

The Use of Petri Networks for the Optimal Complexity Empirical Models Building Parallel Algorithm Synthesis on the Genetic Algorithms Basis

M. I. Gorbichyck¹, O. T. Bila², T. V. Humeniuk³

^{1,2,3}Ivano-Frankivsk National Technical Oil and gas University, str. Karpatskaya 15, Ivano-Frankivsk, 76019, Ukraine
(¹gorb@nung.edu.ua, ²ksm@nung.edu.ua, ³taraskm@gmail.com)

Abstract- Conducted parallelism analysis of the optimum complication empiric models building method on the genetic algorithms basis using Petri network. It is rosined that internal parallelism which enables to develop the effective program of algorithm realization on the parallel structure computer system has such algorithm, and it, in its turn, will result in machine time usage reduction during his practical realization.

Keywords- Empirical Model, Petri Net, Internal Parallelism, Level, Computational System, Parallel Structure

I. INTRODUCTION

Data obtained in the study of a complex (e.g., environmental) processes number, usually presented in the form of time series, which is described by the empirical models [1]. It is assumed that the structure of the model is somehow chosen and the identification task is only to determine the parameters of such a model, which in most cases is solved using the least squares method. Obviously, the model matching and empirical data accuracy depend on the model structure itself. For example, when choosing a model in the regression equation form with an increase in its members, the error, determined at all experimental points, monotonically decreases. As soon as the number of regression members becomes equal to the experimental points number, the error becomes zero. In the case when the experimental points number is greater than the regression model members number, one can always obtain an infinitely large number of empirical models [2], that is, the least squares method generates an infinitely large models number for a given numerical series. This statement is true only if one internal criterion is used to select models.

According to the Gödel's incompleteness theorem [3], it is essentially impossible to find an optimal and unified model using the sequence points that have been already used to find the coefficients by the least squares method. Only external additions allow finding a single model that is optimal for this complexity criterion [4].

The model structure choice is based on the experimental data numerical series types. There is a question of choosing the

basic function for a mathematical model. For example, for oscillatory processes with non-frequencies with zero average value is recommended to use harmonic models, and if the oscillatory process tends to increase or decrease, then a model representing the sum of the polynomial trend and the harmonic balance should be formed.

In the general case, the construction of empirical models is carried out by selecting a function from a given class, which in a certain sense best describes the experimental data. To solve the given task, acad. A. G. Ivakhnenko proposed a number of methods, united by a common name - an inductive method of the models self-organization that generates three structures of algorithms [5, 6]: multi-row selection algorithm of the group accounting arguments method (GMDH), combinatorial algorithm GMDH with a complete overview of all possible models from a given polynomial and multi-row algorithm, where the coefficients equal to zero of a definite polynomial are zeroed on each row of selection. The multichannel GMDH algorithm has arguments for some of the intermediate values generated by its nature, and the combinatorial GMDH algorithm for its implementation requires significant computational costs [7], which grow exponentially with an increase in the number of arguments of the empirical model.

To reduce the computational cost and expand the scope of empirical models, for the implementation of which combinatorial algorithms are used, the method of constructing empirical models using genetic algorithms was proposed in [8, 9]. Later, a similar algorithm for constructing empirical models was described in [10].

II. POLYNOMIAL EMPIRICAL MODELS SYNTHESIS METHOD BASED ON GENETIC ALGORITHMS

The implementation of the models self-organization inductive method is carried out in stages: the first stage is the applicant's models generation (in a certain order increasing their complexity); the second stage is the choice of the best model by the minimum of one of the selection criteria. As a selection criterion, choose [5]: regularity criteria

$$\Delta^2(B) = \frac{\sum_{i=1}^{N_A} (y^{(i)}(B) - \bar{y}^{(i)}(B))^2}{\sum_{i=1}^{N_A} (y^{(i)}(B))^2} \quad (1)$$

and bias

$$\Delta^2(A, B) = \frac{\sum_{i=1}^N (y^{(i)}(A) - y^{(i)}(B))^2}{\sum_{i=1}^N (y^{(i)})^2} \quad (2)$$

