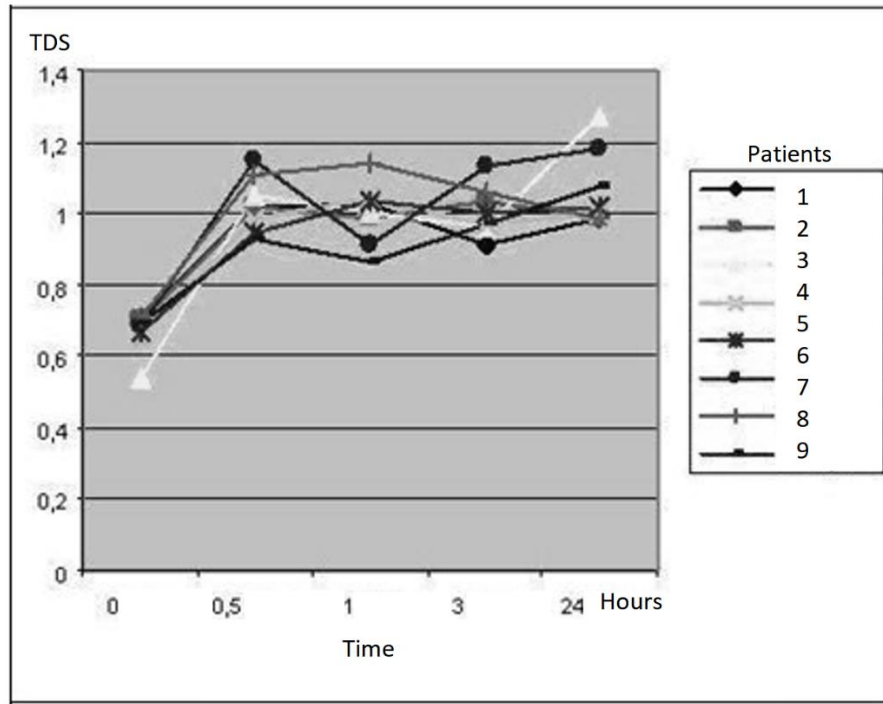


Figure 5.12 – Change in TDR value as a result of exposure to certain drug on blood samples of eight patients

The task was set to reduce the number of measurements of the TDR value in order to save time and money in the personal selection of drugs for treating a patient.

5.2.1 Problem Statement

There are many observations of W - row-vectors of the data matrix $\mathbf{Z} = (\mathbf{x}_0 : \mathbf{z}_1 : \mathbf{z}_2 : \mathbf{z}_3 : \mathbf{z}_4)$. The matrix contains the results of measuring the characteristics of TDR over time for patients tested with the same drug with the same dosage, $\dim \mathbf{Z} = n \times 5$, where n is the number of patients. The vector-columns of the $\mathbf{Z}$ matrix are: $\mathbf{x}_0$ – a vector consisting of control values of TDR in a blood sample, measured for different patients at the initial moment of change in the reaction of the blood sample to distilled water (0 hours); vectors $\mathbf{z}_1, \dots, \mathbf{z}_4$, the elements of which $z_{ij} = y_{ij}/x_{ij}$ – normalized TDR values at fixed points in the time interval of 0.5 hours, 1 hour, 3 hours and 24 hours ($j = \overline{1,4}$) and for different patients $i = \overline{1,n}$, where x_{ij} - the TDR values of the control curve (the response of a blood sample to distilled water recorded at j -th given time points); y_{ij} - the TDR values of the patient's blood sample cultured with a certain dose of the drug at the corresponding points of the time interval.

To assess the effectiveness of a medicine, the indicator set by the expert (doctor) is calculated:

$$H_i = 1 - \sum_{j=1}^4 \frac{x_{i,1} - x_{i,j}}{x_{i,1}} = 1 - \sum_{j=1}^4 (1 - z_{i,j}) = \sum_{j=1}^4 z_{i,j} - 3, i = \overline{1,n} \quad (5.1)$$

The goal is to replace the last two TDR measurements with model values, so that the value of the indicator (5.1) changes minimally, which will reduce the time for testing blood samples from 24 hours to 1 hour. So, it is necessary to construct a model for the third

$$\hat{z}_3 = f_3(x_0, z_1, z_2, \hat{\Theta}_3) \quad (5.2)$$

and the fourth

$$\hat{z}_4 = f_4(x_0, z_1, z_2, \hat{z}_3, \hat{\Theta}_4) \quad (5.3)$$

measuring TDR from the condition of minimizing the criterion:

$$Cr(\hat{\mathbf{z}}_i) = |H_i - \hat{H}_i(\hat{\mathbf{z}}_i)| \cdot 100\%, \quad i = \overline{1, n} \quad (5.4)$$

where $\hat{\mathbf{z}}_i = (\hat{z}_{i,3}, \hat{z}_{i,4})^T$ denotes the pair $\hat{z}_{i,3}, \hat{z}_{i,4}$ of calculated TDR values in the last two measurements.

Then the estimated value of the indicator:

$$\hat{H}_i(\hat{\mathbf{z}}_i) = \sum_{j=1,2} z_{i,j} + \sum_{j=3,4} \hat{z}_{i,j} - 3.$$

At the same time, it is necessary to maximize the number of patients n^* from the number of n , for which the values of criterion (5.4) will be no more than 5% when using the pair $\hat{\mathbf{z}} = (\hat{z}_3, \hat{z}_4)$:

$$\beta = \frac{n^*}{n} 100\% \rightarrow \max.$$

5.2.2 Problem solution

The problem can be solved in two ways:

The first method. Apply sequential two-criteria selection of models. The initial set W is divided into four subsets: $W = A \cup B \cup C \cup D$, where A - the training set, used to estimate the coefficients of the models; sample B is used to select the best model structures by the regularity criterion; the C set is used to select the best pair of models by the criterion β ; D - the examination set used for the final check of the adequacy of the constructed models. In accordance with the partition, the data matrix $\mathbf{Z}$ is divided into submatrices $\mathbf{Z}_A, \mathbf{Z}_B, \mathbf{Z}_C, \mathbf{Z}_D$.

The second method. The initial set W is divided into three subsets of observations: $W = A \cup B \cup C$, where A - for estimating the coefficients of the models; B - to select the best model structures according to the criterion β ; C - the examination set used for the final check of a pair of models on "fresh" data.

This example illustrates the solution of the problem in the first way. To improve the modeling results we propose to apply:

1. Algorithm for partitioning the set W into A, B, C .
2. Methodology for constructing a forecasting model as the sum of a trend component and a residuals model.
3. Construct a model for the fourth observation with adaptive forecast (see subsection 1.1.2 for definition).

The partitioning of the set W into working $X = U \cup C = A \cup B \cup C$ and examination D is carried out randomly. Sets A, B, C, D are pairwise mutually disjoint.

To solve this problem, we use a modeling algorithm with three-stage partitioning (MATP) for the set X of observations. We denote the cardinalities of the sets X, D, U, A, B , and C as n_X, n_D, n_U, n_A, n_B and n_C ; $n_X = n_A + n_B + n_C$, $n_U = n_A + n_B$.

The paper compares two schemes of MATP operation presented in Fig. 5.13.

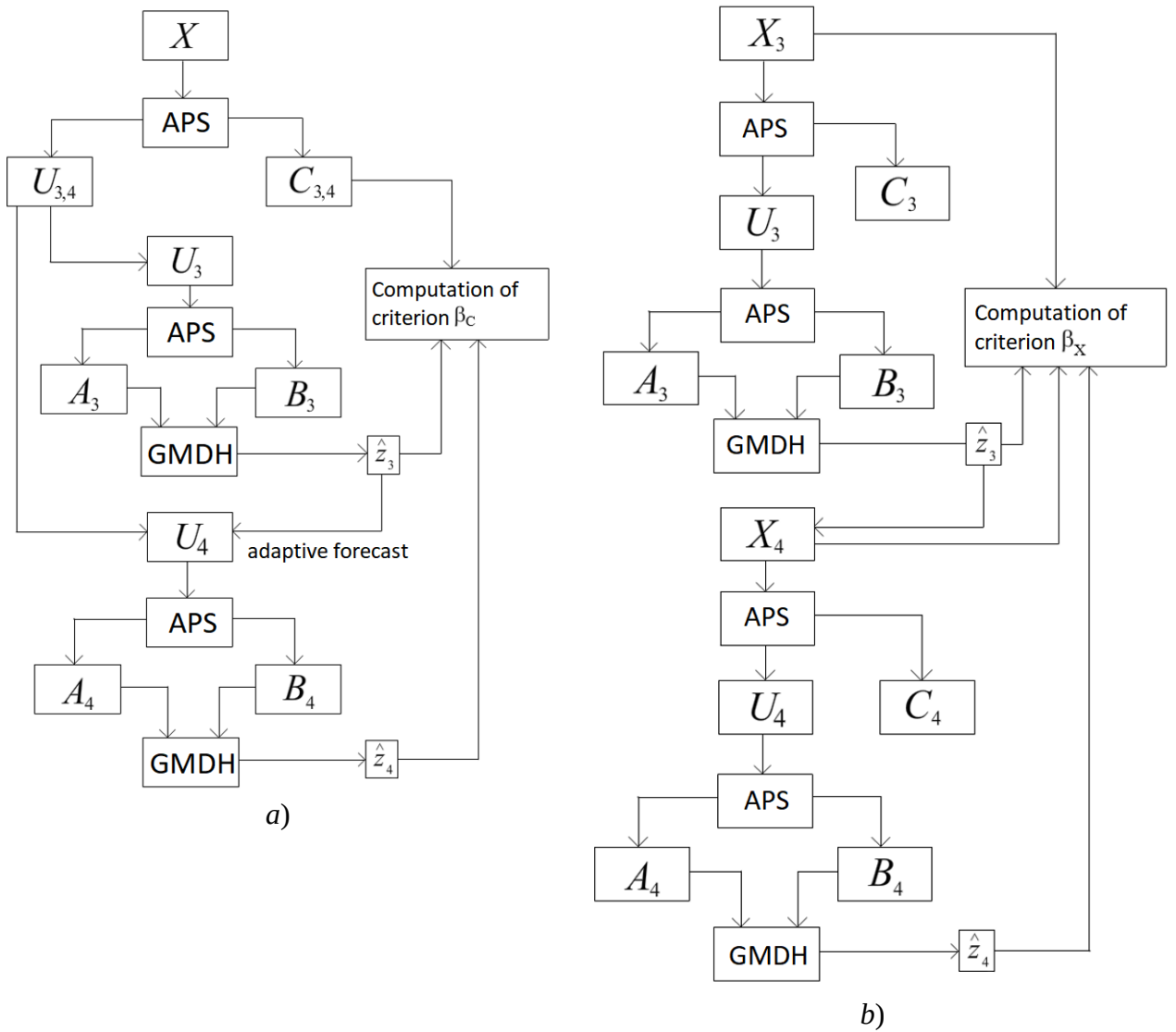


Figure 5.13 – MATP schemes for finding the best pair of models:

APS – algorithm for partition the sample of observations;

GMDH – model construction algorithm

The main difference is that in scheme *a*) both models are built on the same set $U_{3,4}$ (the subscripts of the sets indicate the numbers of the models, for which they are used to construct). Since in scheme *b*) models can be built on samples C_3 and C_4 with different lengths, the best pair is not chosen according to criterion β_C :

$$\beta_C = \frac{n_C^*}{n_C} \cdot 100\%,$$

but by the criterion

$$\beta_X = \frac{n_X^*}{n_X} \cdot 100\%.$$

We also examine the case where in scheme *a*) the criterion is not β_C , but β_X , and this scheme is designated *c*).

The goal of MATP is to maximize the β_C (β_X) criterion by sorting the values of the following parameters:

- sample lengths $n_{U_3}, n_{U_4}, n_{A_4}, n_{A_4}$;
- the composition of samples $U_3, U_4, A_3, A_4, C_i = X \setminus U_i, B_i = U_i \setminus A_i, i = 3, 4$. The sample partition index coincides with the index of the model for which it is used. The optimization algorithm

consists of a complete sorting of sample length parameters values $n_{U_3}, n_{U_4}, n_{A_4}, n_{A_4}$ within range:

$$n_{U_3} = n_{U_4} = \overline{n_{U_{4,\min}}, n_{U_{4,\max}}}, \quad n_{A_3} = n_{A_4} = \overline{n_{B_{4,\min}}, n_{B_{4,\max}}}.$$

The following sampling methods described in subsection 2.6.1 were investigated:

- 1) "Similar" by Variance Partition (SVP) [49];
- 2) "Dissimilar" by Variance Partition (DVP) [49];
- 3) Quasi-optimal partitioning (QOP) [50].

