Method of paralleling grid nodes for multidimensional regions when solving parametric identification tasks

O G Kantor¹, S I Spivak², G N Yusupova² and E S Podvalny³

¹ Department of Corporate Finance and Accounting Technologies, Ufa State Petroleum Technological University, 1 Kosmonavtov Street, Ufa 450062, Russia
² Department of Mathematical Modeling, Bashkir State University, 32 Zaki Validi Street, Ufa 450076, Russia
³ Department of Mathematics and Information Technology in Management, Russian Presidential Academy of National Economy and Public Administration, 84 Vernadsky Prospekt, Moscow 119571, Russia

E-mail: o._kantor@mail.ru

Abstract. One of the approaches to solving parametric identification problems is based on numerical methods. In fact, their use is reduced to the process of iterating over the elements of multidimensional areas of the desired parameters. Increasing the density of the introduced grid increases the amount of calculations and time spent on the process of their implementation. In this regard, it is desirable to improve the efficiency of the computer systems used. An important direction in the context of the above is the development of procedures for parallelization of programs. The article presents an algorithm for searching the nodal points of multidimensional areas based on the principle of data decomposition. In regard to solving the problem of parametric identification of the diffuse Bass model of the spread of innovation, its practical implementation has been carried out.

1. Introduction

The problem of the parametric identification of functional dependence

\[ y = f(\bar{a}, \bar{x}). \] (1)

where \( y \in Y \subset \mathbb{R}^1 \), \( \bar{x} \in X \subset \mathbb{R}^n \) consists of determining of a vector of numerical parameters \( \bar{a} \), which provides the best correspondence of experimental data \( \{\bar{x}_t, y_t\}, \ t = 1, m \), and the values calculated from the model (1) of the function \( \hat{y}_t = f(\bar{a}, \bar{x}_t), \ t = 1, m \).

The use of statistical methods is traditional for solving parametric identification problems. In particular, the least squares method (LSM) is widely used in practice. Its application is based on the fact that the experimental data should consist of independent equally distributed random variables, and their quantity should be sufficient for the obtained results to be considered reliable. However, in practical problems in the processing of observations, the listed prerequisites for the use of a statistical approach may not be fulfilled. Thus, the independence of the results of observations and experiments, as well as their equal distribution, are most often accepted a priori as a given or a consequence of...
"General assumptions", which is not always true, for example, due to the impossibility of reproducing absolutely identical conditions for observations and experiments.

The observations and experiments themselves can be so unique that the results of applying probabilistic models that focus on the study of mass phenomena are initially questioned. In the case of an obvious small number of observations, it may not be possible to verify the fulfillment of the LSM prerequisites.

Difficulties of LSM application arise at parametric identification of the models which in principle do not allow linearization. In such cases, numerical methods are used, the logic of which is that it is necessary to set the initial approximation \( \Lambda^0 \) for a set of parameter values \( \vec{a} \), and then organizing of "viewing" of the set in discrete points and selecting the one that provides the best value of the optimality criterion entered. Taking into account the fact that the presence of an error in the experimental data \( \{x_i, y_i\} \), is characteristic for practical problems, the values of the parameters \( \vec{a} \) obtained in this way should be considered only quasi-optimal. Moreover, it is obvious that the degree of inaccuracy of the initial data affects the results of parametric identification, distorting the model (1).

In this regard, it may be useful to analyze the entire set of discrete points belonging to \( \Lambda^0 \), from the standpoint of considering the values of both the optimality criterion and other conditions of interest to the researcher. According to the results of this analysis, the boundaries of the area \( \Lambda \), consisting of parameters that provide the required values of the optimality criterion, can be specified, among other things.

Identification of area \( \Lambda \) provides the researcher with the opportunity to analyze all sets of parameter values that meet the specified requirements, and thereby increase the degree of validity of the exact type of model (1) and obtain additional information for model verification. Analysis of the characteristics of the size of this area allows us to judge the degree of uncertainty of the resulting solution, and can be the basis for adjusting the requirements for the desired model.

For the first time, the idea of expediency of definition of areas of an arrangement of required sizes has been stated in L. V. Kantorovich's work [1]. Its relevance has been repeatedly confirmed in practical studies [2]. This work is carried out in line with the development of approaches based on this idea.

The complexity of implementing this approach is associated with obvious computational difficulties due to the following factors:

- large dimension of the vector of model \( \vec{a} \) parameters;
- the number of nodal points;
- a significant amount of calculations carried out for each node.