The inductive method of self-organizing models assumes that all the data obtained as a result of the experiment, are divided into two parts, educational and verifiable. Then in formulas (1) and (2) $y^{(i)}(A)$, $y^{(i)}(B)$ is the output values of the model calculated on the sets of experimental values N_A i N_B . If the criterion of regularity (1) is selected, then the following division of experiment data is removed [4]: $N_A = 0,7N$ i $N_B = 0,3N$, and when choosing the criterion (2) is $N_A = 0,5N$ and $N_B = 0,5N$.

Most often, the functional relationship between the output of an object and its inputs are chosen as follows, when building empirical models:

$$y = \sum_{k=0}^r a_k f_k(\bar{x}) \quad (3)$$

Where a_k , $k = \overline{1, r}$ are parameters of the model; $\bar{x} = (x_1, x_2, \dots, x_n)^T$ is a vector of input quantities.

We choose polynomial degree m as an empirical model

$$y = \sum_{i=0}^{M-1} a_i \prod_{j=1}^n x_j^{s_{ji}} \quad (4)$$

where a_i are the coefficients of a polynomial; s_{ji} are arguments degrees that must satisfy the limit:

$$\sum_{j=1}^n s_{ji} \leq m$$

The polynomial members 'number (4) is determined by the following formula [8]:

$$M = \frac{(m+n)!}{m!n!} \quad (5)$$

The model (4), where the part of the coefficients a_i , $i = \overline{1, M}$ acquire zero values, is obtained from the complete polynomial using the combinatorial method of optimal complexity models synthesis (4). Since the optimal complexity empirical models synthesis task with the use of GMDH multi-row algorithm has a large dimension, the genetic approach is applied to remove the large dimension. The ordered sequence is created, where on the i -th place will be a unit or zero,

depending on whether the model parameter a_i , $i = \overline{1, M}$ (4) is different from zero or will be zero. In the theory of genetic algorithms, such an ordered sequence is called the chromosome, and the atomic element (unit or zero) of the chromosome is a gene. A set of chromosomes forms a population. An important concept in the theory of genetic algorithms is the function of adaptation, which determines the degree of the separate individuals' suitability in the population. It allows the entire population to select those individuals that are most adapted, that is, those that match the least (highest) value of the adaptation function. The selection criterion (1) or (2) is chosen as a function of adaptation in the optimal complexity models synthesis task.

Thus, based on the observation data of an object or the process, and on the selected model structure (4), the genetic algorithm forms the set of most perspective structures and selects the best of them using the selection criterion (1) or (2).

Formalized recording of the algorithm will be as follows:

$$GA = \langle I, CP, SA, SE, NP \rangle \quad (6)$$

Operators belonging to the formalized record (6) are executed in the following sequence.

The initialization operator I randomly generates a population from I individuals each of which is a chromosome with length M . The number of genes in each chromosome is calculated by the formula (5).

Operator CP evaluates the degree of the chromosome adaptation in the I powered population by calculating the selection criterion value (1) or (2) for each chromosome. The matrix F with $N \times M$ size, that is divided into two matrices F_A and F_B , is formed from the partial functions (regressions) with the model coefficients a_i , $i = \overline{0, M-1}$ (4). The matrices F_A and F_B have M columns each, and the rows number of F_A and F_B matrices is determined by the capacity of the training set N_A and the verifiable set N_B . The i -th column is deleted from F_A matrix, if the i -th chromosome position is zero; if it's one, then the corresponding column remains unchanged. As a result, we get the \tilde{F}_A matrix where c columns are removed (by the number of zeros in the chromosome). The size of such a matrix will be $N_A \times (M - c)$. Similarly, the \tilde{F}_B matrix with the $N_B \times (M - c)$ size is obtained. Nonzero coefficients a_{Aj} , $j = \overline{0, M - c - 1}$ of a model (4) are calculated by solving the normal Gaussian equation on the points set N_A