Methodology for constructing a forecasting model as a sum of a trend component and a residual model. It is proposed to represent the models $\hat{z}_3, \hat{z}_4$ as a sum of the trend model $\hat{z}_{trend}$ and the models of residuals $\hat{z}_{3,res}, \hat{z}_{4,res}$. The trend model has the form:

$$z_{trend}(t, \Theta) = \sum_{j=1}^p \theta_j \psi(t), \quad (5.5)$$

where $t = \{1, 2, 3, 4, 5\}$ is discrete time; $\psi(t): \mathfrak{R} \rightarrow \mathfrak{R}$ is a power function of t from a given set of functions $\mathfrak{F}$. After obtaining a trend model $\hat{z}_{trend}^*$, a matrix of the output values of this model is constructed $\hat{\mathbf{Z}}_{trend}^* = (\hat{\mathbf{x}}_{trend,0}^* : \hat{\mathbf{z}}_{trend,1}^* : \hat{\mathbf{z}}_{trend,2}^* : \hat{\mathbf{z}}_{trend,3}^* : \hat{\mathbf{z}}_{trend,4}^*)$, $\dim \hat{\mathbf{Z}}_{trend}^* = \dim \mathbf{Z}$. Next, the 'residuals' matrix is calculated $\mathbf{Z}_{res} = \mathbf{Z}_5 - \hat{\mathbf{Z}}_{trend}^*$. Residuals models $\hat{z}_{3,res}, \hat{z}_{4,res}$ are constructed in the form (5.2), (5.3) where the matrix $\mathbf{Z}_{res}$ is used instead of matrices $\mathbf{Z}$.

In addition to the three research directions mentioned above, when partitioning the set of observations $\mathbf{X}$, the matrix $\mathbf{Z}$ containing complete and incomplete information was fed to the MATP input. Completeness of information is characterized by the set of vector-columns of the matrix. Complete information corresponds to the presence of all vector-columns: $\mathbf{x}_0, \mathbf{z}_1, \mathbf{z}_2, \mathbf{z}_3, \mathbf{z}_4$, incomplete – $\mathbf{x}_0, \mathbf{z}_1, \mathbf{z}_2$. Models obtained from data containing incomplete information are also of interest, since when using a pair of models in practice, only the first three measurements of TDR will actually be carried out.

Thus, for each of the three proposed schemes of MATP (a), b), c)) we have the following areas of research:

- Completeness of information in the initial data sample: complete and incomplete information;
- Models construction methodology: with or without a trend model;
- Variants of model construction for the fourth point: with or without adaptive forecast;
- Sample partitioning algorithm: SVP, DVP, QOP.

Summing up, we get a total of 72 different options for research directions.

As it was shown in subsection 3.3.4, under the condition of a small number of data points and a small number of arguments, the GRIA and MSA algorithms are comparable in terms of performance and with some ratios of small sample parameters, GRIA is inferior to MSA. Therefore, as the GMDH algorithm was used a modified version of MSA – MAKSO [66], equal to it in computational complexity, but having a user-friendly interface. The class of models and the construction procedure in MAKSO and MSA are identical. The set of functions $\mathfrak{F}$ used to construct the trend model in the MAKSO algorithm is as follows: $\left\{ \frac{1}{t}, \frac{1}{\sqrt[3]{t}}, \sqrt[3]{t}, 1, t, t^2, t^3 \right\}$.

The set of cases contained $n = 28$ blood samples of patients tested with immunofan at a dosage of 100 ml. The initial set W was randomly divided into working set X , $n_X = 18$, and examination set D , $n_D = 10$.

5.2.3 Modeling results

The constructed trend model has the following form:

$$\hat{z}_{trend}^*(t) = 1.03559 - 0.361196 \frac{1}{t^3}.$$

The best pair of residual models with adaptive forecasting has the following form:

$$\begin{aligned}\hat{z}_{3,res} &= -0.0701731 + 6,5202929 \cdot z_1^2 + 306x_0z_1^3 - 0.142217z_1, \\ \hat{z}_{4,res} &= 0.0042689 - 4946x_0z_1z_2\hat{z}_{3,res} - 0.59237x_0 + 0.5184115 \cdot \hat{z}_{3,res}\end{aligned}$$

General conclusions across all research areas when testing models on the working set X :

- MATP schemes $b)$ $\beta_X = 65\%$ and $c)$ $\beta_X = 62\%$ turned out to be more accurate than schema $a)$ $\beta_X = 51\%$;
- Methodology of construction with trend model allocation leads to more accurate results;
- Scheme with adaptive prediction improves the accuracy of modeling;
- Quasi-optimal partitioning makes it possible to obtain more accurate models than models under random (sequential), SVP and DVP partitions.

The study options in which the best models for each of the MATP schemes were obtained are shown in Table 5.9.

Table 5.9 – Best models obtained from studies

Scheme	Completeness of information	Partitioning algorithm	Prediction scheme	Method of construction a model	Criterion	Points number
$a)$	Complete	QOP	Adaptive	Trend pattern	$\beta_C = 87.5\%$	$n_C = 8$
$b)$	Incomplete				$\beta_X = 77.8\%$	$n_X = 18$
$c)$	Complete					

General conclusions across all directions of research when testing models on the examination set D :

- Scheme $c)$ $\beta_D = 23.8\%$ was found to be more accurate than schemes (a) $\beta_D = 20\%$ and (b) $\beta_D = 20\%$;
- The conclusions regarding the model construction methodology, the model construction scheme for the last measuring point, and the sampling partitioning algorithm coincide with the conclusions obtained from sample X .

The study options in which the best models for each of the MATP schemes were obtained are shown in Table 5.10.

Table 5.10 – The best models on the examination set

Scheme	Completeness of information	Partitioning algorithm	Prediction scheme	Method of construction a model	Criterion
$a)$	Incomplete	QOP	Adaptive	Trendless model	$\beta_D = 30\%$
$b)$			Common	Trend pattern	$\beta_D = 30\%$
$c)$			Adaptive		$\beta_D = 60\%$

From these results, it can be concluded that the best variant of the MATP partitioning scheme corresponds to variant c), i.e. when the models are built on the same set $U_{3,4}$, and the selection of the best pair of models is carried out according to criterion β_X . The best pair of models $\hat{z}^* = (\hat{z}_3^*, \hat{z}_4^*)$ is obtained with a quasi-optimal partition of the entire sample $U_{3,4}$, the adaptive forecast scheme and model with a trend. Let's perform a statistical analysis of the residuals for the best pair of models.

5.2.4 Analysis of modeling results

The task was to determine such a model $\hat{z}^* = (\hat{z}_3^*, \hat{z}_4^*)$ in which as many values of the (5.4) criterion $\varepsilon_j, j = 1, \dots, n$ as possible fell into the interval $[-0.051; 0.051]$, with a significance of no more than 0.05.

Histograms of the error distribution for the best pair of models $\hat{z}^*$ in samples X, W and D are shown in Figure 5.14.

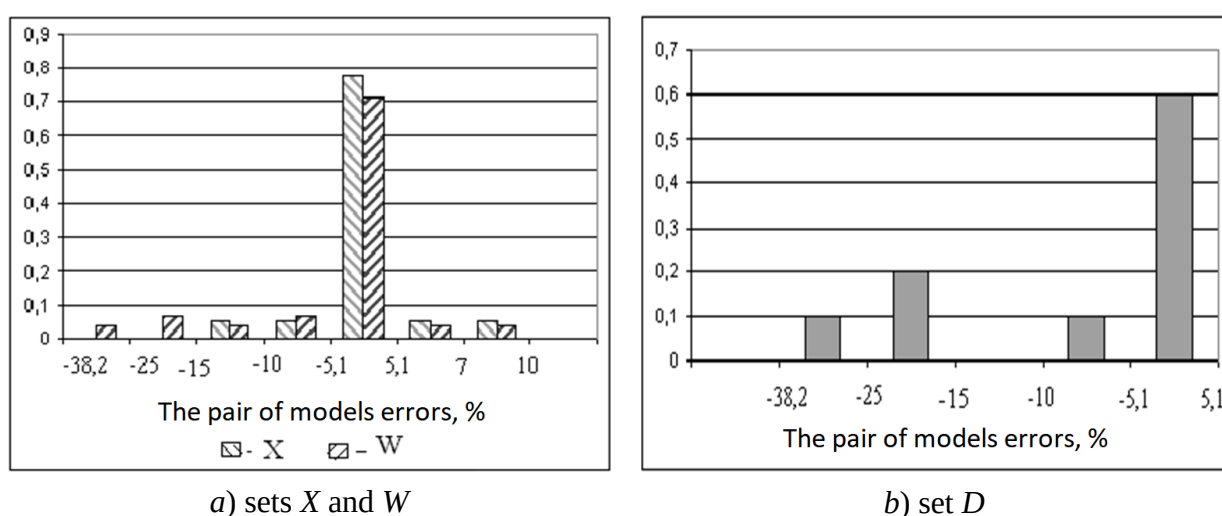


Figure 5.14 – Histograms of the error distribution for the pair of models

In the graph in Fig. 5.14 (a), the relative frequency functions (a discrete version of the density function) on both sets W and X are close to the normal law. Due to the asymmetric distribution of errors on the examination set D in the region of negative values (see Fig. 5.14 (b)), the relative frequency function on the original set W is also asymmetric.

As a rule, no compelling reason can be given why the distribution of the results of processing specific observations should belong to a particular family of distributions. Exceptions are known: if a probabilistic model involves the summation of independent random variables, then the sum is described by a normal distribution; if, however, the model considers the product of such variables, then the result approaches a log-normal distribution [86].

The histogram of the error distribution on the working set X is close to the normal distribution, and on the entire set W it looks like a mirror image of the log-normal distribution (see Figure 5.14 a)). However, in order to generalize the conclusions about the predictive properties of the pair of models to the entire population using nonparametric hypothesis testing criteria, the set of observations W has insufficient power [87].

Statistical characteristics of the pair of models $\hat{z}^*$ are presented in Table 5.11, the analysis of which shows that for the absolute majority of patients, the errors of the pair of models $\hat{z}^*$ on all sets X, D and W fell within the interval $[-0.051; 0.051]$ (see the criterion β in Table 5.11). Model errors on the set W fell into the interval $[-3\sigma; 3\sigma]$ with a probability of 0.96 and a significance not lower

than $\alpha = 0.04$, i.e. the main distributed mass was within the specified interval, and only 1 point out of 28 was outside it. From subsets X and D , 100% of the points (model errors, validated against patient data from X and D) fell into the $\pm 3\sigma$ interval. The resulting best model for this dataset is reliable and applicable for practical purposes.

Table 5.11 – Statistical characteristics of the best pair of models

Characteristics	Working subset X	Examination subset D	Original set W
Standard deviation σ , %	5,32	13,08	9,52
Value 3σ , %	15,95	39,24	28,57
Criterion β , %	78	60	71
Percentage of points at which the error fell within the interval $[-3\sigma; 3\sigma]$, %	100	100	96

5.2.5 Conclusions on the task of assessing drug's efficacy

Based on the analysis results presented in Figure 5.14 and Table 5.11, it can be concluded that the constructed pair of models can be used in the decision-making system to guide an expert in assessing the efficacy of Immunofan. However, a larger number of observations is required to obtain statistically reliable conclusions about the model.

5.3 Differential diagnosis of haemostasis pathologies

Haemostasis (from the Greek *haima* – blood, *stasis* – stop) is a concept that includes cessation of bleeding. Pathology is a painful deviation from the norm.

Haemostasis pathologies include von Willebrand disease (WD), coagulopathies (CP), disaggregating thrombocytopathy (DT), and combined pathology of the haemostasis system (CPHS) [88]. All of these are associated with incoagulability of human blood. The prevalence of thrombocytopathy, for example, has not been established, but it is believed that this disease can be diagnosed in 5-10% of the population [89, 90]. When making a diagnosis, doctors rely on the number of haemorrhagic signs and their experience. However, it is quite difficult for a doctor to reliably determine a diagnosis for many reasons, the main one being the overlap of combinations of signs in various diseases. Furthermore, the presence of haemorrhagic signs is not always associated with the disease.

In a survey of 1,000 Kyiv residents, 65.2% had a history of haemorrhagic syndrome. When 65 of the respondents visited the Department of Surgical Haematology and Haemostasis of the Institute of Haematology and Transfusiology of the Academy of Medical Sciences of Ukraine, clinical and laboratory tests revealed that 38.0% had von Willebrand disease; 32.0% had disaggregating thrombocytopathy; 18.0% had coagulopathy; and 12.0% had a combined pathology in the haemostatic system [88].