The dimension of the vector \( \vec{a} \) is determined by the specification of the model (1), and it can be changed only if the researcher makes the appropriate decision. The use of a large number of nodal points contributes to a more thorough viewing of the set of \( \Lambda^0 \), and, as a consequence, to obtain useful information about the structure of the area \( \Lambda \). The density of coverage of the set \( \Lambda^0 \) with nodal points is of particular importance in the problems/tasks of parametric identification of functions having high sensitivity to small changes in parameter values. The amount of required calculations also depends on the specification of the model (1), the type of the selected optimality criterion, and those characteristics that are of interest to the researcher.

One of the ways to solve the problem of optimizing the counting process is to use algorithms of parallelization of calculations. This work presents an algorithm for parallelizing the process of searching for nodal points of a uniform grid for multidimensional areas and its practical usage in solving the problem of parametric identification of the diffuse Bass model.

2. The parallelization method
The parallelization of the calculation process, first of all, is reduced to the fact that it is necessary to identify the possibility of extracting independently calculated parts of the program that can be performed independently of each other. Then performing of a decomposition of the task, that is, the
division of one main task into a number of smaller subtasks in order to perform them in parallel is required. The decomposition can be performed at both at the functional level and at the data level. A functional decomposition is divided into independent subtasks that generally solve the original problem. In data-level parallelization, the data is distributed into subtasks, each of which produces the same type of calculations. In other words, each subtask solves the same problem, but with different data.

After the stage of decomposition of the problem it is necessary to design communications between subtasks [3, 4], that is, to establish links between fragments (subtasks) to solve the "common" problem. At this stage such questions as issues data exchange between parallel parts of the program and transfer of found values to subsequent stages of the task, etc., should be solved. The type of decomposition is determined by the type of parallelization and the problem to be solved.

Since individual blocks of subtasks can solve a certain part of the original problem, these problem fragments are "enlarged" to reduce communicative problems in order to simplify the development of the parallelization algorithm. The last stage of algorithm development is a calculation planning, that is an assignment of enlarged blocks to processors.

Taking into account all the foregoing aspects of decomposition algorithms, this work presents a description of the author's method of parallelization at the data level [5] and its testing in relation to solving problems of variable values in specified areas.

We will consider that a multidimensional area \( \mathcal{A}^0 = [x_1, \bar{x}_1] \times \cdots \times [x_n, \bar{x}_n] \) is specified, in which it is necessary to enumerate the values of the parameters \( x_i \in [x_i, \bar{x}_i], \ldots, x_n \in [x_n, \bar{x}_n] \), satisfying certain criteria. We assume that the values of the parameters \( x_1, \ldots, x_n \) are in the nodes of the uniform grid. Thus, the set of nodal points of a uniform grid built, on the area \( \mathcal{A}^0 \) must be distributed across the processors.

The basis of such parallelization algorithms is the uniform distribution of the set of node points along the streams so that they do not intersect and simultaneously cover all possible variants of the iterated values. The ordering of the sets allows them to be distributed across the streams according to the order numbers. In work [6] we consider the case of forming a set of parameter values using the iteration procedure depending on the iteration step. The work [7] presents the distribution of the checked lines passing through two points by ordering of sequence of points.

For parallelization, it is advisable to arrange all sets of parameters in such a way that there is a one-to-one correspondence between multiple sets and multiple processes: that is, on the one hand the same set must be checked only in one process, and on the other hand, the number of the process must uniquely determine the set tested therein. Obviously, the easiest way to do this is to assign sequence numbers to the sets. A step-by-step description of the set ordering algorithm proposed for the task is given below.

The algorithm of sets ordering.

Stage 1. Select the number (k) of spacing sections \( x_i \in [x_i, \bar{x}_i], i = 1, n \).

Stage 2. Determine the total number of set \( N = n^k \).

Stage 3. Select the number of processes \( j \) and distribute sets by process. To do this, define a number \( M = \text{div} \left( \frac{N}{j} \right) \), where \( \text{div} \) – is the whole part of the division. To the first process we assign sets with numbers from 0 to \((M-1)\), to the second – from \(M\) to \((2 \cdot M - 1)\), etc., to the last – from \((j-1) \cdot M \) to \(N\)-th.

Stage 4. Converts the decimal representation of numbers \( m = 1, N \) to the number system corresponding to the selected parameter \( k \). (The result of this step for each single \( m \) is a sequence of \( n \) numbers, each of which takes values from a set \( \{0, 1, \ldots, k-1\} \).)