$$\tilde{M}_{F,A} \bar{a}_A = \tilde{F}_A^T \bar{Y}_A \quad (7)$$

where $\bar{a}_A = (a_{A0}, a_{A1}, \dots, a_{A, M-c-1})^T$ is a vector of nonzero parameters of the model, which is associated with the next chromosome; $\tilde{M}_{F,A} = \tilde{F}_A^T \tilde{F}_A$. $\bar{Y}_A = (Y^{(1)}, Y^{(2)}, \dots, Y^{(N_A)})$ is the vector of output values of the object in a set N_A .

The values are calculated by the found coefficients \bar{a}_A of a polynomial model on a set of points N_B

$$\bar{y}(B) = \tilde{F}_B \bar{a}_A \quad (8)$$

$\bar{y}(B)$ is known and the adaptation function value $\Delta_j^2(B)$, for each chromosome from the initial population is calculated by formula (1).

In the case when the displacement criterion (2) is used as a function of the adjustment, the equation (7) that is solved by the Gaussian method related to the parameters \bar{a}_A is created.

Then $\bar{y}(A) = \tilde{F}_A \bar{a}_A$ is calculated. Obtained values $\bar{y}(A)$ and $\bar{y}(B)$ give possibility to find values $\Delta_j^2(A, B)$, $j = \overline{1, I}$ for each chromosome from the population.

The operator *SA* checks the execution of the algorithm stop conditions. Determine

$$\Delta_m^2(B) = \min_j \Delta_j^2(B) \quad (9)$$

or

$$\Delta_m^2(A, B) = \min_j \Delta_j^2(A, B) \quad (10)$$

If the minimum value (9) or (10) of the (1) or (2) selection criterion does not exceed a given value E , the calculation stops. The calculations can also stop if there is no significant reduction of the adaptation function as a result of the algorithm execution, or in the case when a given number of iterations is executed.

When one of the three conditions is fulfilled, the chromosome ch^* , which has the condition (9) or (10) fulfilled, is selected from the regular population. This chromosome defines the structure of the optimal complexity model and forms the F^* matrix in such a way that the columns which are associated with the null values of the corresponding genes are removed from the original matrix F . The calculation of the model parameters (4) is performed on the whole points set N using the LSM.

The operator *SE* carries out the chromosomes selection using the defined by the operator *SA* adaptation function values that allows selecting those chromosomes which will participate in the descendants' creation for the new population. Such selection is performed with the natural selection principle when the chromosomes with the best adaptation function values (1) or (2) have the highest chances of a new population creating. The most common selection methods are the roulette method and the tournament method [11]. The roulette method can be used when the adaptation function is positive, that makes it suitable only for maximization tasks. The tournament method can be used for both, maximization and minimization tasks.

During the tournament selection, all the chromosomes are divided into subgroups with the following selection from each chromosomes group with the best adaptation. Subgroups may

have arbitrary size, but most often the population is divided into subgroups with 2 - 3 individuals in each.

The operator *NP* carries out the formation of a new descendants' population using two basic processes - crossing and mutation. The probability of crossing P_c is selected from the interval [0.5; 1] and the probability of the mutation P_m lies within [0; 0.1]. The crossing process is carried out over selected chromosomes belonging to the parent pool. To do so, a pair of chromosomes is randomly selected from the individuals population. A random number is generated in the interval [0; 1] and, if its value does not exceed P_c , the pair of chromosomes is crossed. In the opposite case, the pair of chromosomes remains unchanged. If there is a chromosome crossing, then the gene position (locus), that determines the crossing point, is determined for each pair. If the parent pool's chromosome has m genes, the crossing point is a natural number less than m . Therefore, the crossing point fixation is reduced to the integer random selection from the interval [1; $m - 1$].

