

ALGORITHM OF CONSTRUCTING CONTROL MODELS OF COMPLEX SYSTEMS IN THE LANGUAGE OF FUNCTIONING TABLES

ANVAR KABULOV¹, IBROKHIMALI NORMATOV, ANVAR KARIMOV, AND ERKIN NAVRUZOV

ABSTRACT. The algorithm for constructing control models of complex systems should be based on the hypothesis that the research and control processes of complex systems are subject to a single technology. A comprehensive study of objects is needed, starting with preliminary study, obtaining adequate models, process algorithms, and ending with the creation of effective control systems.

Therefore, the algorithm of control processes using modern computer technology requires not only improved management at all levels, but also the creation of new effective methods for the rational and adequate description of objects of research, optimization and construction of a control algorithm.

The general scheme of such an algorithmic approach is described below.

In this paper, we give methods and models of algorithms that are of particular importance in the automation of all stages of system research and the construction of control algorithms. From this point of view, the research process of various objects should be divided into seven consecutive stages, which are a cybernetic chain with feedback.

The paper also gives a formalized standard description based on function tables, an algorithmic method for solving the problems of managing complex systems based on algebra over function tables, and methods for synthesizing workplace complexes using production systems as an example.

¹*corresponding author*

2020 *Mathematics Subject Classification.* 93C95, 68Q12, 68Q85.

Key words and phrases. optimization, mathematical models, function table, mathematical methods, network security, enterprise architecture, firewall, algorithm.

1. INTRODUCTION

The global trend in the development of control systems is associated with the use of modern mathematical methods and high-performance electronic computers to introduce intelligent algorithmic systems in various spheres of economic activity, primarily in the management of equipment and technological processes. In this area, large-scale studies are conducted in developed countries of the world.

In the works of scientists of the Institute of Control Problems of the Russian Academy of Sciences, methods for modeling aggregate control systems for complex objects are proposed. Scientists from the Cambridge University of Great Britain solved the problems of a unified approach to the formalization of studies of dynamical systems, which was called the «Algorithmic direction in the study of cybersystems». Specialists from the V.M.Glushkov Institute of Cybernetics of Ukraine have developed automatic models for solving problems that allow designing the process of controlling complex systems (CS) [1, 2].

At present, the algorithm of control processes using modern methods and means of information technology to improve management at all levels of simulation of control dynamic systems, create new effective methods for rational and adequate description of objects of research, optimize and construct a control algorithm using a single technology is not well understood.

Thus, a comprehensive study of objects is needed, starting with preliminary study, obtaining adequate models, algorithm processes and ending with the creation of effective control systems. Therefore, the algorithm of control processes using modern computer technology requires not only improved management at all levels, but also the creation of new effective methods for the rational and adequate description of objects of research, optimization and construction of a control algorithm [3, 4].

The aim of the study of this work is to develop a method and technology for the control algorithm of dynamic complex systems based on a unified and standard description of research objects and modeling the control process using production systems as an example.

2. METHODOLOGY OF THE ALGORITHMIC APPROACH FOR MANAGING COMPLEX SYSTEMS

For an indivisible element of the system we take the workplace, denoting it by r with indices, and the set of workplaces (WP) – by R . Each r is represented in the form of workers, an employee plus machines and machines. Each r has inputs x and outputs y , an internal state Z . Signals (information) or materials in the form of products, substances (liquid or gaseous), etc. are transmitted to the inputs. Part of the input signal effects can be control (g).

As machines, machine tools and chemical devices, computers or just a shovel are used. Machines act as tools, and information, materials as objects of labor.

The workplace r generally corresponds to the aggregate of N.P.Buslenko [5].

Each r is assigned a certain number of operations P . We denote the set of operations by P . In addition, they function in time and have spatial coordinates. The set r is interconnected by arcs and forms a communication network with flows. This refers to the flow of information, substances, as well as transport, human flows, etc.

So, the system is represented in the form of a communication network, the vertices of which depict jobs capable of performing a certain number of operations (solving problems, processing materials, etc.), and the arcs correspond to flows between these places. We will call such a network R , a network that differs from the aggregate network of N.P.Buslenko [5] in that it allows the motion in time of various flows in the network with possible distortions and does not introduce additional vertices instead of arcs. During the functioning of the system, the network structure may change over time - old arcs and vertices are canceled, and new ones are added. We will call such networks situational or – RC - a network [5, 6].

When solving a certain class of problems over time (t_1, t_2) , one of the operations assigned to it is performed on each WP . Therefore, the construction of the network itself and the definition of the operation attributed to it is the main task of system research. In a certain period of time, the network can be depicted as a directed graph of constant structure (Figure. 1):

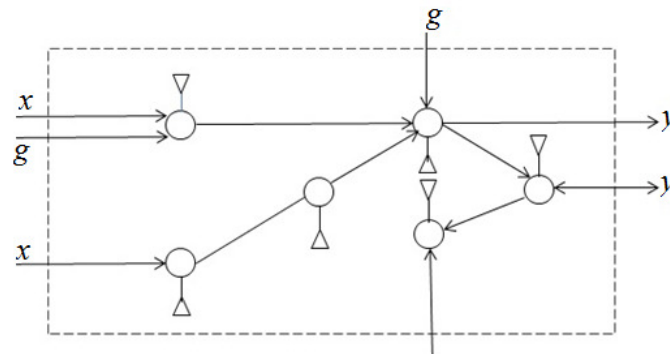


FIGURE 1. Oriented graph of constant structure

On (Figure. 1), circles indicate operations, triangles workplaces where these operations are performed in a given period of time.