Stage 5. If the sequence of numbers obtained in step 4 consists of less than \( n \) numbers, then assign to it the number of zeros in such an amount that in total the sequence contains \( n \) numbers.
Stage 6. Identifying the numbers of the sequence obtained in step 5 with the interval partitioning points \( x_i \in [x_{0i}, x_{ni}], i = 1,n \) by the following rule. The sequence number of the digit in the sequence corresponds to the \( i \)-th parameter number. The value of this digit defines the number of the subinterval from which its center point is taken. The subintervals are numbered from left to right starting from zero (thus the last subinterval will have a \( k-1 \) number).

The essence of the proposed approach for the case when the intervals of parameter setting are divided into two subintervals \( (k = 2) \) illustrates the Figure 1 (points that are involved in the construction of sets are highlighted).

![Figure 1. Illustration of the selection of points from the given intervals at \( k = 2 \).](image)

It is obvious that either a point from the first section or from the second one is included in the set for each parameter. Thus, in the case when a point is selected from the first section, the digit 0 can be mapped to it, and if from the second – the digit 1. With this approach, each set of parameters will correspond to a set of numbers consisting of 0 and 1. This set of numbers can be identified with a number in binary number system. Thus, each set of numbers, when dividing the initial parameter setting intervals into two subintervals, will correspond to a number written in the binary number system. In other words, set \{ 00...00, 00...01..., 11...11 \} will represent all sets in a binary numeral system (number 00...00 will be minimum, and 11...11 – maximum in this set). The first set (corresponding to record 00...00) contains discrete points \{\( x_i, i = 1,n \)\}, each of which represents the center of the first subinterval in the field of the corresponding parameter; The second set (corresponding to record 00...01) consists of discrete points, the first \((n-1)\) of which are the centers of the first subintervals, and the last - the center of the second subinterval of the corresponding parameter value areas; etc. In the range from 00...00 to 11...11 various combinations 0 and 1 are covered. To organize parallelization, it is enough to convert these numbers into a decimal number system: \{00...00, 00...01..., 11...11\} \( \rightarrow \mathbb{0}, 1, ..., 2^n \}_{0,1} \), and then distribute sets to each process according to their sequence numbers. If the intervals on which the parameters are defined are divided into 3 sections, in each of which points are selected, then a set of numbers consisting of 0, 1 and 2 can be matched to each set of parameters. A similar approach can be used when partitioning of a larger number of sections. However, it should be noted that splitting the interval of setting parameters from two sections to ten, the sequence numbers of the sections will be single-digit. Therefore, in the sequence 11 ... 01, each component corresponds to the number of one section from which the point will be taken. And when dividing the intervals into eleven or more subintervals, the corresponding members of the sequences of their numerical representation will also have dimension \((n-1)\), but two-digit numbers denoting the number of one section rather than two will also occur. For example, at \( k = 13 \), sequence 11 12... 10 8, means that the first point must be taken from the 12-th subinterval, the second from the 13-th, the penultimate from the 11-th, and the last from the 9-th.

Thus, in the numerical representation of the set, only the numbers used in the counting system corresponding to the number of sections by which the intervals are divided are involved. That is why this approach can be used when dividing the variable setting intervals by any number of sections. When using the above algorithm, there is no need to generate sets in advance. You can organize its implementation as follows: send a range of numbers to the process for verification (the decimal
analogue of numbering sets), and already in the cycle, within each process, convert this number into the appropriate number system, design the appropriate set, check the set for compliance with the criterion of the task, restrictions, conditions, etc., and either "remember" this set, or perform any operations on it, depending on the purpose of the study. This implementation is convenient because there is no need to store all sets in memory, overloading the system, and you can perform actions only on those sets that satisfy certain characteristics. The main advantage of the developed algorithm is that the problem can be parallelized to the number of processes that will be most effective for each individual computer system, which will allow to accelerate the solution of the problem [3].

Note that this algorithm does not require communication between subtasks because they are independent. However, there must be a main processor that will perform the distribution between other processors of sets of node points. Consolidation is also not required for this type of task, because all subtasks are similar and perform the same operations, but with different data. The planning of calculations consists of the distribution of sets by ordinal numbers.

3. Problem definition

We will test the presented method of parallelization for the model of quantitative analysis of innovative processes developed within the framework of the concept of diffusion of innovation [8–10]. According to this concept, the process of spreading innovations due to the high complexity of economic systems is treated similarly to the existing ideas about diffusion processes in the natural sciences, with the difference that the participants in the diffusion process are the subjects – innovators and imitators. The former are able to take risks when choosing innovations, the latter adopt innovations as they enter the market.