5.3.1 Problem statement

A set of observations $W \subset \mathcal{R}^{m+1}$ is given, the elements of which are vector-strings of the matrix $\mathbf{W} = (\mathbf{X}:\mathbf{y})$, $\dim \mathbf{X} = n_W \times m$, m - the number of features. The vector $\mathbf{y}$ contains labels belonging to one of the k classes in the form of given numbers y_i , $i = \overline{1, n_W}$. For example, labels can be numbers classes.

The goal is to build a classifier that can effectively recognize patients both in the existing set $\mathbf{W}$ and in a new set not used in its construction.

5.3.2 Problem solving concept

The problem solving concept consists of constructing k decision rules (decision functions), each of which assigns an observation to the class for which it (the rule) was constructed, i.e., according to the "one versus all" rule. To construct each classifier, a set of observations W_j , $j = \overline{1, k}$ is formed from the initial set W as follows. The set W_j corresponds to the row vectors of the matrix $\mathbf{W}_j = (\mathbf{X}; \mathbf{y}_j)$, where $\mathbf{y}_j$ contains two values: p – the assignment of the observation to the j -th class; q – the assignment of the observation to the remaining classes.

A type I error ("missing a target") occurs if the classifier misses its own patient (responds "-" instead of "+"). A type II error ("false alarm") occurs when the classifier mistakes a foreign patient for its own (responds "+" instead of "-").

Obviously, the following situations occur in this context:

- 1) a classifier responded "+" to its own patient, while all other classifiers responded "-";
- 2) all classifiers responded "-";
- 3) two or more classifiers responded "+" while the other classifiers responded "-";
- 4) a classifier responded "-" to its own patient, one of the other classifiers responded "+", while all other classifiers responded "-".

The first situation is a case of correct recognition by the system. In cases 2) and 3), there is a conflict between classifiers, and the system cannot provide a definitive answer, since 2) results in a Type I error, and 3) results in a Type II error. Cases 2) and 3) represent a failure to classify. In the fourth case, Type I and Type II errors occur in concert on only two classifiers – this is a case of misclassification.

Each of the formed sets W_j is divided into two: working U_j and examination C_j , $U_j \cap C_j = \emptyset$. Sets U_j are used to construct classifiers, and C_j sets are used to verify the correct recognition of "new" observations.

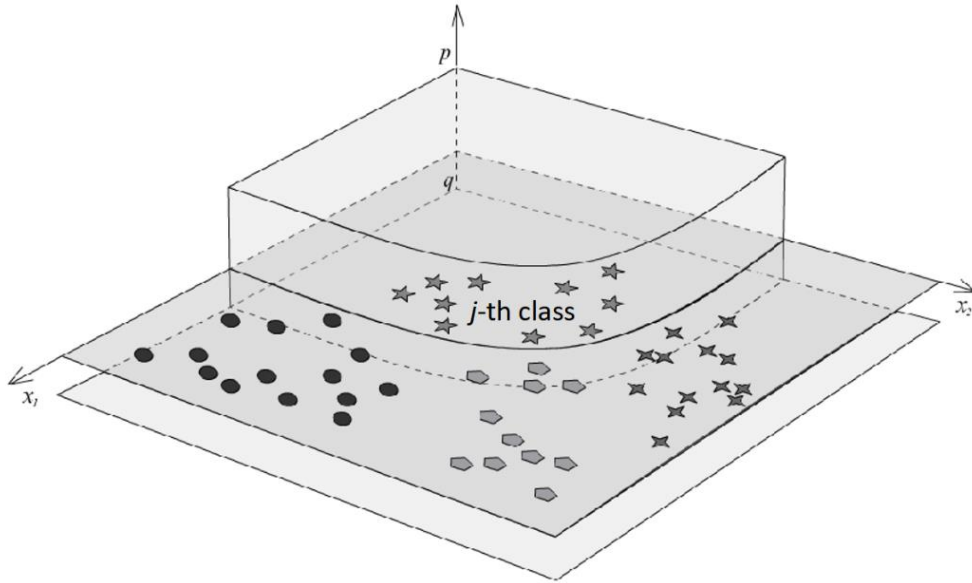


Figure 5.15 – Approximated planes

The decision rule of the j -th classifier is based on the construction of approximating function $y_j(\mathbf{v}) = f_j(\mathbf{v}, \Theta_j): \mathbb{R}^m \rightarrow \mathbb{R}$, $\mathbf{v} \in \mathbf{W}$, separating points of one class from points of all other classes (Θ_j is the parameters vector). In this case, we have a modeling problem where the output variable takes two

discrete values: the value p indicates that the observation belongs to the class for which the classifier is being constructed; the value q determines whether the observation belongs to other classes (see Figure 5.15). After constructing the functions $\hat{y}_j^*(\mathbf{v})$, based on the considerations of minimizing the number of errors on the working set U_j , a recognition threshold is introduced Π_j . The decision function of the j -th classifier is the value, $Q_j(\mathbf{v}) = \hat{y}_j^*(\mathbf{v}) - \Pi_j$, $\mathbf{v} \in \mathbf{W}$ and if $Q_j > 0$ ("+"), then the observation $\mathbf{v}$ belongs to this class, otherwise the answer "-", i.e. does not belong.

5.3.3 Solving the problem

The following twelve clinical features are specified in the task:

- x_1 – nosebleed;
- x_2 – bleeding gums;
- x_3 – bleeding after tooth extraction;
- x_4 – intra- and postoperative bleeding;
- x_5 – posttraumatic hematoma;
- x_6 – bleeding from superficial wounds;
- x_7 – long-term non-healing wound;
- x_8 – post-traumatic hemarthrosis;
- x_9 – post-injection hematoma;
- x_{10} – postpartum haemorrhage;
- x_{11} – juvenile uterine bleeding;
- x_{12} – age.

As a rule, the signs take binary values "yes" or "no", but for some patients a third value is introduced – "there were no conditions for the manifestation of this sign". For example, the sign "postoperative bleeding" could not be observed if the patient did not have surgery. The clinical features corresponding to their presence is assumed to be equal 25, the absence of a feature - to the value (-5), and the absence of conditions for manifestation of the feature is assigned a value of 1.

The above values of the features are taken quite arbitrarily, but in relation to each other: the value of the attribute "the absence of conditions for manifestation of the feature " is taken as "1" for the reason that when constructing nonlinear models (models with multiplicative arguments), such a value of the argument included in the multiplicative argument will not affect the value of the function $\hat{y}_j$. The level of acceptance of "own" diagnosis can be chosen arbitrarily and in this case is equal to $p = 220$, and all other $q = 100$.

To build classifiers, the largest group of patients was selected – women aged from 19 to 49 years. The total number of patients in the group $n_W = 80$. Sets U_j are randomly (sequentially) divided into training A_j and validation B_j . $j = \overline{1,4}$. The first is used to estimate the parameters Θ of the functions f_j , the second is used to select the structure of functions according to the criterion of regularity.

The number of observations in the training A_j , validation B_j , and examination C_j sets for each diagnosis $j = \overline{1,4}$ is presented in Table 5.12.

Table 5.12 – Number of points in samples: $n_{G,d}$ – number of patients with a corresponding diagnosis in a set of patients G ; $n_{G,od}$ – number of patients with other diagnoses in a set of patients G , $G = \{A, B, C\}$

Diagnosis	Training			Validation			Examination		
	$n_{A,d}$	$n_{A,od}$	n_A	$n_{B,d}$	$n_{B,od}$	n_B	$n_{C,d}$	$n_{C,od}$	n_C
WD	17	45	62	3	5	8	4	6	10
CP	13	49	62	2	6	8	2	8	10
DT	24	36	60	4	6	10	4	6	10
CPHS	8	52	60	0	10	10	0	10	10

Due to the small sample size for the diagnosis of CPHS, all 8 points were left in the classifier calculation in training sample, and 0 points were retained for validation and examination.

5.3.4 Modeling results

The results of constructing classifiers using the described technology are as follows. Approximating functions of the classifiers:

1. Von Willebrand disease:

$$\begin{aligned}\hat{y}_1 = & 108.47 - 440.78 \frac{1}{x_4 x_5 x_{10}} + 1.656 \frac{x_5 x_{11}}{x_9 x_6} + 112.06 \frac{x_7}{x_3 x_9 x_8} + 1.365 \cdot 10^{-4} x_4 x_1 x_{10}^2 + \\ & + 6.62 \cdot 10^{-4} \frac{x_{12}^2 x_{11}}{x_4} - 0.455 \frac{x_6^2}{x_1 x_4} + 1.007 \cdot 10^{-4} x_{12} x_2 x_{10} x_7 - 0.045 \frac{x_{12}^2}{x_6 x_4} \\ & + 5407.58 \frac{1}{x_{12} x_5 x_{10} x_3} - 0.065 \frac{x_4 x_6 x_{10}}{x_{12}}.\end{aligned}$$

2. Coagulopathy:

$$\begin{aligned}\hat{y}_2 = & 98.79 - 41.47 \frac{x_{11}}{x_3 x_{12}} + 1037.1 \frac{1}{x_{12}} - 0.527 \frac{x_9 x_5}{x_4 x_7} + 109.4 \frac{x_3}{x_1 x_6} - 0.89 x_{10} + 0.0012 \frac{x_{12}^2 x_3}{x_{10}} - \\ & - 0.052 \frac{x_3 x_{12}}{x_{10} x_4} + 111.9 \frac{1}{x_2 x_4} + 0.89 \frac{x_{10}}{x_3} + 0.446 \frac{x_3 x_4}{x_2 x_{10}}.\end{aligned}$$

3. Disaggregation thrombocytopathy:

$$\begin{aligned}\hat{y}_3 = & 146.2 + 428.57 \frac{1}{x_{10} x_5} - 10.38 \frac{x_1^2}{x_5 x_{12}} + 1275 \frac{1}{x_9 x_7 x_3^2} + 0.043 x_1 x_3 - 1.708 \cdot 10^{-4} x_5 x_4 x_2 x_3 + \\ & + 37.58 \frac{x_6}{x_7 x_9} - 0.076 \frac{x_9 x_5}{x_4} + 849 \frac{x_2}{x_1 x_{12} x_9} + 7326 \frac{1}{x_1 x_4 x_2^2} - 2.46 \frac{x_{10} x_{11}}{x_3 x_{12}} + 4.29 \frac{x_4}{x_{10} x_6}.\end{aligned}$$

4. Combined pathology of the haemostasis system:

$$\begin{aligned}\hat{y}_4 = & 94.5 + 1456 \frac{1}{x_1 x_2} - 32.7 \frac{x_6}{x_{10} x_2 x_3} + 0.0016 x_4^2 x_6 - 3418 \frac{1}{x_{12} x_3 x_9} - 0.018 \frac{x_{12} x_7 x_4}{x_1} + \\ & + 0.062 \frac{x_4 x_{10} x_2}{x_{12}} + 652 \frac{x_1}{x_5 x_{12} x_7} - 0.11 \frac{x_6 x_3}{x_7} - 0.011 \frac{x_{12}^2}{x_3} - 48.5 \frac{x_3}{x_1 x_2 x_4}.\end{aligned}$$

Graphs of the approximating functions of classifiers are shown in Figures 5.16 - 5.19, where the dotted line represents the recognition thresholds: $\Pi_1 = 150$, $\Pi_2 = 145$, $\Pi_3 = 159.5$, $\Pi_4 = 140$; the vertical line coincides with the last operating point, beyond which the examination sample begins.