A different view of this network is shown in (Figure. 2):

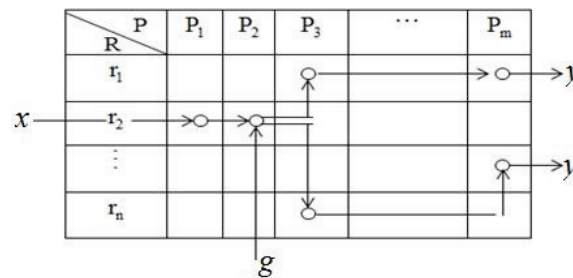


FIGURE 2. Function table

We call such a representation a functioning table (*FT*) and *R*-, *RC* - we will represent the network in the form of a *FT*. On this network, you can fix the flow parameters and the network operation mode in time. The time intervals $(t_0, t_1), (t_1, t_2), \dots, (t_{n-1}, t_n)$, during which the structure *FT* remains unchanged, we will call technological cycles. In addition, with the help of *FT* it is possible to link the functions of various systems, which is crucial for such large *CS* as automated systems. As for mathematical modeling, an extreme (but sometimes necessary) case is possible when models of various operations are written out. This way does not lead to generalizations, does not exclude duplication, and therefore is not acceptable. Therefore, when constructing algorithmic systems, the method of generating models from the general laws of the functioning of systems is used,

and particular models of the functioning of systems, i.e. specific operations are written out in extreme cases [7–11].

In organizational systems, the general value laws can be taken to be the natural-value balances of products, balances of capacities, labor resources, incomes and expenses of the population, etc. When studying production processes, one can take aggregate models of N.P. Buslenko [5], situational models, as well as a queuing model and automatic models. Technological processes are mainly described by the laws of continuum mechanics and molecular physics. For scientific studies in the field of continuum mechanics, the possibility of generating particular models from general conservation laws has been proved [12–14].

As for research in the field of mathematics, a powerful apparatus of mathematical logic for proving theorems has been created here, nevertheless, the possibility of a set of sequences of computer operations to prove new theorems remains controversial and requires additional research [15, 16].

In the algorithmic scheme of formalizing system research, software is proposed to be built using six main ($B_1 - B_6$) and two auxiliary banks (B_n - setting, B_0 - operational). Each main bank has information and operational parts and a set of modules; determining the composition of banks is the subject of further research. Setting bank - B_n provides a dialogue with the user, its structure depends on the customer's requests. The contents of the operating bank - B_0 can be arranged in the form of a block diagram of programs taking into account the composition of the modules of the main banks.

This is the general way to formalize descriptions of similar *CS*, such as automated systems. The task is reduced to the analysis according to this scheme of all the work done in the field of automated systems in the republic and the construction, improvement of *FT*, the scheme for generating models and algorithmic banks of this system.

Algorithmization methods are of particular importance in the automation of all stages of system research and the construction of control algorithms. From this point of view, the process of researching various objects can be divided into seven successive stages (Figure. 3), which are a cybernetic chain with feedback.

In algorithms [1–4], experience is understood in a broad, philosophical sense. This includes the experience of mankind accumulated over centuries, recorded in monographs, articles and other publications, as well as laboratory and field experiments. At the "experience" stage, the creation of information retrieval systems

(information banks) and the wide automation of experiments with the development and implementation of means for collecting, transmitting and processing experimental data are supposed.

The experience accumulated by mankind allows us to formulate the laws of functioning of various systems. For example, in society, these are the laws of economics, in nature, the laws of evolution, in mechanics of continuous media, the laws of conservation, etc. In algorithms, known laws are encrypted and entered into the computer's memory. New laws are formulated based on the results of an automated experiment [17, 18].

In various branches of science, on the basis of universal laws, various problems belonging to different classes are solved. Classification of these tasks and automatic recognition of classes are performed at the "task" stage. In scientific and engineering practice, many specific local problems are daily solved. If, based on experience, the main parameters of the problem are determined, i.e. it is set, then according to well-known laws on a computer, you can derive mathematical dependencies (logical and analytical) that describe this problem. Such dependencies, designed in algebraic, differential, integral and other types of equations, are mathematical models of the process (problem) under study. The derivation of mathematical models is automated at the stage of "mathematical models". Thus, many models $M = \{M_1, M_2, \dots, M_n\}$ - are formed.

In computational mathematics, for the approximate solution of classes of equations, many algorithms $A = \{A_1, A_2, \dots, A_m\}$ have been developed. To solve computer problems, it is necessary to first construct a mapping of the set M on A , i.e. first establish the applicability of algorithms from A to operators from a variety of mathematical models M . Applicability is related to the proof of convergence and stability of the algorithm. So, any one model from M corresponds to a subset of the algorithms from A . To select a single algorithm, it is necessary to set the optimality conditions of the algorithm and check them. This circle of problems is solved at the "algorithms" stage.

After choosing an algorithm, it is necessary to proceed to an approximate system. For example, using the grid method, select a grid region (grid diagram) to build and solve grid equations. All these tasks are solved with the help of logical and arithmetic operations at the stage of "software". At this stage, a programming language is also selected and all necessary programs are compiled [19].