The basic models for the study of diffusion of innovative processes can be divided into models taking into account innovative buying behavior, taking into account imitation buying behavior and taking into account both innovative and imitation buying behavior. The most famous is the Bass model [8], which is a representative of the third group. According to this model, the consumption of innovations is described by the differential equation:

$$\frac{dF}{dt} = p + (q - p)F - qF^2,$$

where $F$ is the number of users of the innovation, expressed in fractions of units, $p$ is the parameter characterizing the degree of penetration of innovations, due to innovators, $q$ is the parameter characterizing the degree of penetration of innovations, due to imitators.

The solution of equation (2) is written as:

$$F = \frac{C + \frac{p}{q} e^{-(p+q)t}}{C - e^{-(p+q)t}},$$

where $C = \frac{q}{F_0 - 1}$, $a F_0$ – initial number of innovation consumers.

Depending on the parameters $p$ and $q$, the increasing rate of function $F$ varies (Figure 2). Knowing them, you can not only predict how long the process of spreading innovation will be. (At the Figure 2 the curve "1" at $p=0.02$ and $q=0.7$; "2" – at $p=0.1$ and $q=0.1$; "3" – at $p=0.01$ and $q=0.1$.)
Figure 2. Graphical representation of the model (3) at $F_0 = 0.088$.

It should be noted, that the analysis of the magnitude $\frac{q}{p}$ can contribute to understanding the qualitative state of the sphere of innovation. For example, F. Bass in his research conducted for certain types of innovative products, on the basis of this ratio, found that the penetration of innovations due to imitators is from 9.0 to 82.4 times higher than the penetration taking place due to innovators [8]. E. Rogers, a follower of F. Bass, came to similar conclusions in his research [10]. He found that the number of consumers of innovative products of the second and subsequent waves can exceed the number of innovators by 39 times. The results of research of the classics of diffuse models of innovation have quite logical explanations and can serve as a guide for research.

Let’s set the problem of parametrical identification of the model (3) describing the state of the sphere of innovation development in the Russian Federation (RF). As a simulated variable we will select an indicator characterizing the involvement of organizations in innovation activity - innovation activity of enterprises (Table 1). This choice is explained as follows: innovation itself will be demanded by enterprises if it brings them benefits. At the same time, it is obvious that among enterprises there are both those that are more prone to risk (early consumers of innovations, or innovators), and those that follow, guided by the experience of the first (imitators).

Table 1. Innovative activity of Russian enterprises in 2000-2016, shares of unit.

| Year | $F_0$ | Year | $F_0$ |
|------|-------|------|-------|
| 2000 | 0.088 | 2009 | 0.093 |
| 2001 | 0.085 | 2010 | 0.095 |
| 2002 | 0.090 | 2011 | 0.104 |
| 2003 | 0.095 | 2012 | 0.103 |
| 2004 | 0.096 | 2013 | 0.101 |
| 2005 | 0.097 | 2014 | 0.099 |
| 2006 | 0.099 | 2015 | 0.093 |
| 2007 | 0.100 | 2016 | 0.084 |
| 2008 | 0.094 | ---  | ---  |

We consider acceptable those sets of model parameter values (3) for which the average approximation error does not exceed the specified level $\bar{A}^i$:

$$
\bar{A} = \frac{1}{2} \sum_{i=1}^{2} \frac{F^{\text{exp}}_i - F^{\text{calc}}_i}{F^{\text{exp}}_i}.
$$

(4)
It is from these that an information set $\Lambda^*$ must be formed (i.e., a set composed of mesh nodes that satisfy the requirements (4)). The best set of parameter values $\{p, q\} \in \Lambda^*$ is one in which the average approximation error takes the minimum value.

4. Computing experiment

On iteration 1, the initial approximation was given by a set $\Lambda^0 = p \in [0.001, 0.05] \times q \in [0.001, 0.05]$. The selection of boundaries for the set $\Lambda^0$ was based on the actual data available (Table 1). A uniform grid on this set was formed by dividing the intervals of the $p$ and $q$ parameter values into 400 subintervals $\overline{A^0} = 7.2\%$.