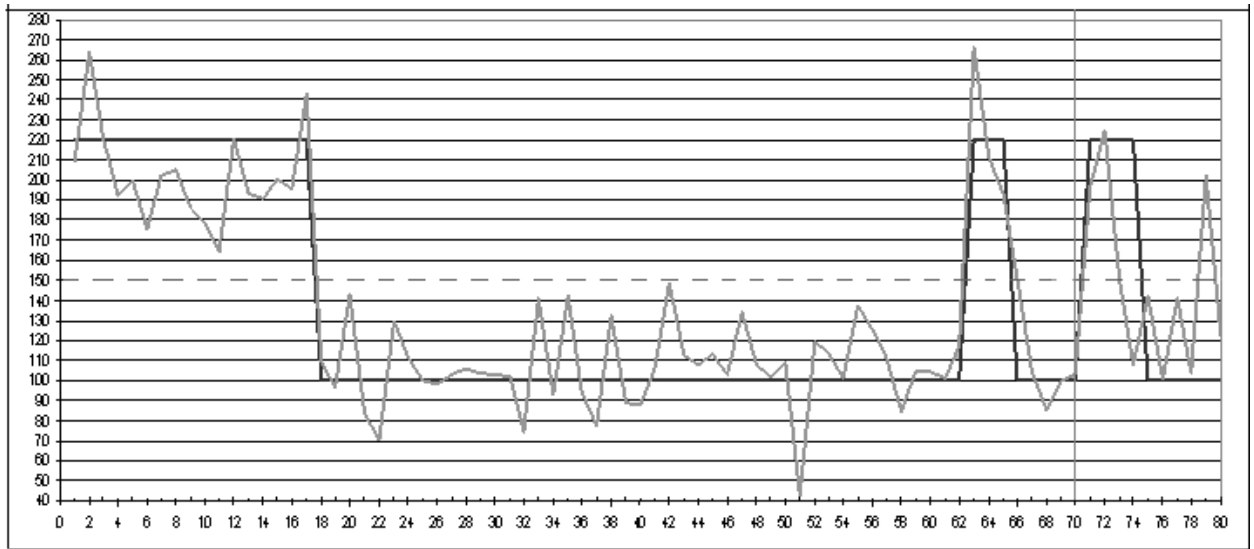


Figure 5.16 – Approximating function of the BW classifier on $W1$

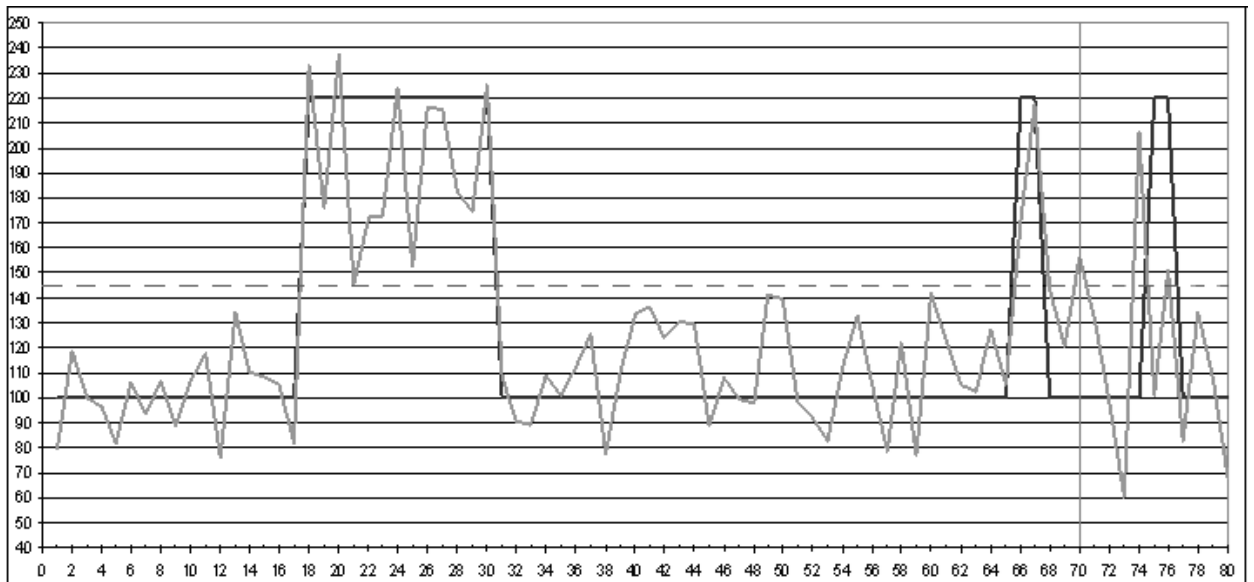


Figure 5.17 – Approximating function of the CP classifier on $W2$

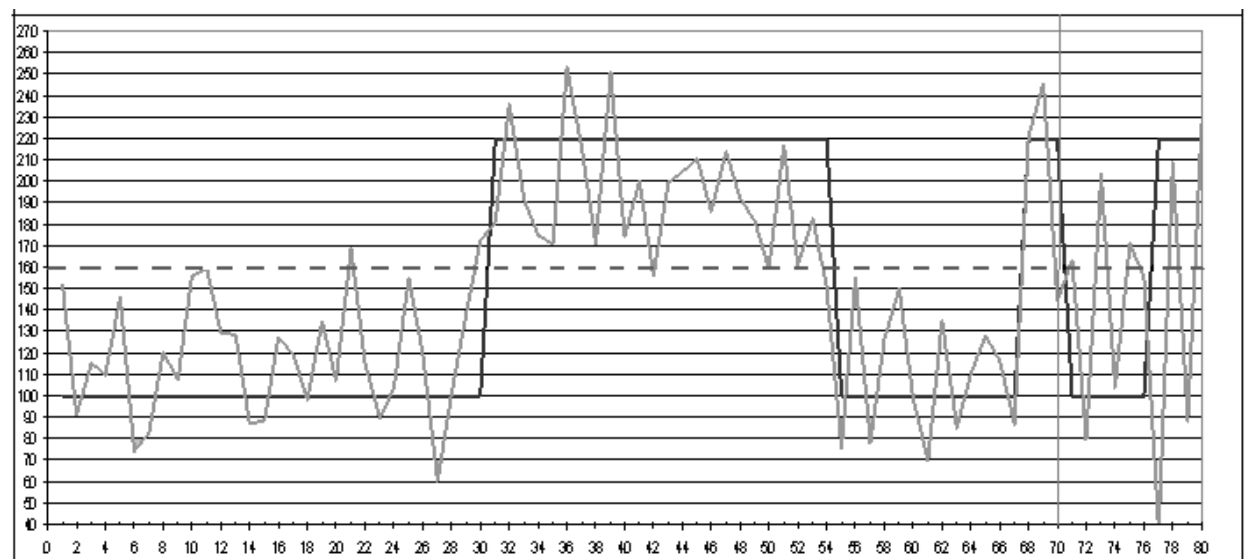


Figure 5.18 – Approximating function of the DT classifier on $W3$

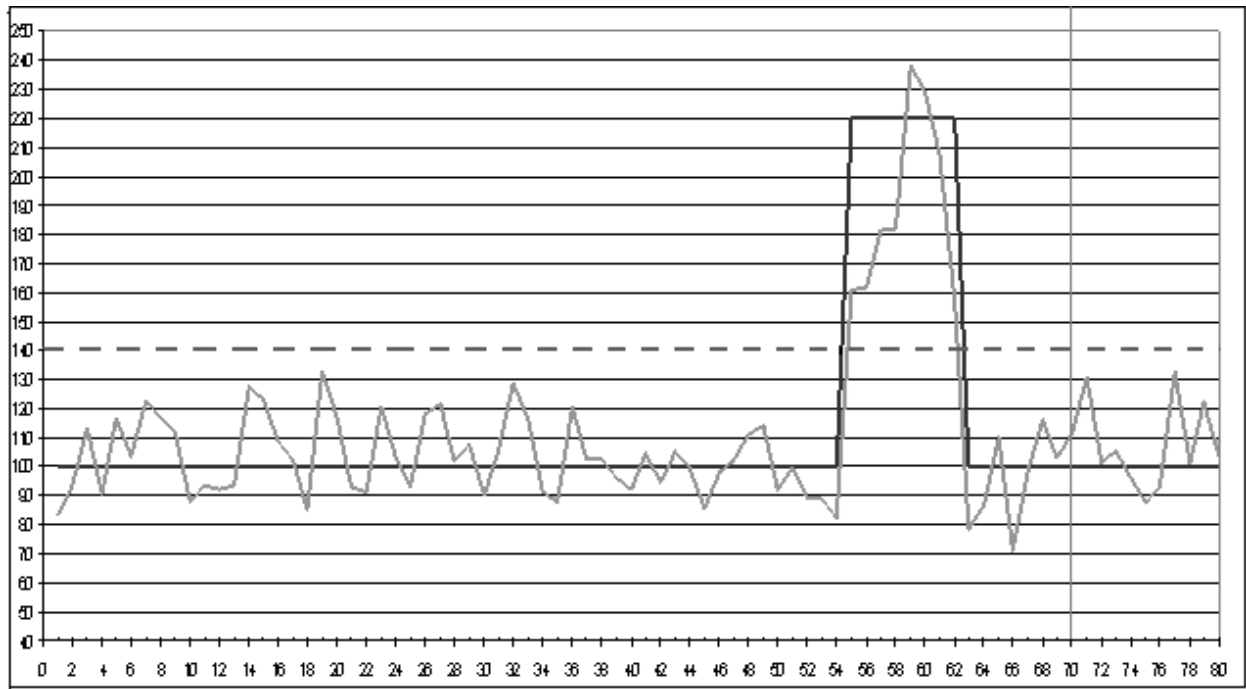


Figure 5.19 – Approximating function of the CPHS classifier on W4

5.3.5 Modeling analysis

The following recognition accuracy indicators were calculated for the classifiers: sensitivity S_E and specificity S_P :

$$S_E = \frac{TP}{TP+FN}, \quad S_P = \frac{TN}{TN+FP},$$

where TP is the number of correct answers "+" by the classifier among all patients with this diagnosis; TN is the number of correct "-" answers among all patients with a different diagnosis; FN is the number of type I made by the classifier; FP is the number of errors of type II errors made by the classifier.

Sensitivity $S_E \in [0; 1]$ and specificity of $S_P \in [0; 1]$ are related to type I and type II errors and provide an indication of the risk of misdiagnosis by the physician.

The values of these characteristics and the percentage of accurate recognition for the classifiers are presented in Table 5.13. Recognition accuracy was calculated as

$$T_G = \left(1 - \frac{FN_G + FP_G}{n_G}\right) \cdot 100\%,$$

where the index $G \in \{W, U, A, B, C\}$.

Table 5.13 – Characteristics of classifiers

Sampling	Accuracy Recognition, %				Sensitivity				Specificity			
	WD	CP	DT	CPHS	WD	CP	DT	CPHS	WD	CP	DT	CPHS
W	97.5	96.2	90	100	0.95	0.94	0.87	1	0.95	0.96	0.92	1
U	100	98.5	95.7	100	1	1	0.93	1	1	0.82	0.98	1
A	100	100	96.7	100	1	1	0.96	1	1	1	0.97	1
B	100	87.5	90	100	1	1	0.67	1	1	0.83	1	1
C	80	80	50	–	0.75	0.5	0.5	–	0.83	0.87	0.5	–

5.3.6 Conclusions on the task of differential diagnosis

The high performance of the CPSG classifier allows us to recommend it for differential diagnosis, however, it will require modification taking into account new data due to the extremely small number of patients (8 people).

When diagnosing patients, the recognizing system made 4 failures of classification and 5 recognition errors. In general, for the entire sample, at least 89% were correctly classified (71 patients out of a total of 80 patients).

5.4 Discussion

The obtained accuracy indicators, especially in the first two tasks, are caused by the complexity of the tasks or the non-triviality of the input data: they are irregular, and no patterns can be seen in them.

In the task of forecasting magnetic storms, the data is non-stationary, acyclic with no trend, the process of storm occurrence is complex, almost random. Experts believe that the only indicator of the beginning of a storm is a small jump in the *Dst* index before its sharp fall. And even then - not "by eye", but using modeling methods.

If the data in a drug evaluation problem contains some trend, then the residuals are chaotic. Obtaining good modeling results is rare here, as, for example, predicting exchange rates or the cost of oil in financial and commodity markets is rare. Therefore, it was decided to apply a sample partitioning algorithm, also taking into account its small volume, and it was possible to apply a quasi-optimal partition, the computational complexity of which is exponential. That is, to improve the modeling results, two combinatorial problems were solved: the first was the selection of an optimal set of points for the training and testing sequences; the second is the selection of the optimal set of variables and their multiplicative transformations, i.e. the model was built on the samples obtained as a result of solving the first problem. It should be noted that the histograms of the error distribution of the model obtained in this problem are based on a very small number of observations, insufficient to speak of any statistically significant conclusions.

The problem of differential diagnosis by formulation is a classical problem of pattern recognition and, in principle, can be solved by any classification method. In the book we showed how it can be solved using the group method of data handling.