At the last stage of the “calculation”, computer complexes are created that are associated with experimental installations and the account is linked to experience.

In practical implementation, the stages of algorithmization are arranged in the form of six main and two auxiliary algorithmic banks (AB). The main banks are called data banks (№1), laws (№2), features (№3), models (№4), algorithms (№5) and application programs (№6). Ancillary bank and operating bank are included in the subsidiary.

Each bank consists of two parts: information and operating. The information part stores numerical or symbolic data in fixed languages, which are entered in advance (permanent information). The operational part of the algorithmic bank contains software packages that process the information arrays of this bank.

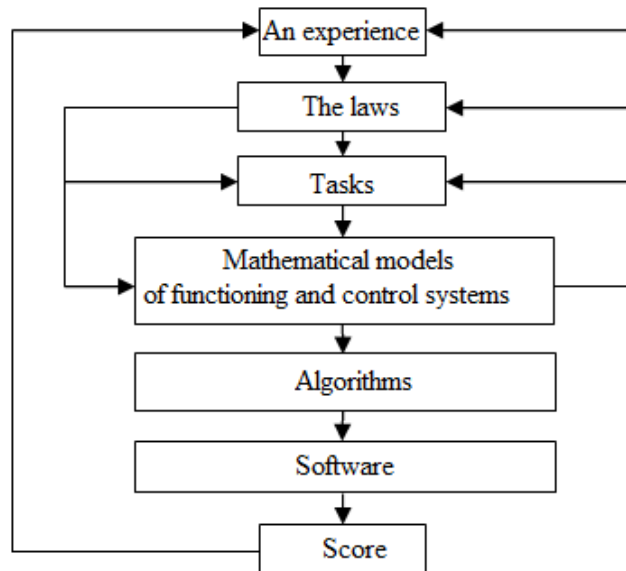


FIGURE 3. Seven consecutive stages of the research process various objects

The banks work order is shown in (Fig. 4). The task is formulated in the statement bank, and then it is entered into the operational bank and sign bank. Digital source data is sent to the databank. In a bank of signs, conditions are translated into a symbolic form, i.e. correspond in a special language, then the signs of the task are developed, which are transmitted to the bank of laws. Using these features, a mathematical model of the problem is generated in the bank of laws, which is transferred to the bank of models. Here the model is analyzed, the type

of operator and its features are determined. The operator (i.e., the mathematical model of the problem) with its attributes is sent to the bank of algorithms, where the optimal algorithm is selected, the resolving equation is constructed. In addition, in the bank of algorithms, a sign is developed for selecting an account program and signals for sampling the numerical values of the coefficients. The resolving equation and the computational program are sent from the application bank to the operational bank, and the signs for the selection of numbers are sent to the data bank. In the operating bank, the resolving equation is finally formed and its numerical solution is implemented. At the same time, control of the account and the issuance of results in a predetermined form is carried out. In the described system, a regular account is made using only one AB (№6), and logical operations are performed in all other banks. The data bank is the accumulator of all the digital characteristics of the object (constants and variables) and is filled before the entire system works. This bank is associated with an experimental system.

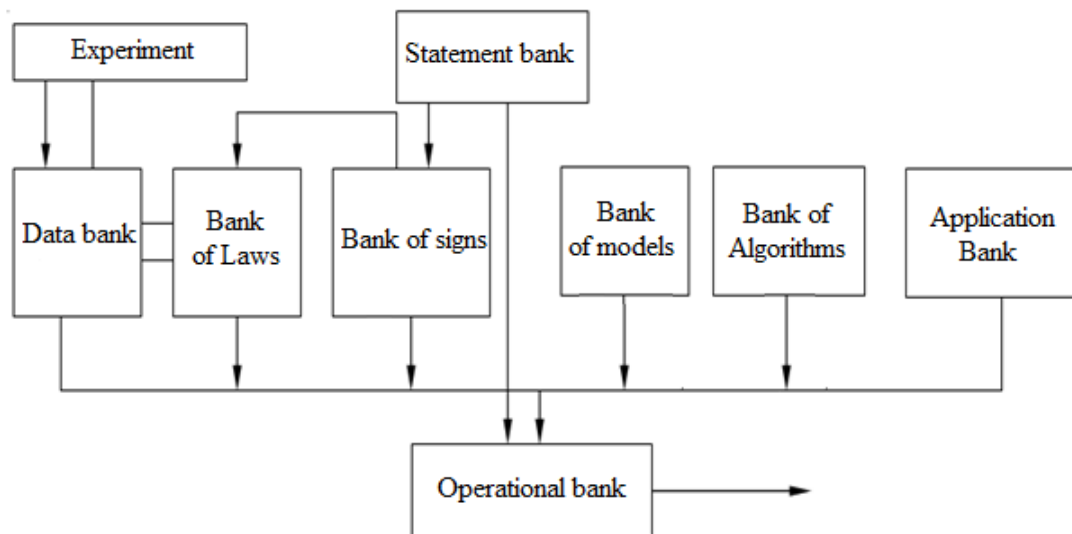


FIGURE 4. The scheme of work of algorithmic banks.

These are the stages of the algorithmization. The application of algorithmic methods to establishing the laws of the functioning of large systems, determining a control strategy, and finally, to building control systems (governing bodies) is of great theoretical and practical importance.