As a crossing process implementation result, a new pair of descendants is formed from a couple of parents as follows: the first descendant in a chromosomes pair, which is on the positions from 1 to L_c consists of the first parent's genes, and for the positions from $L_c + 1$ to m it consists of the second parent's genes; the second descendant in a chromosomes pair consists of the second parent's genes standing at positions from 1 to L_c , and for positions from $L_c + 1$ to m it consists of the first parent's genes.

Those parent pool genes (by replacing the unit by zero and vice versa) that have the drawn number less or equal P_m are suitable for mutations.

When the operator *NP* is performed, the algorithm keeps working starting from the operator *CP*.

III. PARALLEL ALGORITHM SIMULATION USING PETRI NETS

The implementation of the optimal complexity models synthesis algorithm showed that it yields a significant profit over time in comparison with the combinatorial method of the models selection. But the machine time usage increases with the dimension increase of the empirical models synthesis task. In [12] it is shown that such an algorithm has internal parallelism. This algorithm property allows reducing the machine time usage for its implementation.

An efficient simulation tool for parallel processes is Petri networks. The Petri nets construction based on some concepts set and rules that allow building bipartite graphs, which includes two types of tops: transitions tops T and positions tops that are connected with the arcs K by the certain functional rules S [13,14].

The Petri network that is shown in Fig. 1 simulates the optimal complexity models synthesis algorithm for the case when the regularity criterion (1) is selected. Petri network (Fig.

1) contains a predicate represented by positions p_ε , p_a and $p_{\bar{\varepsilon}}$, transitions $t_{a,1}$ and $t_{a,2}$, which determines the algorithm completion in addition to the main operators.

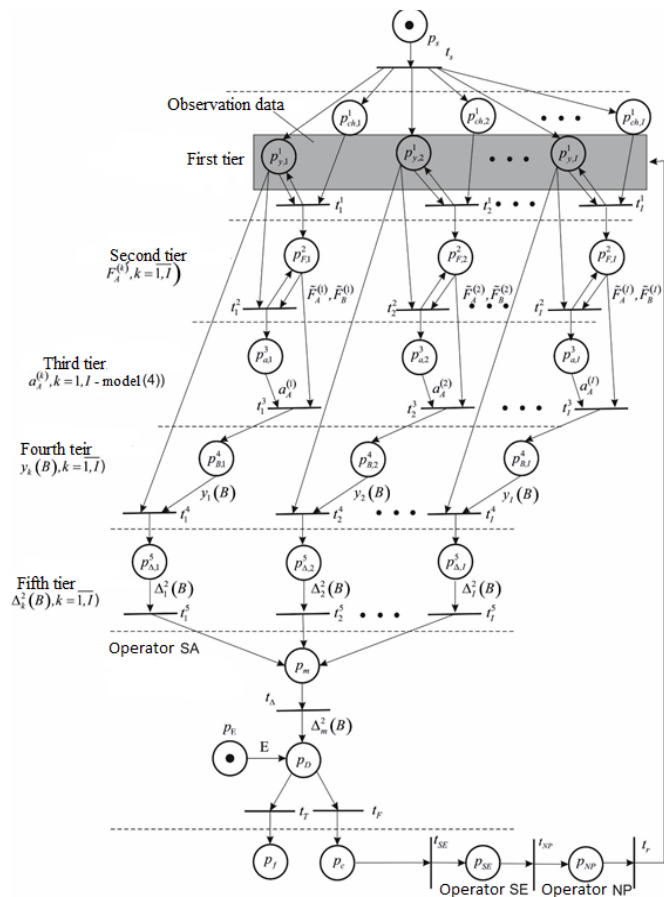


Figure 1. Simulation of the optimal complexity models synthesis algorithm using Petri networks

The Petri network structure (Fig. 1) can be divided into tiers. There are no connections between the vertices in the middle of each tier. The outputs of each tier are arguments for the next tier. The zero tier of the Petri network contains an empty position p_s marked by a marker.