Conclusion

The purpose of this book was to introduce a wide range of readers to the tasks of heuristic inductive computer modeling, methods of inductive modeling based on empirical measurement data, the method of group accounting of arguments, its basic principles and approaches to solving problems; highlight its most difficult tasks: structural-parametric identification, the related problem of sample partitioning and process forecasting; analyse classical algorithms and describe the most modern of them, and also note the importance of their properties such as: noise-immunity, identification and convergence; show practical examples of the use of such algorithms to solve applied problems of inductive modeling.

The book discusses theoretical issues of the conditions for obtaining an optimal partition and the coordination of sample partition criteria with the criterion of external addition when choosing models using GMDH. From a theoretical point of view, it is also important that the use of criteria for quasi-optimal sample partitioning to evaluate the transformation of images is one of the simple ways to determine the presence of their perspective-affine transformations. Attention is paid to ways to improve the performance of the algorithm for searching the result of a quasi-optimal partition; a comparison of quasi-optimal with traditional partitioning methods was carried out; the properties of the sample splitting criteria were analysed; the influence of the method of partitioning the sample on the accuracy of identification and extrapolation was investigated, which made it possible to find ways to improve the results by visualizing and weighing the decisions obtained as a result of various partitions. In the problem of extrapolating the trend in changes in budget revenues of Ukraine, model outputs obtained using quasi-optimal and partitioning “by variance”, which are respectively, the upper and lower estimates of the real process, reflect the trend in changes in budget revenues and are used to increase the accuracy of extrapolation.

New and increasingly complex modeling problems arise every day. Thanks to the latest developments in the field of creating iterative algorithms of GMDH - a generalized relaxation iterative algorithm, it is possible to solve problems in which the empirical data table contains hundreds of thousands of observations and thousands of variables. However, algorithms capable of handling even larger data volumes are needed for problems involving millions and hundreds of millions of variables, such as in the study of signals generated by the human brain.

The range of the number of observations can be expanded by a proportional increase in the amount of RAM, and the interval in the number of variables can be expanded from tens to hundreds of times and even thousands of times due to parallelization of GRIA operations on the central and/or graphics processors.

Problems of overcoming GRIA's shortcomings also remain unresolved: model degradation and the inability to converge to an exact solution in the presence of highly correlated column vectors of the original matrix.

It is necessary to note a significant positive advance in the field of GMDH in the task of studying the properties of GMDH algorithms, in particular, iterative algorithms. The developed methodology for conducting computer experiments guarantees the statistical reliability of the results obtained and makes it possible to evaluate the results depending on the characteristics of the initial data. It should be noted that it can be applied to the study of any iterative algorithm, not just the GMDH algorithm. We hope that the experiments conducted to study the properties of the GRIA helped readers gain a deeper understanding of the theory and fundamental principles of GMDH and see how they work in practice.

Although the examples of solving applied modeling problems presented in the book cover only three application areas: economics, astrophysics, and medicine, the group method of data handling is also used in other areas: engineering, biology, and energy, etc. In general, wherever there is a sample of empirical data, it is necessary to construct a mathematical relationship between input and output.

Appendix A

Additional computer experiments

Justification of the dependence of convergence rate r^* on the number of observations n_A , linear model.

We examine how the histogram of distribution r^* and the corresponding value of average determinant d_{aver} change with changes in the parameter n . To calculate d_{aver} , the values of the determinants d of all matrices for which the algorithm converged to a solution after r^* iterations, falling within the corresponding interval for r^* are calculated. The d_{aver} value is the average of the calculated values of d .

The following model was constructed in the experiment:

$$y = 6.29447 - 7.29046 \cdot x_1 + 8.11584 \cdot x_2 + 6.70017 \cdot x_3 - 7.46026 \cdot x_4 + 9.37736 \cdot x_5 \quad (A.1)$$

As an example, the following parameters of the algorithm were set: freedom of choice $F = 5$, modeling accuracy was chosen as $\varepsilon = 10^{-12}$ based on the consideration of achieving the maximum accuracy that can be obtained by studying recurrent algorithms.

Figure A.1 shows a histogram of the distribution r^* with varying parameter n . The values of average determinant for the main peaks of histograms in figure A.1 are shown in figure A.2. Above the bars of the histograms in figure A.2 are indicated a thousandths of the values of average determinant d_{aver} .

From figures A.1 and A.2 it can be concluded that: the parameter n has a direct proportional effect on:

- the value of the determinant d and consequently, the measure of orthogonality of the column vectors of matrix $\mathbf{X}$. Therefore, the number of observations can be used to obtain a correlation matrix with a given value of the determinant;
- the convergence rate of the algorithm.

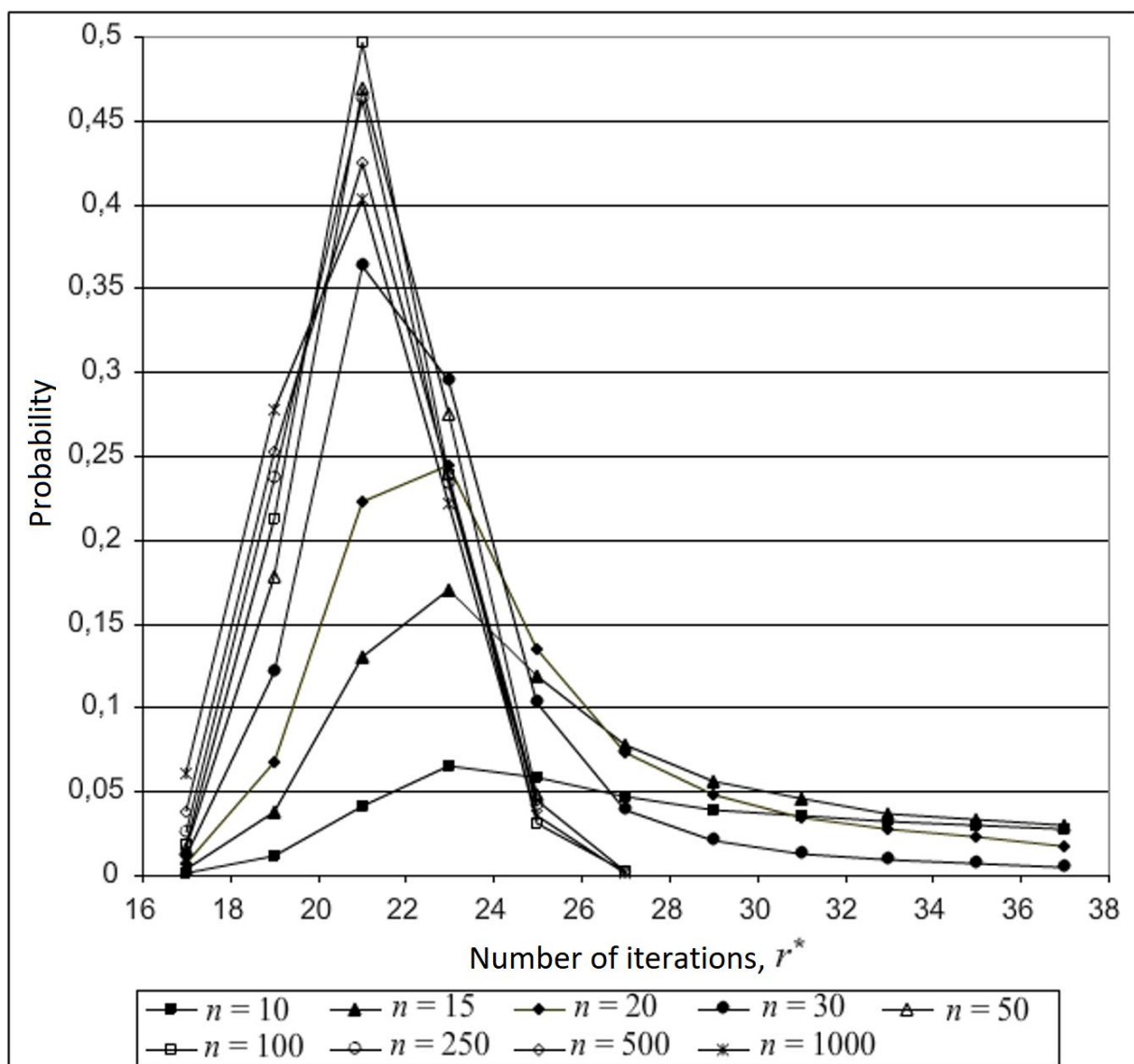


Figure A.1 – Histograms of the r^* distribution when the number of observations n changes, linear model

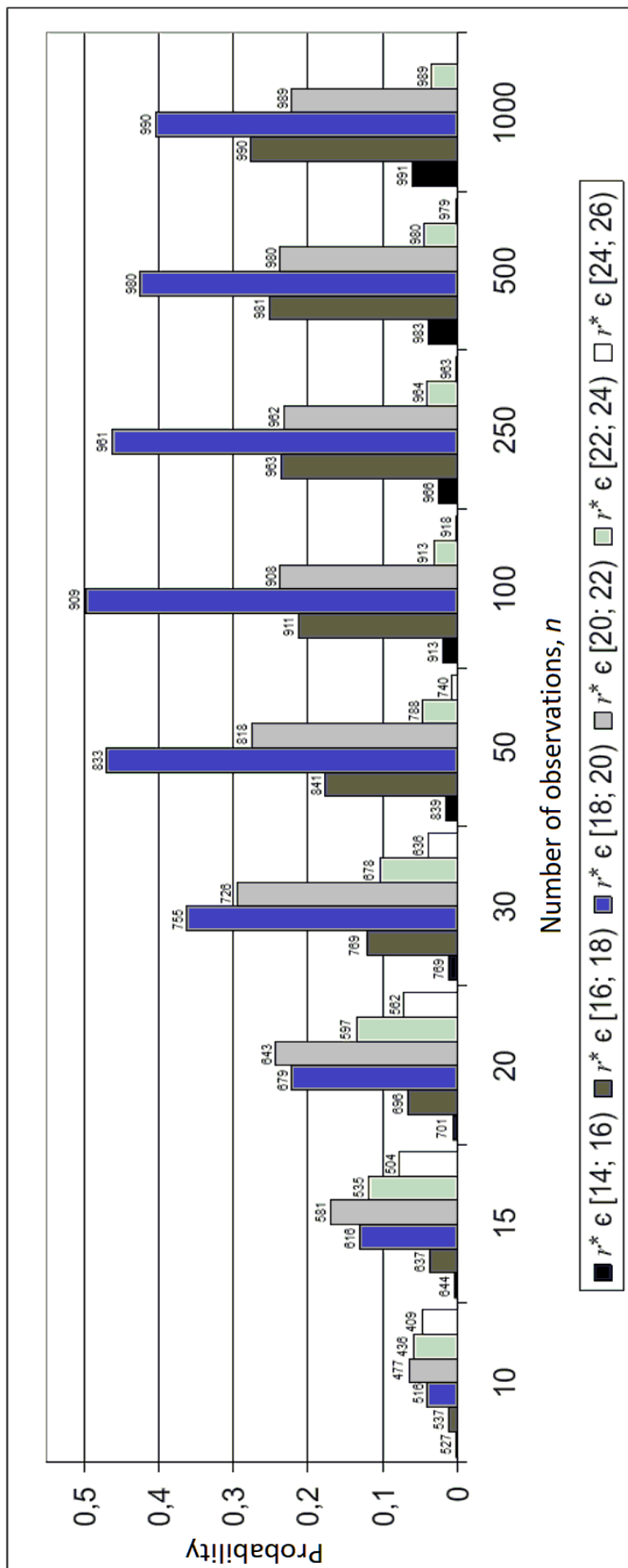


Figure A.2 – Histograms and corresponding d_{aver} values for each n when constructing a linear model

Numerical confirmation of the dependence of convergence rate r^* on the number of observations n , nonlinear model.

We investigate the convergence rate when constructing a nonlinear model based on the initial variables

$$y = 6.29447 - 7.29046 x_1 + 8.11584 x_2 + 6.70017 x_3 - 7.46026 x_4 + 9.37736 x_5 + 8.26752 x_1^2 - 5.57932 x_1^3 + 2.64719 x_1^2 x_2. \quad (\text{A.2})$$

Algorithm parameters: freedom of choice $F = 5$, modeling accuracy $\varepsilon = 10^{-6}$. The results are presented in Figs. A.3 and A.4.