The parameter sets were found using the presented parallelization algorithm. The search involved $2^{400}$ sets, which were selected according to the requirement of the problem (4). Each of the numbers at the interval $[0, 2^{400}]$ was set according to the set built as a result of the converting of this number into a 400-th counting system. For example, the parameter set number 1600 according to the above algorithm was represented by a pair of numbers $(0, 4)$ (i.e., variable $p$ was taken from the 1st line and variable $q$ was taken from the 4-th), and the parameter set number 2536 was represented by a pair of numbers $(136, 6)$ ($p$ of the 136th line, $q$ of the 6th). Thus, all possible combinations of section numbers were obtained, which provided a complete search of nodes of a uniform grid. According to the results of calculations carried out using parallelization into two processes, 230 sets of values of parameters $p$ and $q$ were included in the information set $\Lambda^*$. Their analysis showed that

$$\min_{\Lambda^*} p = \min_{\Lambda^*} q = 0.001, \quad \max_{\Lambda^*} p = 0.00145, \quad \max_{\Lambda^*} q = 0.00474.$$ 

Taking into account the above information, many of the initial approximations $\Lambda^0 = \{p \in [0.001, 0.003] \times q \in [0.001, 0.005]\}$ were changed on iteration 2. The interval for each parameter was divided into 20 subintervals. At this iteration, 67 sets of parameter values were included in the information set $\Lambda^*$ (Figure 3, "1" – set $\Lambda^0$, "2" – uncertainty $\Lambda$, "3" – information set $\Lambda^*$).

![Figure 3. Geometric interpretation of iteration steps 2.](image)

Here, the term "uncertainty area" ($\Lambda$) refers to a plurality of all sets of values that ensure the validity of the criterion (4). At the same time, the fairness of the investment $\Lambda^* \subset \Lambda$ is obvious.

The analysis of the information set revealed that the best value of the average approximation error is:

$$\min_{\Lambda^*} \overline{A} = 5.23\%.$$
This value is achieved at $p = q = 0.001$. This allowed the exact view of the model to be established (3):

$$F = \frac{1.19 - e^{-0.002r}}{1.19 + e^{-0.002r}}. \quad (5)$$

To estimate the parallelization rate – one of the main characteristics of the algorithm on the set $\mathcal{A}^0 = \{p \in [0.001, 0.003] \times \{q \in [0.001, 0.005]\}$, computational experiments were carried out using the Delphi environment. In this environment, you can create multithreaded software products. Each thread is considered by the operating system as an independent task, so when you run such a program on single-processor computers, there is an imaginary parallelism of programs (pseudo-parallelism). This is achieved because the operating system allocates certain quanta of time to each application to ensure multitasking, and because each thread for the operating system is a separate process, separate time is allocated for them. In other words, threads are started for a very short period of time and then pushed out by other threads. If the computer has two or more processors, the parallelism will be real. Since the described parallelization algorithm was based on decomposition from data that was equally distributed across threads, the execution time of each thread would be the same.

The estimation of parallel acceleration, that is, the increase in the speed of performing calculations on a given number of threads, as opposed to single-threaded execution, is made using the formula:

$$S_p = \frac{t_1}{t_p}, \quad (6)$$

where $P$ is the number of threads, $t_p$ is the execution time of the non-parallelized algorithm, $t_p$ is the execution time of the algorithm on $P$ threads.

Parallel efficiency, which allows to evaluate the efficiency of the algorithm depending on the number of threads, is calculated as follows

$$E_p = \frac{S_p}{P}, \quad (7)$$

The "ideal" outcome of evaluating the efficiency of parallelization is a directly proportional increase in the speed of calculations with an increase in the number of threads, that is $S_p = P$. However, parallelization does not always provide such a result. This may be due to the construction of the algorithm itself (for example, not all threads may be loaded equally), and the time delays that occur in connection with access to global data. Such data is available to all threads, but you must wait for your queue to access it directly.

The value restriction $S_p$ is given by Amdahl's law [11]:

$$S_p \leq \frac{1}{\alpha + \frac{1-\alpha}{P}}, \quad (8)$$

where $\alpha$ is the proportion of consecutive calculations in the total volume of calculations.

Having an experimental value of $S_p$ for some fixed number of threads $P$, it is possible to determine the fraction $\alpha$, and then evaluate the expediency of splitting the computational procedure into a larger number of threads.

The parallelization rate evaluation experiment was conducted on a single-processor computer and a 4-core Intel® Core™ i5 CPU 760 @ 2.8GHz. The ranges of values $p$ and $q$ defining the set were divided into 1000 and 1500 intervals. Analysis of experimental results (Table 2) showed that in each of the cases there was a reduction in the running time of the program. At the same time, on a computer
with a single processor, the achieved changes in the speed of calculations were obtained only at the expense of imaginary parallelism.