The development of these problems will lead to the creation of a direction when not only management processes will be automated, but also the process of designing and constructing control systems.

3. ALGORITHMIC CONTROL METHOD BASED ON ALGEBRA OVER FUNCTION TABLES

Let the dynamic functioning tables be defined as follows: $TA = \{P, D, I, O, A, T, \Delta, F\}$, where P, D, I, O, A, T, Δ , respectively, the set of positions, operations, input and output states of the WP , time intervals and coordinates of the WP system; $F(t)$ is the function of changing the functioning table in time. If $\forall t_i \in T$ there is a function $F(t_i) = const$, then such a functioning table is called static (stationary). The function $F(t)$, which defines the changes in the functioning table, is called the control function of the aggregate system or the function of planning processes in the system.

We introduce an algebra over FT and for formal operations on FT , the corresponding rules in matrix form [6–9] are determined.

At each time interval t_i is represented as a marked network [1–4]: $M = \{P, D, I, O, \mu\}$, where is μ the function $N : \mu : P \rightarrow N$. Each marking μ can be represented as a vector $\mu = (\mu_1, \mu_2, \dots, \mu_n)$, $n = |P|$, $\forall \mu_i \in N$, $i = \overline{1, n}$.

We introduce an algebra over FT and for formal operations on FT , the corresponding rules in matrix form are determined.

The time intervals during which the Petri net does not change will be called technological cycles (TC).

TC are a kind of Petri nets, therefore, by analogy with Petri nets, the main language tools for describing processes are determined.

He says that the language - L is the language of the TC L - of the type if there is $\delta : T \rightarrow \Sigma$, a transition room, and the initial marking and the finite set of final markings F are such that $L = \{\delta(B) \in \Sigma^* \setminus \tilde{B} \in T\}$ and $\delta(\mu, \tilde{B}) \in F$, where is $\delta(\mu, \tilde{B})$ the transition function, i.e. for and marking function is the result of sequential start $(t_{i_1}, t_{i_2}, \dots, t_{i_k})$.

Many complex systems are a composition of subsystems. Each of the subsystems can be represented by the corresponding TC with its own language. With a sequential combination of subsystems, the corresponding TC are the concatenation of the TC from one, two three, etc. languages of the TC . The concatenation of

languages is formally defined as:

$$L_1 \times L_2 \times \dots \times L_n = \{x_1, x_2, \dots, x_n : x_1 \in L_1, x_2 \in L_2, \dots, x_n \in L_n\}.$$

The next operation of the TC composition is the union operation. Formally, it is defined as follows:

$$L_1 \bigcup L_2 \bigcup \dots \bigcup L_n = \bigcup_{i=1}^n L_i = \{x : x \in L_1 \text{ or } x \in L_2 \dots \text{ or } x \in L_n\}.$$

Further, the operation of the parallel composition of the TC is an operation that is defined for the TC as follows:

$$\alpha_1 x_1 \parallel \alpha_2 x_2 \parallel \dots \parallel \alpha_n x_n = \alpha_1 (x_1 \parallel \alpha_2 x_2 \parallel \dots \parallel \alpha_n x_n) + \alpha_2 (\alpha_1 x_1 \parallel x_2 \parallel \dots \parallel \alpha_n x_n) + \dots + \alpha_n (\alpha_1 x_1 \parallel \alpha_2 x_2 \parallel \dots \parallel x_n)$$

and

$$\alpha \parallel \lambda = \lambda \parallel \alpha = \alpha.$$

A parallel composition of two or more languages is:

$$L_1 \parallel L_2 \parallel \dots \parallel L_n = \{x_1 \parallel x_2 \parallel \dots \parallel x_n : x_1 \in L_1, x_2 \in L_2, \dots, x_n \in L_n\}.$$

The intersection operation, as in the case of union, is similar to the set-theoretic definition of intersection and is defined for TC languages as follows:

$$L_1 \bigcap L_2 = \{x : x \in L_1 \text{ and } x \in L_2\}.$$

The inverse operation of sentence x is a sentence whose symbols are in the opposite order. We define this operation recursively:

$$\alpha^R = a, (ax)^R = x^R a \text{ for } a, x \in \Sigma.$$

Each of the subsystems can be represented by the corresponding *TC* with its own language. It is easy to see that the languages of the *TC* are closed with respect to any finite number of union, intersection, circulation, parallel composition and concatenation operations carried out in any order. A system consisting of a basic set $FT = \{TC_i\}$ of technological cycles and a set of operations called a signature is a universal algebra if each of the operations belonging to the signature Ω is everywhere defined on the set $FT = \{TC_i\}$.

Thus, the algebra we introduced above the functioning tables allows us to use algebraic methods for constructing control algorithms in the future, where in a

TC each cycle can be described by two matrices C^- and C^+ , and $C^- (C^+)$ is a system of input (output) vectors states.

The control system is usually a three-level system that implements the function of design (planning) and management of the production process.

The upper level - organizational and operational management - forms production tasks for specific local software control systems. At this level, all the necessary stationary FT of the production system are built. Stationary FT are detailed and converted into dynamic FT of control monitors for each WP .

The second level - local control on the basis of stationary FT - loads into the control monitors corresponding to the TC , and then initiates the WP start-up process in accordance with the reference stationary FT and exercises control over the execution of operations on the given FT control monitor.

The lower level is a control monitors in the form of machines with store memory. At this level, sequential execution of operations is performed on a given control monitor of a dynamic FT .