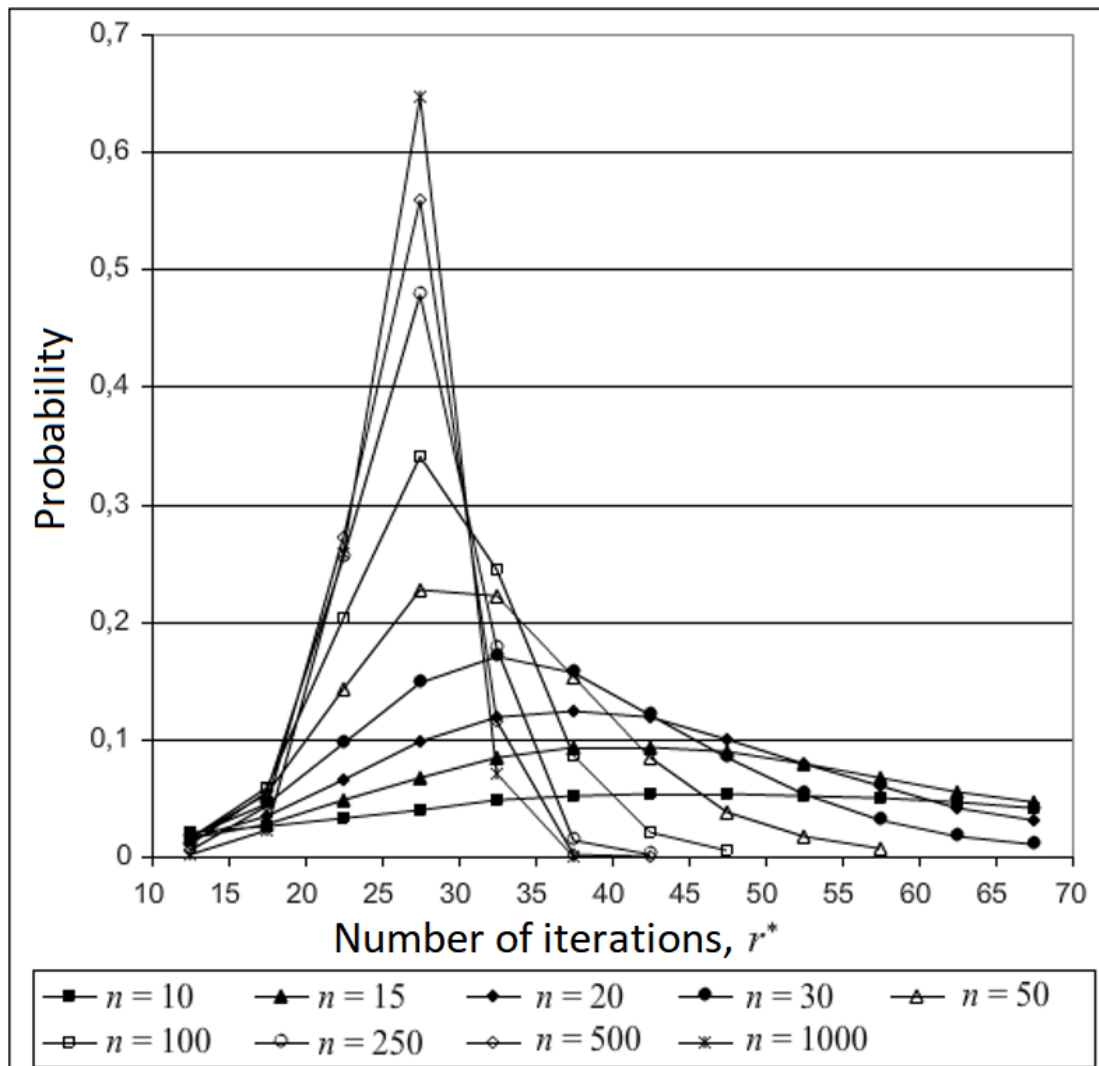


Figure A.3 – Histograms of the r^* distribution when the number of observations n changes, nonlinear model

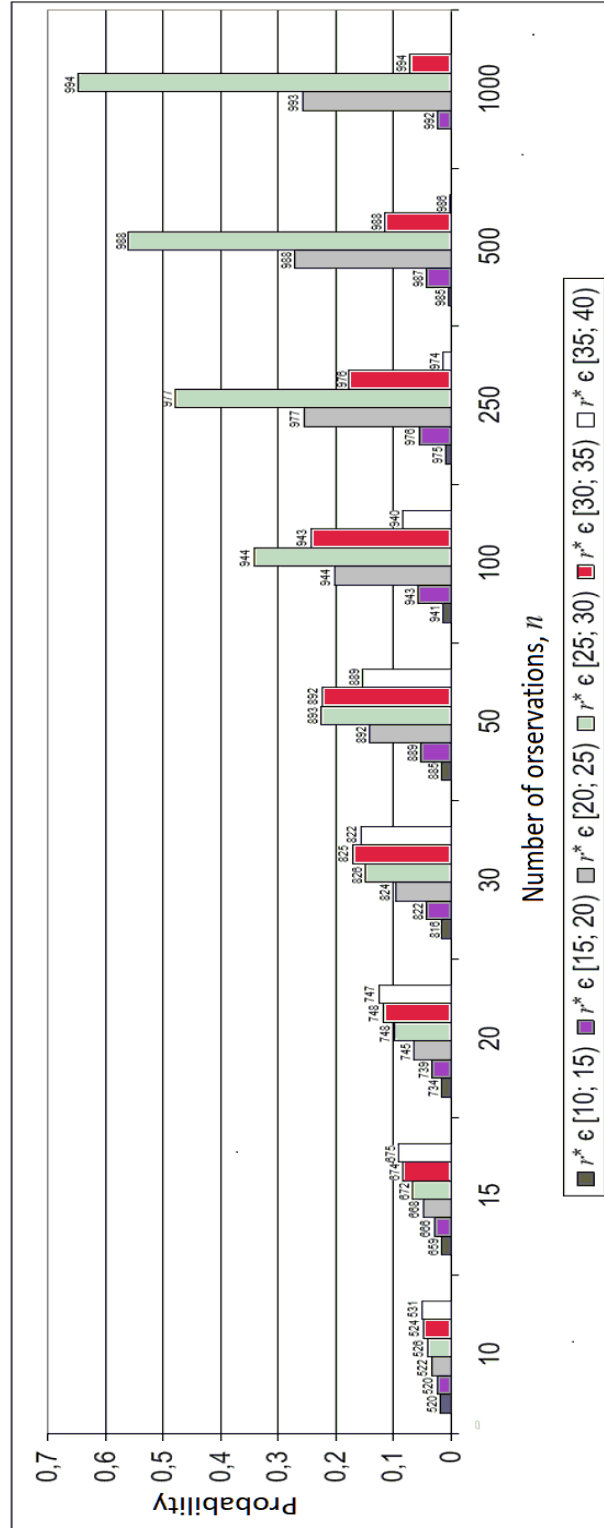


Figure A.4 – Histograms and corresponding d_{aver} values for a series of n values, nonlinear model

Analysis of figures A.3 and A.4 allow us to draw conclusions similar to the case of a linear model, i.e. the parameter n directly proportionally affects:

- the value of the determinant d and consequently, on the measure of orthogonality of the vector-columns of the matrix $\mathbf{X}$;
- the speed of convergence of the algorithm.

A distinctive feature of the nonlinear model is the shift in the histogram peaks toward increasing the number of observations and the number of iterations to achieve the same model accuracy.

To study the algorithm's convergence rate depending on the initial data, it is necessary to search for the values of MGP parameters at which the mathematical expectation of parameter d of the matrix Σ_X is close to the specified value.

Justification of the dependence of the convergence rate r^* on the value of the determinant d of the correlation matrix.

Let us compare the distribution histograms of the following quantities for a series of values of $n \in \{20, 30, 50, 100\}$:

- 1) convergence rate r^* of the algorithm (Fig. A.5);
- 2) values of the matrices determinant d generated by the MGP (Fig. A.6).

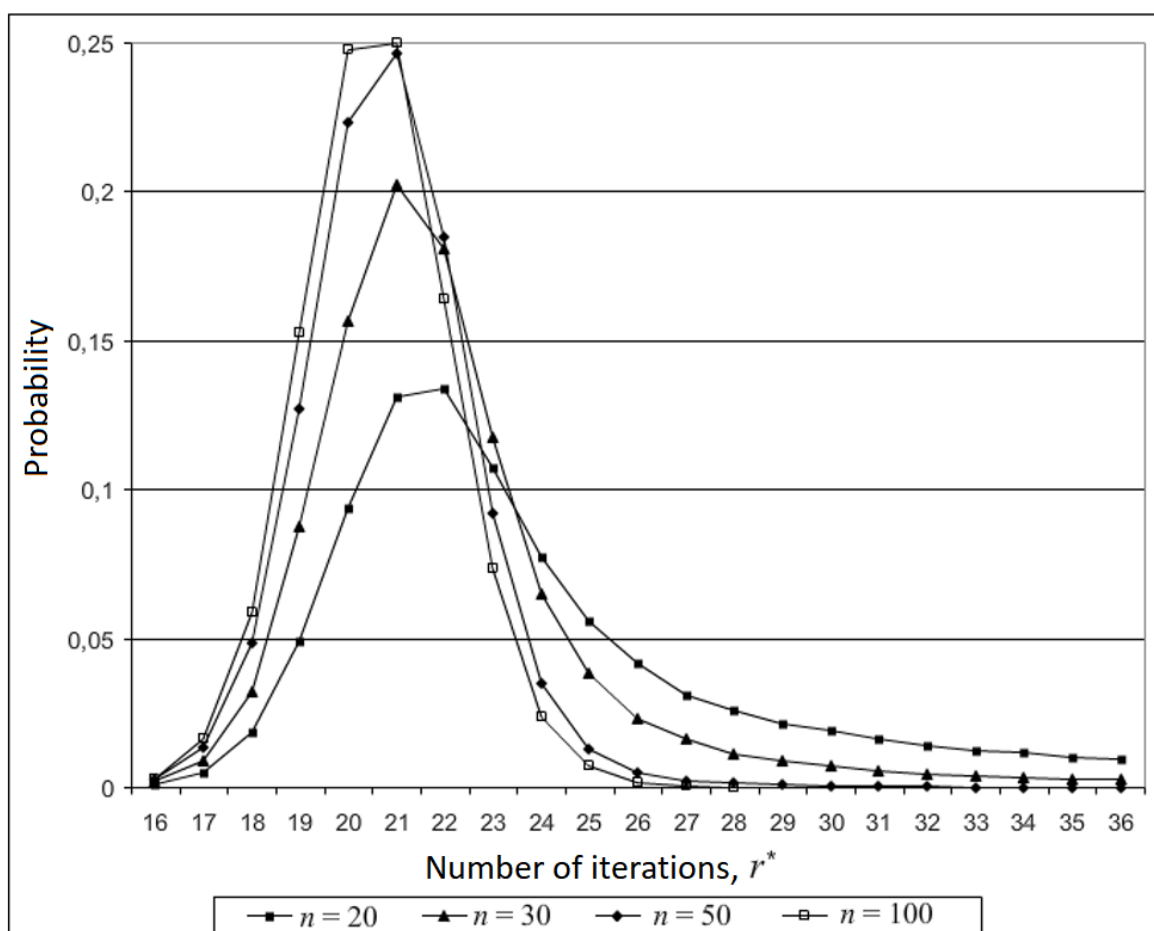


Figure A.5 – Histograms of distribution of the parameter r^* for the selected values of n