The first tier of the Petri network contains positions $p_{y,k}^1, k = \overline{1, I}$ that simulate the process of creation of the matrix of the observations $X = \{x_i^{(j)}\}_{N \times n}$ and a observations vector \bar{Y} with components $y_i, i = \overline{1, N}$, where N is the number of observations by the input value; n is the number of empirical model arguments (4), and positions $p_{ch,k}^1, k = \overline{1, I}$, each of which is associated with the chromosome formed by the operator I .

The second tier of the Petri network is represented by positions $p_{F,k}^2, k = \overline{1, I}$ that are intended to form matrices $F_A^{(k)}$ and $F_B^{(k)}, k = \overline{1, I}$ with the help of an operator CP .

The third tier contains positions $p_{a,k}^3, k = \overline{1, I}$ that simulate the calculating process of the empirical model coefficients (4) by the formula (8).

The fourth tier is presented by positions $p_{B,k}^4, k = \overline{1, I}$, each of which is intended to calculate $y_k(B), k = \overline{1, I}$ on the veritable points set N_B .

The fifth tier contains the positions $p_{\Delta,k}^5, k = \overline{1, I}$ that initiate the calculating process of the regularity criterion (1) for each chromosome that is generated by the operator I .

The position p_m corresponds to the process of selecting the most adapted chromosome from the relatives pool I . Such a choice is made using the regularity criterion (1). The position p_E has a permanent marker and it specifies the required accuracy of the calculations E ; the position of the free choice p_D is responsible for the formation of the condition. If the last one is executed, the transition t_T will work. In case, when a condition $\Delta_m^2(B) > E$ exists, the transition t_F becomes allowed and the calculation process continues with the execution of CP operator.

The algorithm (Fig. 1) starts after the empty position p_s is activated that causes the marker movement over the network. As a result, all the entry positions $In(t_k), k = \overline{1, I}$ of the first tier are marked with markers that will trigger the transition $t_k^1, k = \overline{1, I}$. Since the results of observations for input and output values are used on the first, second and fourth tiers, the markers must be permanently located in the positions $p_{y,k}^1, k = \overline{1, I}$. This is achieved by the fact that the positions $p_{y,k}^1, k = \overline{1, I}$ are both input and output positions of transitions $t_k^1, k = \overline{1, I}$ [14], that is $p_{y,k}^1 \in O(t_k^1) \cap In(t_k^1), k = \overline{1, I}$, and after their activation, the preliminary marking of positions $p_{y,k}^1, k = \overline{1, I}$ is retained.

When the transitions $t_k^1, k = \overline{1, I}$ are triggered, markers appear in the second tier positions $p_{F,k}^2, k = \overline{1, I}$. The situation is similar to the situation of the first tier: the matrices \tilde{F}_A and \tilde{F}_B are used to calculate the model parameters (4) by solving the equation (7) (third tier) and to calculate the values $y_k(B), k = \overline{1, I}$ by the formula (8) (fourth tier).

Therefore, markers will always be in positions $p_{F,k}^2, k = \overline{1, I}$ after the $t_k^1, k = \overline{1, I}$ transaction is performed.

Such markers movement occurs consistently from the previous to the next tiers, that leads to the corresponding transitions activation (Fig. 1).

When the condition $\Delta_m^2(B) \leq E$ is fulfilled, a transition t_T triggers and a marker will appear in the position p_f , indicating the calculation process completion. In the case when the given accuracy of the optimal complexity model synthesis task solution on the genetic algorithms basis is not achieved $\Delta_m^2(B) > E$, the transition to the operator SE (Fig. 1) is carried out and the calculations continue until the algorithm stop condition is fulfilled.

It should be noted, that two other algorithm stop conditions are not shown in Fig1. They are similar to the conditions specified by the operator SA , and they are executed sequentially after the operator SA execution.