Consider the control model production systems (PS). A relationship of the form $R : D$, where R is the set of jobs, and D is the set of lots of parts obtained as a result of constructing a dynamic table of functioning, is an integral part of a wider ratio $R; D; O$ to the transitivity of all relations, we can construct the following FT transformation:

$$R : D \rightarrow D : O \rightarrow O : U, \quad (U = \{\text{CONTROL}\}).$$

Hence $R : U$. Such a sequential transformation is carried out at the top level of management, when planning the production process, we get FT with a ratio of the form $R : D$, and stationary FT with ratios of the type $D : O$ and $O : U$ are obtained as a result of the work of production preparation algorithms. In constructing dynamic FT with the $R : D$ ratio, we will use the scheme shown in Fig. 5. FT analysis for the feasibility of a solution is based on the resolution of two equations in matrix form:

$$\bar{Y}_1 = Y_0 + A \cdot \bar{X}, \quad A\bar{W} = 0.$$

Based on the analysis of the first equation, the existence of a non-trivial fundamental solution system for FT is determined, defined in the form of matrix A . Matrix A is obtained from the matrices A^+ , and A^- describing FT , and is connected with them by the relation $A = A^+ - A^-$. Finding a system of vectors W

means preserving the FT during labeling by one of the vectors. Here, as the vector Y , the initial state of the FT is used, X is the vector of the operations performed, A is the FT in the matrix form, and $\bar{Y}_1$ - the final state of the FT .

The solution of the equations is based on the methods of linear algebra described in [6–8]. When determining the restrictions on the throughput of the FT , we will normalize, calculate the upper and lower boundaries of the positions that determine the material flow in the FT : $t_k \setminus \alpha_{ij} \leq \chi_{ij} \leq t_k$ where is χ_{ij} - the normalized value of the position in the t_k -th time interval; α_{ij} - is the value of the i -th position for the j -th operation in the FT . Conversion to the network model is performed as follows:

$$\sum_{k=1}^m \sum_{j=1}^n [(c_{2j}^k - c_{1j}^k) x_{2j}^k + c_{1j}^k b_j^k] \rightarrow \min,$$

where are c_{2j}^k - the cost of demand of the k -th product for the j -th node, c_{1j}^k - the cost of the offer of the k -th product of the j -th node, x_{ij}^k - the stream of the k -th product from the i -th node to the j -th node, b_j^k - demand of the j -th node for the k -th product, provided that, $\sum_{k=1}^m \chi_{2j}^k + S_{1j} = U_{2j}, j = \overline{1, n}, - \sum_{j=1}^n \chi_{1j}^k = a_2^k, k = \overline{1, n}, - \sum_{j=1}^n S_{2j} = - \sum_{j=1}^n U_{2j} + \sum_{k=1}^m a_2^k, a_2^k$ - where are a_2^k - the offers of the j -th node for the k -th product, $\chi_{2j}^k + \chi_{1j}^k = b_j^k, S_{2j} + S_{1j} = U_{2j} + U_{1j} - \sum_{k=1}^m b_j^k, \chi_{ij}^k, S_{ij} \geq 0, i = \overline{1, 2}, j = \overline{1, n}, k = \overline{1, \infty}$ or $\chi_{2j} \leq \chi_{ij}^k$.

Thus, FT with $R : D$ and $R : O$ ratios give us the basic initial technological information necessary for managing software. At the local control level, dynamic FT with an $R : U$ ratio are loaded into the control monitors in the order prescribed by a stationary FT with $R : U$. Monitoring of processing operations is carried out on a stationary FT with $R : O$.

The supply of semi-finished products on the line from the warehouse, transportation and warehousing of finished products are synchronized and controlled by local transport and warehouse management systems based on two stationary FT with ratios $R : D$ and $R : O$. At the lower level are the control monitors, each of which works according to its dynamic $FT C : U$.

If the signal of the end of the operation does not enter the local control system, then in the FT a mark about the fault of the PM is made and the FT is analyzed for the possibility of further functioning of the substation. To do this, re-mark the