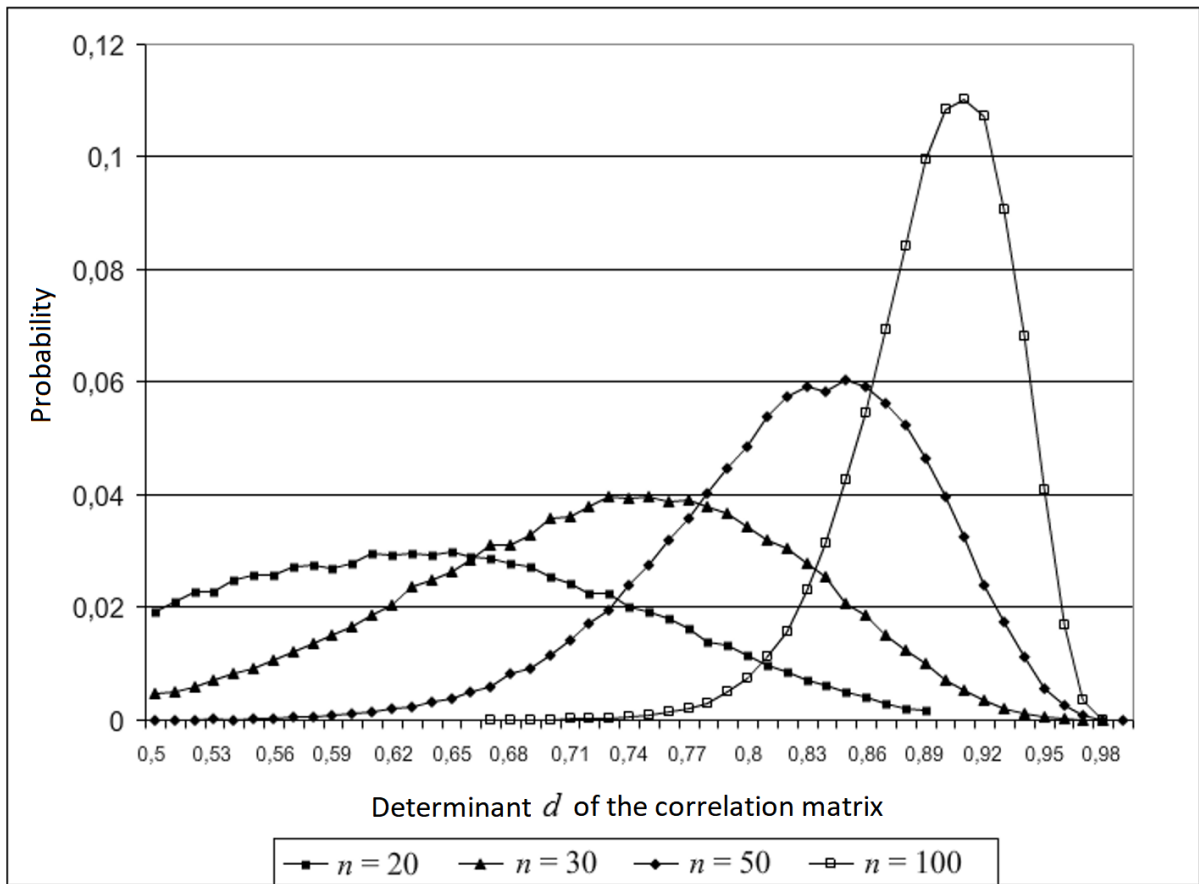


Figure A.6 – Histograms of distribution of the determinant of matrices d generated by the MGP for the selected values of n

The following conclusions can be drawn from these figures:

1. Each value of n corresponds to its own value of the matrices determinant d , corresponding to the probability peak.
2. With the larger value of n , the smaller variance of the parameter d of generated matrices, and the histogram shape is not symmetrical.
3. The histograms in figures are mirrored and the compression of distribution histogram of parameter r^* to the left is caused by compression of the histograms of parameter d to the right.
4. Because the values of the parameter d for the peaks of histograms are close to each other, the d parameter can be used to determine the convergence rate of the algorithm.

References

1. Ивахненко А.Г. Системы эвристической самоорганизации в технической кибернетике. – Киев: «Техніка», 1971. 372 с.
2. Ивахненко А.Г. Метод группового учета аргументов – конкурент метода стохастической аппроксимации // Автоматика. – 1968.– №3.– С. 58–72.
3. Ивахненко А.Г. Индуктивный метод самоорганизации моделей сложных систем. – Киев: Наукова думка. 1982. - <https://www.gmdh.net/articles/theory/bookInductModel.pdf>
4. Кондаков Н.И. Логический словарь-справочник. М. Наука. 1976.
5. Бертран Рассел. Искусство мыслить, М., «Дом интеллектуальной книги», 1999.
6. Винер Н. Я – математик. М.: Наука , 1964.
7. Финн В.К. Индуктивные методы Д.С. Милля в системах искусственного интеллекта. Часть 1, II.// Журнал «Искусственный интеллект и принятие решений» 2010, №3, с.3-21; №4 с.14-40
8. Советов Б.Я. Моделирование систем. Учебник для ВУЗов / Советов Б.Я., Яковлев С.А. – М.: Высшая школа, 1985. – 320 с.
9. Ивахненко А. Г. Помехоустойчивость моделирования / Ивахненко А. Г., Степашко В. С. – К.: Наукова думка, 1985. – 214 с. - <https://www.gmdh.net/articles/theory/bookNoiseIm.pdf>
10. Ивахненко А.Г. Долгосрочное прогнозирование и управление сложными системами / Ивахненко А.Г. – К.: Техніка, 1975. – 312 с. - <https://www.gmdh.net/articles/theory/ForecastBook.pdf>
11. Madala H.R. Inductive Learning Algorithms for Complex Systems Modeling / Madala H.R., Ivakhnenko A.G. – Boca Raton, Florida: CRC Press Inc., 1994. – 368 p. - <https://www.gmdh.net/articles/theory/GMDHbook.pdf>
12. Бабенко К. И., Васильев М. М. О доказательных вычислениях в задаче об устойчивости течения Пуазейля // ДАН СССР. – 1983. – Т. 273. – № 6. – С. 1289–1294
13. Степашко В.С. Автоматизована структурна ідентифікація прогнозуючих моделей складних об'єктів [Текст] : автореф. дис. ... доктора техн. наук : 05.13.01 / Степашко Володимир Семенович; Ін-т кібернетики НАН України. – К., 1994. – 46 с.
14. Ивахненко А.Г. Моделирование сложных систем по экспериментальным данным / Ивахненко А.Г., Юрачковский Ю.П. – М.: Радио и связь, 1987.– 120 с. - <https://www.gmdh.net/articles/theory/Complex Systems Modelling by Experimental Data.djvu>
15. Степашко В. С. Метод критических дисперсий как аналитический аппарат теории индуктивного моделирования / Степашко В. С. // Проблемы управления и информатики. – № 2. – 2008. – С. 8-26.
16. Sarychev A. P. Modeling in the Class of Systems of Regression Equations by the Group Method of Data Handling // Journal of Automation and Information Sciences. V. 45, 2013 Issue 3. p. 4-22
17. Єфіменко С.М. Інструментальні засоби для дослідження та застосування методів моделювання за статистичними даними : Дис. ... канд. техн. наук : 05.13.06 „Інформаційні технології” / С. М. Єфименко – К., 2009, – 145 с.
18. Юрачковский Ю.П., Грошков А.Н. Оптимальное разбиение исходных данных на обучающую и проверочную последовательности на основе анализа функции распределения критерия // Автоматика. - 1980.- N 2.- С. 5-9
19. Степашко В. С. Потенциальная помехоустойчивость моделирования по комбинаторному алгоритму МГУА без использования информации о помехах / В. С. Степашко // Автоматика. – 1983.– №3.– С. 18–27.
20. Канторович Л. В. Математические методы организации и планирования производства . Л.: Изд-во Ленинградского университета. 1939. 64 с.
21. Канторович Л. В. Экономический расчет наилучшего использования ресурсов. М.: Изд-во АН СССР.1960. 347 с.
22. Дрейпер Н. Прикладной регрессионный анализ : в 2-х кн. / Дрейпер Н., Смит Г. – М.: Статистика. Кн. 2. – 1973. – 392 с.
23. Себер Дж. Линейный регрессионный анализ / Себер Дж. – М.: Мир, 1980.– 456 с.

24. Акаике Х. Развитие статистических методов / Акаике Х. // Современные методы идентификации систем. – М.: Мир, 1983. – С.148-176.
25. Вапник В.Н. Восстановление зависимостей по эмпирическим данным / Вапник В.Н. – М.: Наука, 1979. – 447с.
26. Mallows C.L. Some comments on Cp / Mallows C.L. // Technometrics. – 1973. – v.15. – P.661-667.
27. Kohavi, Ron. A study of cross-validation and bootstrap for accuracy estimation and model selection // Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence. – 1995. – 2 (12). – P. 1137–1143.
28. Mosteller F. A k-sample slippage test for an extreme population / Mosteller F. // Annals of Mathematical Statistics. – 1948. – Vol. 19 (1). – p. 58–65.
29. Tukey J.W. Bias and confidence in not-quite large samples (abstract) / Tukey, J.W. // The Annals of Mathematical Statistics. – 1958. – Vol. 29, 2. – P. 614.
30. Тьюки Дж. Анализ результатов наблюдений: Разведочный анализ / Тьюки Дж. – М.: Мир, 1981. – 693с.
31. Efron B. Bootstrap methods: Another look at the jackknife / Efron B. // The Annals of Statistics. – 1979. – 7. – P. 1-26.
32. Breiman L. Technical Report No. 421 / Breiman L. – 1994.
33. Ивахненко А.Г. Самообучающиеся системы распознавания и автоматического управления. – Киев: Техника. 1969.
34. Ивахненко А.Г. Непрерывность и дискретность: Переборные методы моделирования и кластеризации. – Киев: «Наук. думка», 1990. 223с.
35. Ивахненко О.Г., Зайченко Ю.П. Машины починають мислити. – Київ: Тов. "Знання" УРСР, 1967.
36. Ивахненко О.Г., Лапа В.Г. Передбачення випадкових процесів. – Київ: «Наук. думка», 1969.
37. Перцептрон – система распознавания образов / Под ред. А.Г. Ивахненко. – Киев: «Наук. думка», 1975.
38. Ивахненко А.Г., Зайченко Ю.П., Димитров В.Д. Принятие решений на основе самоорганизации. – М.: «Сов. радио», 1976.
39. Ивахненко А.Г., Пека Ю.П., Востров Н.П. Комбинированный метод моделирования водных и нефтяных полей. – Киев: «Наук. думка», 1984.
40. Self-organizing methods in modeling: GMDH type algorithms / Ed. S.J.Farlow. – New York, Basel: Marcel Decker Inc., 1984. – 350 p.
41. Бир С. Кибернетика и управление производством / Бир С. [пер. с англ. В. Я. Алтаева]: – М.: Наука, 1963. – 276с.
42. Колмогоров А. Н. О представлении непрерывных функций нескольких переменных в виде суперпозиции непрерывных функций одного переменного и сложения / А. Н. Колмогоров // Докл. АН СССР. – 1957. – Т. 114, – № 5. – С. 953-956.
43. Арнольд В. И. О представлении функций нескольких переменных в виде суперпозиции функций меньшего числа переменных / В. И. Арнольд // Мат. просвещение. – 1958. – Вып. 3. – С. 41-61.
44. Bary N.K. Mémoire sur la representation finie des fonctions continues// Math. Ann. 103. 1930. p.145-248. p. 598-653.
45. Нагель Е., Ньюмен Д. Теорема Геделя [пер. с англ.]: – М.: Знание, 1970 – 62 с.
46. Gabor D. Cybernetics and the future of industrial civilization / Gabor D. – J.Cybern. – 1971. – No2. – P. 1-4.
47. Trie [Web resource] – Access : <https://en.wikipedia.org/wiki/Trie>
48. Сб. Читающие автоматы. Под. ред. Ковалевского В. А. – К.: Наукова думка. 1964.
49. Павлов О.А. Критерій ранжування для порогового самовідбору змінних в алгоритмах МГУА // Автоматика. - 1969.- N 4.- С. 89-91.
50. Степашко В.С. Структурная идентификация прогнозирующих моделей в условиях планируемого эксперимента // Автоматика. - 1992. - N 1. - С.26-35.