Petri network that is shown as a dipole graph (Fig. 1), sets the initial network state (network static). When the transition t_s is activated (the beginning of the algorithm), the marker moves the network from one tier to another. Such a change in the Petri network state is called network dynamics [14]. The Petri network behavior is reflected through a marking diagram [14]. Diagram given in Fig. 2, corresponds to Petri network that is shown in Fig. 1. The network conditions change is characterized by a marking mechanism that defines the active transition activation rule. When the marker moves from position p_k to position p_{k+1} , that is displayed by $1 \rightarrow 0$ or $0 \rightarrow 1$. This means that when the active transitions are triggered, the marker is removed from the original positions in its input positions (or vice versa) (Fig. 2). The transitions activation and the corresponding markers removal from positions start when an appropriate event occurs due to the next operator from the tuple execution (6).

According to the diagrams of markings (Fig. 2), it is possible to determine the sequence of transitions that are the trajectories of the computing process: $t_k^1 t_k^2 \dots t_k^5 t_\Delta t_T \oplus t_F t_f$ or $t_k^1 t_k^2 \dots t_k^5 t_\Delta t_T \oplus t_F t_{SE} t_{SP} t_r t_k^1 t_k^2 \dots t_k^5 t_\Delta t_T \oplus t_F \dots t_f$, where $t_T \oplus t_F$ means that one of the transitions t_T or t_F is triggered, depending on the condition fulfillment $\Delta_m^2(B) \leq E$ or $\Delta_m^2(B) > E$.

The Petri network diagrams analysis (Fig. 2) shows that it is 1-limited, because the condition $\mu(p_i) = 1$ is fulfilled on its achievable states set in all positions. The 1- limitation network property determines its safety [14]. Considering that each Petri network transition $t \in T$ (Fig. 2) is potentially active, such a network is alive [14].

In addition, Petri network (Fig. 2) is stable. The network resilience property stems from the fact that the activation one of the transitions $t_1^1, t_2^1, \dots, t_l^1$ and $t_1^2, t_2^2, \dots, t_l^2$ does not remove activity from other transitions.

Thus, Petri network (Fig. 1) has the properties of 1-limitation, safety, and also is alive and stable, which determines the effectiveness of the optimal complexity empirical models building algorithm on the genetic algorithms basis.

The efficiency of the developed algorithm is confirmed by the construction of empirical polynomial models for a processes number taking place in such fields as gas transportation [8], drilling of wells for oil and gas [15] and ecology [16].

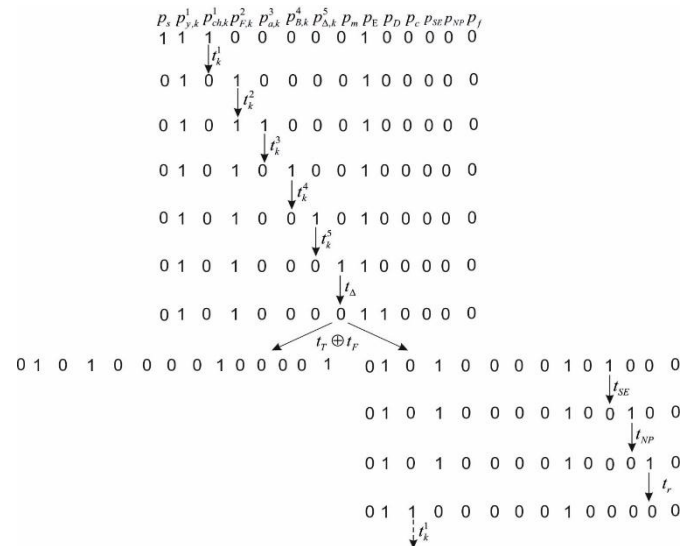


Figure 2. Petri Network Marking Diagram

IV. CONCLUSION

Petri network that is divided into five tiers is constructed for the optimal structure empirical models synthesis algorithm based on genetic algorithms that have internal parallelism. The positions number in each tier is determined by the chromosomes amount generated in the first step of the algorithm. The Petri network positions number are associated with such operations as a linear algebraic equations system solving and multiplying the matrix by a vector that may also have internal parallelism. Internal parallelism is a characteristic property of the algorithm and does not depend on the computing system where such an algorithm is implemented. The presence of internal parallelism in the algorithm enables the implementation of such an algorithm on a computing system with a parallel architecture that will reduce the machine time usage.