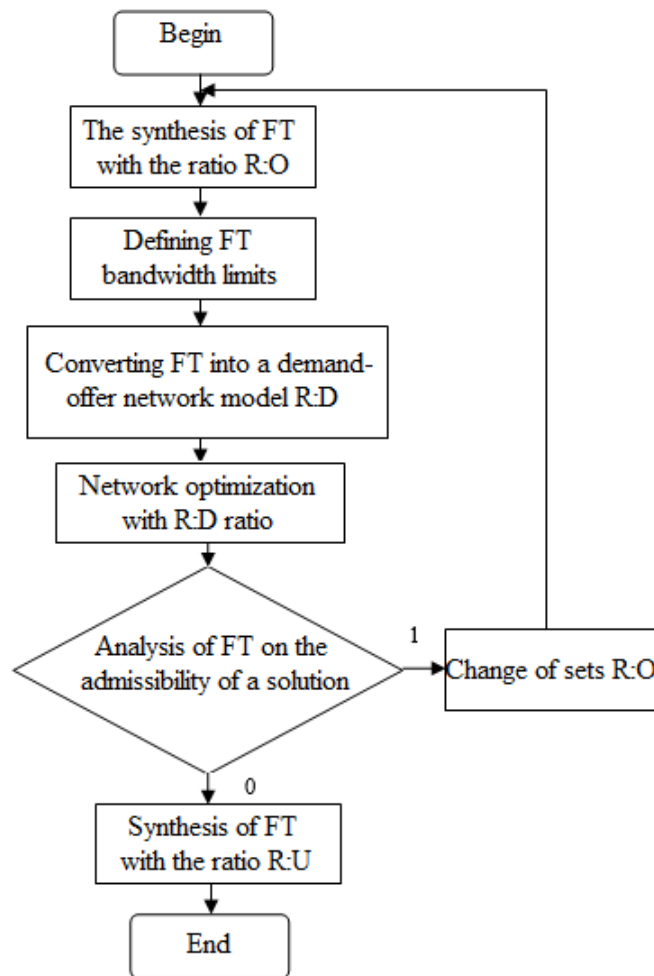


FIGURE 5. The scheme for constructing dynamic FT with relations of the form $R : O, R : U$.

FT , and solve the equation of reachability of the final state of the network. To determine the malfunction of a particular WP in the FT there are control points (special control operations). To determine the state of FM at the current time t_m , the type of structure between the control points is determined.

Three types of structures are considered: sequential, parallel and series - parallel.

For each type, there is a time delay for not receiving an operation completion signal.

For the serial structure of the compound WP $t_m = (n - i) \cdot t_r$, where n - is the number of series connected WP ; t_r - is the rhythm of the system; i - is the operation number of the faulty WP .

This implies $t = (n - t_m) \bmod(t_r)$. For a parallel structure, the delay time is $t_m = i \cdot \Delta t$ where, $\Delta t = t_r/m$, m - is the number of parallel connected WP .

We have from here $i = [t_m/\Delta t]$. For a series-parallel structure $t_m = (n - i)t_r + \Delta t \cdot i$. Then $i = \frac{t_m - n \cdot t_r}{\Delta t - t_r}$.

After determining the fault number of the WP and subsequent analysis of reachability in the event that the final state between the control points is not reachable, a decision is made to block the WP circuit and replace the faulty WP with a backup one.

Consider the process of constructing stationary FT with given relations $R : O$ and $R : U$. The process of constructing a FT with a given ratio is based on algorithms for the synthesis of complexes from individual elements. In our case, there are three sets: $R = \{r_i\}$, $D = \{d_e\}$ and $O = \{o_j\}$ where are r_i, d_e, o_j - the jobs, parts and operations, respectively. At each workplace, you can perform a specific group of operations and, therefore $d_e : \{O_k^l\}$. In turn, for the manufacture of parts requires a certain sequence of operations and, therefore $d_e : \{O_k^l\}$.

Thus, the given relations $R : O$ and $D : O$ in the general case do not coincide; moreover, the ratio $R : O$ is not ordered. In this regard, the establishment of an ordered $R : O$ ratio is closely related to the task of establishing the $D : R$ ratio. The synthesis algorithm allows to establish this ratio.

Consider three main cases that describe all the synthesis processes:

1. Let condition $\{O_k^l\} \cap \{O_m^i\} = \{O_k^l\}$ be satisfied for some subset $D_i = \{d_e\}$ and the corresponding $R_k = \{r_i\}$. This means that any detail can be completely processed at the workplace, and this trivial case does not need to be considered; we only note that in this case many details are processed at one WP .
2. Let D_i and R_k not intersect on the set of operations O , that is $\{O_k^l\} \cap \{O_m^i\} = \emptyset$. Then the process can be stopped, and in this case the subset is that partition on the set R by the operation O_k^l .
3. Let D_i and R_k partially intersect in the operation:

$$\{O_{1n}^l\} = \{O_{ok}^l\} \cap \{O_{om}^i\}, \text{ where } O_{1n}^l, O_{ok}^l, O_{om}^i \in O,$$

If the intersection is not empty, then we get a subset $\{O'_{2k}\} = \{O'_{1k}\} \setminus \{O'_{1m}\}$ and proceed to step 1. If the intersection is empty, then the algorithm stops. At the same time, we obtain the following sequence of subsets of operations:

Constructed according to the above *FT* algorithm with the ratio $D : R : O$ for processing operations, it contains, in turn, the initial data for the synthesis of *FT* with the same ratio for transportation operations. The synthesis algorithm in this case is simplified, since for each vehicle there is only one operation - transportation operation O .

4. APPLICATION OF FUNCTIONING TABLES IN THE PRODUCTION OF ELECTRONIC EQUIPMENT (EE)

The application of the algorithmic system is the construction of a discrete-type PS control algorithm based on FT in the process of dynamic changes in production situations, maintaining a real-time simulation model for analyzing the state of a technological process (TP). To simulate the TP for the manufacture of REA by metallization of through holes between the layers, a FT is proposed and the

sequence of TP operations in sections and WP is indicated. In the FT , operations are denoted by O and by A by sections.

Performing an operation in a certain area corresponds to a transition to the network. Transition - performing the operation during the period T . The three entry points of transitions in the network mean the following: the top position is the presence of UE and additional devices, materials; middle position - the presence of a semi-finished product; the bottom position is the condition of the equipment. Resources, procurements and information in the network correspond to position markers.

The FT network reflects the flow of materials and information in accordance with the TP of the EE production.