51. Self-organizing methods in modeling: GMDH type algorithms / Ed. S.J.Farlow. – New York, Basel: Marcel Decker Inc., 1984. – 350 p.
52. Ивахненко Н.О. Комбинаторні алгоритми МГВА, в яких екзаміну на регулярність піддіються як „симетричні”, так і „несиметричні” поліноми / Ивахненко Н.О., Кваско М.З. // Автоматика. – 1972.– №5.– С. 42–48.
53. Степашко В.С. Комбинаторный алгоритм МГУА с оптимальной схемой перебора моделей/ Степашко В.С. // Автоматика. – 1981. – №3.– С. 31–36.
54. Степашко В.С. Потенциальная помехоустойчивость моделирования по комбинаторному алгоритму МГУА без использования информации о помехах / В.С. Степашко // Автоматика. – 1983.– №3.– С. 18–27.
55. Фаддеев Д. К. Вычислительные методы линейной алгебры / Д. К. Фаддеев, В. Н. Фаддеева. – 2-е изд. – М. : «Наука», 1963. – 656 с.
56. Шелудько О.И. Самоорганизация математических моделей при решении некоторых задач надежности и контроля: Дис. ... канд. техн. наук : 05.13.01 / Шелудько Олег Иванович – К., 1975. – 166 с.
57. Ивахненко Н. А., Марчев А. А. Самоорганизация математической модели для перспективного планирования строительно-монтажных работ. // Автоматика. – 1978, № 3, с. 12-18.
58. Юрачковский Ю. П. Восстановление полиномиальных зависимостей на основе самоорганизации / Юрачковский Ю. П. // Автоматика. – 1981. – № 4. – С. 15-20.
59. Ivakhnenko A. G. Self-organization of Neural Networks with Active Neurons / Ivakhnenko A. G., Ivakhnenko G. A., Muller J.-A. // Pattern Recognition and Image Analysis. – v.4. – No. 2. – 1994. –P. 185-196. - https://www.gmdh.net/articles/algor/ann_GMDH.pdf
60. Kordik P. Fully automated knowledge extraction using group of adaptive model evolution : Ph.D. thesis. : Electrical Engineering and Information Technology / Kordik P. – Prague, 2006. – 150 p.
61. Hybrid Particle Swarm Optimization and Group Method of Data Handling for Inductive Modeling / [Godfrey Onwubolu, Anurag Sharma, Ashwin Dayal Deepak Bhartu et al.] // Proceedings of 2nd International Conference on Inductive Modelling (ICIM'2008). – 2008.– К. : IRTC. – P. 95-103.
62. Булгакова О.С. Узагальнений ітераційний алгоритм індуктивного моделювання з застосуванням мережевих технологій. // Дис. на здоб. вчен. степ. канд. техн. наук. Киев 2011. С. 171.
63. Белозерский Е. А. Об одном подходе к построению алгоритмов МГУА с линейными частными описаниями / Белозерский Е. А., Ивахненко Н. А., Юрачковский Ю. П. // Автоматика. – 1981. – №5. – С. 3-7.
64. Buryan P. Enhanced MIA-GMDH Algorithm / Buryan P. // Proceedings of International Workshop of Inductive Modelling (IWIM'2007). – Prague : CTU in Prague, 2007. – P. 144-155.
65. Kordik P. The modified GMDH method applied to model complex systems / Kordik P., Snorek M., Genyk-Berezovskyj M. // Праці Міжнар. конф. з індуктивного моделювання в 4-х томах (МКІМ-2002). – Т.3.– Львів .: Державний НДІ інф. інфраструктури, 2002. – С. 150-155.
66. Павлов О. В. Алгоритми самоорганізації в задачах підвищення інформативності геометричних моделей процесів, заданих точковим каркасом : дис. ... канд. техн. наук [Текст] : дис... канд. техн. наук: 05.01.01 / Павлов Олександр Володимирович ; Національний технічний ун-т України "Київський політехнічний ін-т". - К., 2006. - 197 с.
67. Светальский Б. К., Ковальчук П. И. Многорядный алгоритм МГУА с селекцией первичных аргументов / Светальский Б. К., Ковальчук П. И. // Автоматика. – 1979. – № 4. – С. 31-35.
68. Marce Jirina. Genetic Selection and Cloning in GMDH MIA Method // Marce Jirina, Marce Jirina. // Proceedings of International Workshop of Inductive Modelling (IWIM'2007). – Prague : CTU in Prague, 2007. – P. 165-171.
69. Юрачковский Ю. П. Сходимость многорядных алгоритмов МГУА / Ю. П. Юрачковский // Автоматика. – 1981. – № 3. – С. 36-43.
70. Зайченко, Юрий Петрович Нечеткие модели и методы в интеллектуальных системах: учебное пособие для вузов / Юрий Петрович Зайченко : Под общ. ред. М. З. Згуровский . – Київ: Слово, 2008 . – 342 с.

71. Дивак М. П. Особливості побудови інтервальної системи алгебричних рівнянь та методу її розв'язку в задачах ідентифікації лінійного інтервального різницевого оператора [Web resource] / М. П. Дивак, Т. М. Дивак // Індуктивне моделювання складних систем. – 2009. – Вип. 1. – С. 35-43.
72. Зайченко Ю. П., Засць І.О. Синтез та адаптація нечітких моделей прогнозування на основі методу самоорганізації. Наук. вісті НТУУ КПІ, № 2. 2001.
73. Ю. П. Зайченко. Нечіткий Метод Групового урахування аргументів та його застосування для прогнозування Економічне процесів " Наукові дослідження, - Vol. 7, № 1, 2006. С.83-96.
74. Tadashi Kondo, Junji Ueno. Feedback GMDH-Type Neural Network Self-Selecting Optimum Neural Network Architecture and Its Application to 3-Dimensional Medical Image Recognition of the Lungs // Proceedings of International Workshop of Inductive Modelling (IWIM'2007). – Prague : CTU in Prague, 2007. – P. 63-70.
75. Космические супер-бури в истории [Web resource] / G. Marchenko – 2011. – С.1 – Access : – <http://sw.astron.kharkov.ua/sw1700-now.htm>
76. Розенблатт Ф. Принципы нейродинамики: Перцептроны и теория механизмов мозга. – М: Мир, 1965.
77. Розенблатт Ф. Аналитические методы изучения нейронных сетей// Зарубежная радиоэлектроника, 1965, №8.
78. Івахненко О.Г. Алгоритми методу групового врахування аргументів (МГВА) з лінійними операторами / О.Г. Івахненко, Ю.В. Коппа // Автоматика. – 1969.– №4, с.69–78.
79. Ivakhnenko A.G., Savchenko E.A., Ivakhnenko G.A. Discrete Optimization of Square Data Samples as the First Stage of the Optimization of a Discriminant or Predictive Model // Pattern Recognition and Image Analysis, Vol. 14, 2004, pp.489-494.
80. Aksenova T.I., Volcovich V.V., Tetko I.V. Robust Polynomial Neural Network in Quantative-structure Activity Relationship Studies // System analysis modeling simulation.- 2003. – 43, N 10. P. 1331-1341 - <https://www.gmdh.net/articles/algoprnn.pdf>
81. Четверухин Н.Ф. Проективная геометрия. – М: Учпедгиз, 1953. – 350с.
82. Computer-assisted proof [Web resource] – Access: https://en.wikipedia.org/wiki/Computer-assisted_proof
83. Ляхов А.Ф. Прямые методы решения систем линейных алгебраических уравнений / А. Ф. Ляхов, Е. В. Солдатов, Е. В. Чернова – Н.Новгород: ННГУ, 1999.–16 с.
84. Ободников А.А. Способ индивидуального подбора лечебных средств для определенного больного / Ободников А. А. – М.: Описание изобретения к патенту Российской Федерации № 2189587, 20.09.2002, – Бюл. № 26.
85. Соколовский В.В. Тиол-дисульфидное соотношение крови как показатель состояния неспецифической резистентности организма / Соколовский В.В. – СПб: СПб Медицинская Академия последипломного образования, 1996. – 27с.
86. Statistics: methods [Web resource] – Access: <https://en.wikipedia.org/wiki/Statistics#Methods>
87. Ключин Д.А. Непараметричні методи розпізнавання з гарантованим рівнем значущості : дис. доктора фіз.-мат. наук : 01.05.01 / Ключин Дмитрий Анатольевич. – К., 2008. – 278 с.
88. Томілін, В. В. Етіологія, прогнозування, профілактика та лікування геморагічних ускладнень при легких формах коагулопатій і тромбоцитопатій [Текст] : автореф. дис. ... доктора мед. наук : 14.01.31 / Томілін Володимир Володимирович; АМН України, ДУ "Ін-т гематології та трансфузіології". – К., 2011. – 39 с.
89. Баркаган З.С. Исследования системы гемостаза в клинике / З.С. Баркаган. – Барнаул, 1975. – 186 с.
90. Баркаган З.С. Геморрагические заболевания и синдромы - переработанное и дополненное / Баркаган З.С. – 2-е изд. – М: Медицина. – 1988. – 528 с.
91. База данных OMNI2. FTP Directory. [Web resource] – 2012. – Access to base: ftp://nssdcftp.gsfc.nasa.gov/spacecraft_data/omni/
92. Мостелер Ф.,Тьюки Дж. Анализ данных и регрессия: В 2-х вып. Вып. 2. – М: Финансы и статистика, 1982. – 239с.

93. Парновский А.С. Метод регрессионного моделирования и его применение к задаче прогнозирования космической погоды / Парновский А. С. // Проблемы управления и информатики.– 2009. – № 3. – С. 128-135.
94. Бюлетень економічної кон'юнктури України – Київ: НДІ статистики Мінстату України. – 1997. – випуск № 3 – 137с.
95. GMDH Applications: http://www.gmdh.net/GMDH_exa.htm.
96. Самарский А. А. Введение в теорию разностных схем.– М.: Наука, 1971.- 552 с.
97. Boost C++ library [Web resource] – 2011. – Access: <http://www.boost.org/>
98. Mersenne twister [Web resource] – Access: http://en.wikipedia.org/wiki/Mersenne_twister
99. Степашко В. С. Конечная селекционная процедура сокращения полного перебора моделей / Степашко В. С. // Автоматика. – 1983.– №4.– С. 84–88.
- 100.Справочник по типовым программам моделирования / под ред. А.Г. Ивахненко. – Киев: Техника, 1980.-182 с. - <https://www.gmdh.net/articles/theory/RefProgGMDH.pdf>
- 101.Справочник по специальным функциям. – М.:Наука, 1979. – 830с.
- 102.Available information about S. Gabor – Access: <http://histsoc.stanford.edu/pdfmem/SzegoG.pdf>
- 103.McCulloch Warren; W. Pitts. A Logical Calculus of Ideas Immanent in Nervous Activity. Bulletin of Mathematical Biophysics. – 1943, 5 (4): 115–133.
- 104.Юрачковский Ю.П. Метод группового учета аргументов: статистические предпосылки и теоретические аспекты применения. С.105-111. В монографии: Розенберг Г.С., Шитиков В.К., Брусиловский П.М. Экологическое прогнозирование (Функциональные предикторы временных рядов). – Тольятти: Ин-т экологии Волжского бассейна, 1994.–182 с.
- 105.Alexandrov M., Danilova V., Koshulko O., Tejada J. Classifiers for Opinion Mining of Peruvian Facebook (GMDH Shell application) // Proceedings of 4th International Conference on Inductive Modelling (ICIM'2013). – Kyiv : IRTC ITS in Kyiv, 16-20 September, 2013. – P. 241-246.
- 106.Garaev T., Alexandrov M., Koshulko O. Text Classifier as a tool for Short-Term Forecast of Currency Rates (GMDH Shell application) // Proceedings of 4th International Conference on Inductive Modelling (ICIM'2013). – Kyiv : IRTC ITS in Kyiv, 16-20 September, 2013. – P. 261-266.
- 107.Kaurova O., Alexandrov M., Koshulko O. Classifiers of Medical Records Presented in Free Text Form. (GMDH Shell application) // Proceedings of 4th International Conference on Inductive Modelling (ICIM'2013). – Kyiv : IRTC ITS in Kyiv, 16-20 September, 2013. – P. 273-278.
- 108.Martyanov V., Polovinkin A., Tuv E. Feature Learning for Image Classification with Code Book Based on Decision Tree Ensembles // Proceedings of 4th International Conference on Inductive Modelling (ICIM'2013). – Kyiv : IRTC ITS in Kyiv, 16-20 September, 2013. – P. 188-191.
- 109.Высоцкий В.Н., Ивахненко А.Г., Чеберкус В.И. Долгосрочное прогнозирование колебательных процессов при помощи выделения гармонического тренда оптимальной сложности по критерию баланса переменных. – Автоматика,1975, №1, с.23-31.
- 110.Юрачковский Ю.П., Чеберкус В.И. Самоорганизация моделей полигармонических полей. – Автоматика,1981, №1, с.37-45.
- 111.Степашко В. С. Основи теорії МГУА як методу індуктивного моделювання. Праці Міжнародної конференції з індуктивного моделювання МКІМ-2002. Львів 20-25 травня. С. 16-30.
- 112.Степашко В. С. Комбинаторный алгоритм МГУА для структурной идентификации объектов управления с применением различных критериев селекции / В. С. Степашко // Автоматика. – 1978.– 18 с. – (Республиканский фонд алгоритмов и программ: №4951)
- 113.Кошулько А.А. Технология распараллеливания комбинаторного алгоритма МГУА : Дис. ... канд. техн. наук : 05.13.06 „Інформаційні технології” – К., 2008. 20 с.
- 114.Yefimenko S. Comparative Effectiveness of Parallel and Recurrent Calculations in Combinatorial Algorithms of Inductive Modeling // Proceedings of 4th International Conference on Inductive Modelling (ICIM'2013). – Kyiv : IRTC ITS in Kyiv, 16-20 September, 2013. – P. 231-234.
- 115.GMDH [Web resource] – Access: <https://www.gmdh.net>