Table 2. Multi-threaded application rate evaluation results.

| Processor type | Quantity of intervals | Time, c without parallelization | Time, c with two parallel threads ($P = 2$) | Parallel acceleration, $S_2$ | Parallel effectiveness, $E_2$ |
|----------------|-----------------------|-------------------------------|---------------------------------|-----------------|-----------------|
| One-nuclear    | 1000                  | 4.52                          | 3.18                            | –                | –               |
|                | 1500                  | 9.20                          | 6.53                            | –                | –               |
| The 4-th nuclear | 1000                  | 2.75                          | 1.40                            | 1.96             | 0.98            |
|                | 1500                  | 5.65                          | 2.90                            | 1.95             | 0.98            |

According to Amdahl’s law (8), $\alpha = 0.025$, it follows that with more parallel flows, acceleration will increase (so, $S_3 = 2.86$, $S_4 = 3.72$) and efficiency will decrease ($E_3 = 0.95$, $E_4 = 0.93$). Thus, increasing the number of threads is not appropriate. In general, the presented results show that the parallelization algorithm is built correctly and provides a reduction in time in proportion to the number of threads started for calculations.

5. Discussion of simulation results

The results of the calculations made it possible to draw the following conclusions.

1. The number of consumers of innovations of the first and second waves in the Russian Federation is extremely insignificant. This follows from the fact that in the summary of all calculations, the ranges of variation of the parameter values were

$$p \in [0.001, 0.00145], \quad q \in [0.001, 0.0048].$$

2. There are currently no trends to significantly change the existing indicators (Figure 4, "1" – actual values; "2" – calculated values according to the model (5)).

![Figure 4. Innovative activity of Russian enterprises.](image)

3. The qualitative state of the sphere of distribution of innovations in the RF due to the fact that

$$\min_{P} q \approx 0.71, \quad \max_{P} q \approx 4.8,$$

is significantly different from the estimates obtained by the classics of diffuse processes. This is primarily due to the extremely weak activity of the simulators.
6. Conclusion

The paper presents a method of parallelization of the process of sorting nodal points in multidimensional regions. Its distinctive feature is universality, since the process of sorting the nodal points of multidimensional areas is not tied to the process of checking them for compliance with the conditions and other functional features of each individual task.

Due to the fact that it is possible to specify a parameter characterizing the number of parallel processes \(j\), the degree of parallelism of the algorithm is actually limited only by the potential of the computer used. Thus, regardless of the technical characteristics of the equipment used, it is possible to ensure the best correspondence between the parallelism of the algorithm and the parallelism of the computer.

Numerical experiments on the study of the parallelization rate in the framework of solving the problem of parametric identification of the Bass model have shown that the presented method provides the best indicators of parallel efficiency when dividing the calculation process into two streams. The results obtained indicate the expediency of using the algorithm presented in solving problems in which searching the nodal points of grids in multidimensional regions are required.

References

[1] Kantorovich L 1962 On some new approaches to computational methods and processing of observations Sib. Math. J. 3 701–09
[2] Spivak S, Ismagilova S and Kantor O 2014 The areas of uncertainty in the mathematical theory of measurement analysis Control Systems and Information Technologies 58 17–21
[3] Karpov V 2010 Introduction to Parallelization of Algorithms and Programs Computer research and modeling 2 231–72
[4] Starchenko A and Bertsun V 2013 Parallel computing methods (Tomsk: Tomsk University Press)
[5] Yusupova G, Kantor O and Spivak S 2015 Surveying the parameters defined in the multidimensional area by paralleling Certificate of state registration of a computer program no. 2015611655, 2015
[6] Demjanovich L and Kolegov K 2016 Innovative development of the transport and logistics complex of the Caspian region: materials of the 5-th modern scientific and practical conference (Astrakhan: Caspian Institute of Sea and River Transport) I 7–12
[7] Fedotova S, Seredin O and Kushnir O 2016 The algorithms of adjustment of reflection symmetry axis found by the skeleton primitive sub-chains comparison method Izvestiya TulGU 11 99–111
[8] Bass F 1969 A New Product Grows for Model Consumer Durables Management Science 15 215–27
[9] Fourt L and Woodlock L 1960 Early prediction of Market success for new grocery products J. of Marketing 26 31–38
[10] Rogers E 2003 Diffusion of Innovations (New York: Free Press)
[11] Amdahl G 1967 AFIPS Conference Proceedings (AFIPS Press) 30 483–85