Application of an algorithmic system. We will present the table of the functioning of the TP of the metallization of printed-circuit boards production in the form of matrices D^+ , D^- (D^+ (D^-))-the matrix of the output (input) positions of the FT transitions) as follows: Matrix $D = \|a_{ij}\|$, (43×120) , where, $a_{ij} = 1$, if

$$a_{ij} = \begin{cases} K_1 = \overline{1, 3} \\ K_5 = \overline{12, 14} \\ K_9 = \overline{23, 25} \\ K_{13} = \overline{34, 36} \\ K_{17} = \overline{44, 46} \\ K_{21} = 57 \\ K_{25} = 68 \\ K_{29} = \overline{77, 79} \\ K_{33} = \overline{89, 91} \\ K_{37} = \overline{101, 103} \\ K_{41} = \overline{112, 114} \end{cases} \begin{cases} i = \overline{1, 43} \\ K_2 = \overline{4, 6} \\ K_6 = \overline{15, 17} \\ K_{10} = \overline{26, 28} \\ K_{14} = 38 \\ K_{18} = \overline{47, 49} \\ K_{22} = \overline{58, 60} \\ K_{26} = \overline{69, 71} \\ K_{30} = \overline{80, 82} \\ K_{34} = \overline{92, 94} \\ K_{38} = 105 \\ K_{42} = 115, 117 \end{cases} \begin{cases} j = K_i \\ K_3 = \overline{7, 9} \\ K_7 = 19 \\ K_{11} = 30 \\ K_{15} = \overline{39, 41} \\ K_{19} = \overline{50, 52} \\ K_{23} = \overline{61, 63} \\ K_{27} = \overline{72, 74} \\ K_{31} = \overline{83, 85} \\ K_{35} = \overline{90, 97} \\ K_{39} = \overline{106, 108} \\ K_{43} = \overline{118, 120} \end{cases} \begin{cases} K_4 = 11 \\ K_8 = \overline{20, 22} \\ K_{12} = \overline{31, 33} \\ K_{16} = 43 \\ K_{20} = \overline{53, 55} \\ K_{24} = \overline{64, 66} \\ K_{28} = 76 \\ K_{32} = \overline{86, 88} \\ K_{36} = \overline{98, 1} \\ K_{40} = \overline{109, 111} \end{cases}$$

If

$$i, j = 4, 10 \wedge 7, 18 \wedge 11, 29 \wedge 14, 37 \wedge 16, 42 \wedge \\ \wedge 25, 67 \wedge 28, 75 \wedge 38, 104 \wedge 43, 119, 0$$

otherwise.

Matrix $D^- = \|b_{ij}\|$, (43×120) , where $b_{ij} = 1$, if $i = 1, 43$, $j = K_{ij}$ and $K_1 = 1, 3, 5, .$

$$\begin{cases} K_2 = 4, 6, 8 \\ K_3 = 7, 9 \\ K_4 = 3, 13, 33 \\ K_5 = 12, 14, 16 \\ K_6 = 15, 17, 21 \\ K_7 = 13 \\ K_8 = 20, 22, 24 \\ K_9 = 23, 25, 27 \\ K_{10} = 20, 26, 28, 32 \\ K_{11} = 22 \\ K_{12} = 31, 33, 35 \end{cases} \quad \begin{cases} K_{13} = 34, 36, 40 \\ K_{14} = 28, 30, 34 \\ K_{15} = 44, 46, 48 \\ K_{16} = 47, 49, 51 \\ K_{17} = 47, 49, 51 \\ K_{18} = 50, 52, 54 \\ K_{19} = 53, 55, 49 \\ K_{20} = 41 \\ K_{21} = 58, 60, 62 \\ K_{22} = 61, 63, 65 \\ K_{23} = 64, 66, 70 \end{cases}$$

$$\begin{cases} K_{24} = 60 \\ K_{25} = 69, 71, 73 \\ K_{26} = 72, 74, 78 \\ K_{27} = 72 \\ K_{28} = 77, 79, 81 \\ K_{29} = 80, 82, 84 \\ K_{30} = 83, 85, 87 \\ K_{31} = 86, 88, 90 \\ K_{32} = 89, 91, 93 \\ K_{33} = 92, 94, 96 \\ K_{35} = 35, 97, 99 \end{cases} \quad \begin{cases} K_{36} = 102 \\ K_{37} = 101, 103, 107 \\ K_{38} = 97 \\ K_{39} = 106, 108, 110 \\ K_{40} = 109, 111, 113 \\ K_{41} = 112, 114, 116 \\ K_{42} = 115, 117, 118 \\ K_{43} = 63, 110 \end{cases}$$

$b_{ij} = 0$ otherwise [6–8].

Matrices and have dimensions 43×120 , where 43 is the number of transitions, 120 is the number of positions. They are close to diagonal view and very sparse. For the analysis of FT , the equation is solved:

$$D \overline{W} = 0,$$

where is a system of variable vectors whose values correspond to the fundamental system of solutions of equation 4.

Since in $\overline{W}$ at least one component is nonzero, and it preserves marking of markers, to verify the correct functioning of the constructed model, we solve the

TABLE 1. Description of the table

Operation	Upper flow limit	Lower float limit	Number of WP	Normal upper flow limit	Normal lower flow limit	Cost unit flow
Blank	10	5	2	5	3	5
Drill	45	5	12	3	1	10
Electro plating	140	140	2	70	70	3
Drawing	20	10	2	10	5	5
Process controll	10	5	20	0,5	0,25	30
Stamping	65	65	2	32	32	3
Pressing	720	600	2	360	300	15
Milling	30	80	3	10	6	10
Marking	5	3	5	1	0,3	5
Cleaning	10	5	2	5	2	5

areas of economic activity, primarily in the management of equipment and technological processes.