REFERENCES

- [1] Box J. Analysis of time series. Forecast and control, per. from english M.: World, 1974. 406p.
- [2] Ivakhnenko AG Long-term forecasting and management of complex systems. K.: Technics, 1975. 312 p.

- [3] Uspensky VA Theodel's Theorem on the incompleteness of M.: Nauka, 1982. 110 p.
- [4] Ivakhnenko A. G., Stepashko V. S. Impact Tolerance Modeling Kiev: Sciences. thought, 1985. 216 s.
- [5] Ivakhnenko AG, Lapa VG Foresight of random processes. Kiev: Naukovadumka, 1969. 420 p.
- [6] Ivakhnenko AG The inductive method of self-organization of models of complex systems. Kiev: Naukovadumka, 1981. 286 p.
- [7] Gorbichiuk M.I., Medvedchuk V.M., Lazoriv A.N. Analysis of the parallel algorithm for the synthesis of empirical models on the principles of genetic algorithms. Problems of Management and Informatics. 2016. No. 1. S. 112 - 130.
- [8] Gorbichiuk M.I., Kogutiak M.I., Vasilenko O. B., Shchupak I.V. The method of synthesis of empirical models on the basis of genetic algorithms. Exploration and development of oil and gas fields. 2009. No. 4 (33). 72-79.
- [9] Gorbichiuk M.I., Samulyak S.T., Shchupak I.V. Synthesis of Classification Functions Based on Genetic Algorithms. Artificial Intelligence. 2010. Number 2. S. 24 - 31.
- [10] Moroz O. G. Pertinent algorithm of MSUA with genetic search of optimal model. Control systems and machines. 2016. No. 6. S. 73-88.
- [11] Rutkovskaya D., Pylinsky M., Rudkovskii L. Neural networks, genetic algorithms and fuzzy systems, per. from polsk I. D. Rudinsky. M.: Hot Line-Telecom, 2004. 452 p.
- [12] Gorbichiuk M.I., Medvedchuk V.M., Pashkovsky B.V. Parallelism of algorithm of synthesis of models of optimal complexity on the basis of genetic algorithms., East-European Journal of Advanced Technology. 2014. No. 4/2 (70). Pp. 42-48.
- [13] Peterson J. Petri Networks Theory and Systems Modeling, Per. from englishM.: Mir, 1984. 264 p.
- [14] Marakhovsky VB, Rosenblum L. Ya., Yakovlev AV Modeling of parallel processes. Petri Networks St. Petersburg: Professional Literature, 2014. 400 p.
- [15] Construction of empirical models of the process of deepening of wells taking into account uncertainty, MI Gorbichik, T. V. Gumenyuk, Ya. I. Zayachuk, N. T. Lazorov, Methods and instruments of quality control. 2016. No. 1 (36). Pp. 86 - 94.
- [16] Gorbichich MI, Shufnarovich MA Methods of Artificial Intelligence in the Ecology: Monograph. Latvia, EU: LAMBERT Academic Publishing. 2018. 192 c.

How to Cite this Article:

Gorbichyck, M. I., Bila, O. T. & Humeniuk, T. V. (2019) The Use of Petri Networks for the Optimal Complexity Empirical Models Building Parallel Algorithm Synthesis on the Genetic Algorithms Basis. International Journal of Science and Engineering Investigations (IJSEI), 8(89), 123-128. <http://www.ijsei.com/papers/ijsei-88919-19.pdf>