The paper proposes a unified approach to the formalization of studies of dynamic systems, which is called the "algorithmic direction" in the study of cyber systems, which allows you to flexibly and without additional costs for the development of new software tools to manage the object, taking into account the impact of external factors on the production system, in particular on the course of the production process.

The article developed a formalized standard description based on function tables, an algorithmic method for solving the problems of managing complex systems based on algebra over function tables, and methods for synthesizing workplace complexes using production systems as an example.

REFERENCES

- [1] V. M. GLUSHKOV: *Introduction to cybernetics*, Kiev.: AN UzSSR, (1964), 324.

- [2] V. K. KABULOV: *Algorithmization in the mechanics of continuous media*, Tashkent: "Fan" of the UzSSR, 1979.
- [3] V. K. KABULOV: *Questions of the standard description of systems and aggregate systems N.P. Buslenko*, Questions of Cybernetics, Tashkent, **130** (1985), 3–20.
- [4] V. K. KABULOV: *Formalization issues in systems research*, Questions of Cybernetics, Tashkent, **126** (1984), 3–15.
- [5] N. P. BUSLENKO: *Modeling complex systems*, M.: Nauka, Tashkent, **126** (1978), 390.
- [6] J. PETERSON: *Theory of Petri nets and systems modeling*, M.: World, 1984.
- [7] A. V. KABULOV, I. H. NORMATOV, A. O. ASHUROV: *Computational methods of minimization of multiple functions*, Journal of Physics, (2019), 1260.
- [8] A. V. KABULOV, I. H. NORMATOV: *About problems of decoding and searching for the maximum upper zero of discrete monotone functions*, Journal of Physics, (2019), 1210.
- [9] I. H. NORMATOV: *Principle of independence of continuation of functions multivalued logic from coding*, Journal of Physics, (2018), 1210.
- [10] N. T. FISUN, A. S. STUKALO: *Stukalo Application package for the automated construction of network models*, USIM, **2** 1987.
- [11] L. FORD, D. FULKERSON: *Streams in networks*, M.: Mir, 1986.
- [12] X. AHUJAH: *Network management and design methods in production*, M.: Mir, (1979), 638.
- [13] A. V. KABULOV, E. URUNBAEV, I. H. NORMATOV, A. O. ASHUROV: *Synthesis methods of optimal discrete corrective functions*, Advances in Mathematics: Scientific Journal, **9**(9) (2020), 6467–6482.
- [14] A. V. KABULOV, I. H. NORMATOV, SH. BOLTAEV, I. SAYMANOV: *Logic method of classification of objects with non-joining classes*, Advances in Mathematics: Scientific Journal, **9**(10) (2020), 1857–8365.
- [15] I. H. NORMATOV, E. KAMOLOV: *Development of an algorithm for optimizing the technological process of kaolin enrichment*, IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS), Vancouver, BC, Canada, (2020), 1–4.
- [16] A. V. KABULOV, I. H. NORMATOV, A. SEYTOV, A. KUDAYBERGENOV: *Optimal Management of Water Resources in Large Main Canals with Cascade Pumping Stations*, IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS), Vancouver, BC, Canada, (2020), 1–4.
- [17] A. V. KABULOV, I. H. NORMATOV, A. KARIMOV: *Algorithmization control of complex systems based on functioning tables*, Journal of Physics Conference Series, **6** (2020), 1–9.
- [18] A. V. KABULOV, I. H. NORMATOV: *About problems of decoding and searching for the maximum upper zero of discrete monotone functions*, Journal of Physics Conference Series, **6** (2019), 1260.
- [19] A. V. KABULOV, I. H. NORMATOV, I. I. KALANDAROV, A. A. KARIMOV: *Algorithmic Method of organization of Specialized Workshops*, Inter/Jour.IJARSET, **5**(4) (2018), 5670 – 5675.

DEPARTMENT OF INFORMATION SECURITY
NATIONAL UNIVERSITY OF UZBEKISTAN NAMED AFTER MIRZO ULUGBEK
UNIVERSITY STREET, 4, 100174 TASHKENT, UZBEKISTAN
Email address: anvarkabulov@mail.ru

DEPARTMENT OF INFORMATION SECURITY
NATIONAL UNIVERSITY OF UZBEKISTAN NAMED AFTER MIRZO ULUGBEK
UNIVERSITY STREET, 4, 100174 TASHKENT, UZBEKISTAN
Email address: ibragim_normatov@mail.ru

DEPARTMENT OF INFORMATION SECURITY
NATIONAL UNIVERSITY OF UZBEKISTAN NAMED AFTER MIRZO ULUGBEK
UNIVERSITY STREET, 4, 100174 TASHKENT, UZBEKISTAN
Email address: karimov.anvar11@mail.ru

DEPARTMENT OF INFORMATION SECURITY
NATIONAL UNIVERSITY OF UZBEKISTAN NAMED AFTER MIRZO ULUGBEK
UNIVERSITY STREET, 4, 100174 TASHKENT, UZBEKISTAN
Email address: erkinbek0989@gmail.com