Algorithmization control of complex systems based on functioning tables

A V Kabulov[1], I H Normatov[2], A Karimov[1]

[1]National University of Uzbekistan, University street, 4, 100174 Tashkent, Uzbekistan
[2]Tashkent University of Information Technologies named after Muhammad al-Khwarizmi, Amir Temur street, 108, 100200 Tashkent, Uzbekistan

E-mail: ibragim_normatov@mail.ru

Abstract. This paper is devoted to algorithmic models and methods of managing complex systems based on algebra over the functioning tables. To build a control system, a description of processes, both strictly sequential and parallel, is required. For this, a unified description of processes with specified operations is introduced in the form of functioning tables. Then, algorithmic support for the equipment and organizational structure under consideration is built on the basis of algebra over the tables of the functioning of a complex system.

1. Introduction

According to academician V. M. Glushkov [1], the complexity of objectively necessary management tasks is growing faster than the number of people employed in production. Experimental data show that in the modern era the complexity of the necessary tasks of production management is growing faster than the square of the number of people employed in this production.

The way out of this situation is to increase labor productivity in the management of complex systems through the use of computer technology and algorithmic methods.

The process of control algorithmization of complex systems consists in constructing a control algorithm in a given time interval, which allows for any control system of this class to find a functional characteristic according to its scheme, information and coordinates. In this sense, algorithmization is a universal method for modeling control systems [2].

Already known algorithmic methods are suitable for analyzing the individual components of complex systems: automaton models, operator circuits, linear programming, etc. [1–3]. A feature of these methods is a complete description of the system components in a predetermined space of parameters and time, the complexity of the calculation in the decision-making process and the rigidity of the algorithm for determining the optimal control function. In this regard, there is a need to develop new algorithmic methods capable of solving complex "system-wide" problems for the entire complex system based on the analysis of individual components, which are quite fully confirmed by a full-scale experiment.

The most suitable is the algorithmic method of dynamic control modeling, which plays a significant role in solving control automation problems and allows to reveal the patterns of control processes, determine the flow of control information and reasonably choose a control algorithm. Moreover, the algorithmization of constructing control models for complex systems should be based on the hypothesis that the processes of research and management of complex systems are subordinated to a single technology.

Thus, a comprehensive study of objects is needed, starting from preliminary study, obtaining adequate models, process algorithmization and ending with the creation of effective control systems. Therefore, the algorithmization 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.

Algorithmization methods are of particular importance in automating all stages of system research and construction of control algorithms. From this point of view, the process of researching various objects can be divided into seven successive stages (Fig. 1), which represent a cybernetic circuit with feedback.

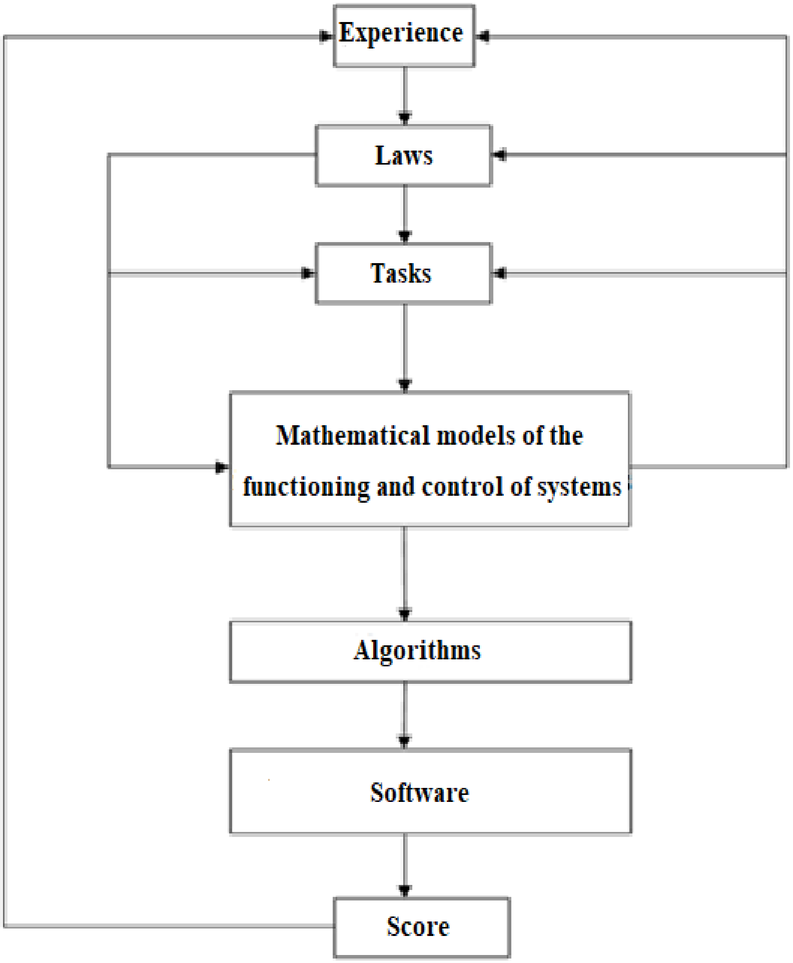
Experience
Laws
Tasks
Mathematical models of the functioning and control of systems
Algorithms
Software
Score

Figure 1. Seven consecutive stages of the research process of various objects

This article discusses the problem of control algorithms for complex systems based on their standard description in the form of jobs and job complexes, for the simulation of which dynamic functioning tables are proposed [4].

The analysis of the processes of solving problems when constructing an algorithm for managing complex systems is given. A description of the process of solving the object control problem based on an algorithmic approach is proposed.

2. Algorithmic method of managing complex systems based on the functioning tables.

Consider a single scheme for the representation of complex systems using graphs, model generation for task classes and software. We describe the general scheme of such an algorithmic approach. As an elementary indivisible element of the production system, we will take an element for which the following characteristics exist: coordinates, time intervals, operations and states. Let's call such an element a workplace (WP) and we will designate it as follows:

$$\alpha_i \in A, \ \ \alpha_i = \left\{\delta_{ij}, \mathrm{t}_j, d_{ij}, p_{ij}\right\},$$

where

$\delta_{ij} \in \Delta$ is the vector of coordinates in the time interval t_j, where Δ is the set of coordinates (geometric) WP;

$t_j \in T \ \ j$ – th time interval (T-set of time intervals);

$d_{ij} \in D$ – i-th operation performed in the t-th time interval (D is a set of operations)

$p_{ij} \in P$ – is the state vector of the *i*-th WP in the time interval (P is the set of WP states).

The following classes of operations can be distinguished as operations:
– technological (processing, assembly, transportation, warehousing);
– logical (AND, OR, excluding OR, NOT);
– planning (preliminary planning, scheduling, shift-daily task, operational dispatch control);
– regulatory reference (search of documents, control programs (CP), sorting of documents);
– exchange operations (CP, documents, directives, signs of condition);
– input-output operations (write, read).

As states of the WP can be taken as follows:
– the states of material flow (there is blank or there is not blank, there is rig or there is not rig, there is tool or there is not tool, there are materials or there are not materials);
– the state of information flow (operation stack CP, processing plan stack, directive stack, document stack, stack length № 1, stack length № 2, stack length № 3, stack length № 4);
– state of WP (corrected/not corrected, free/busy, ready/not ready, work/wait, processing time).

The set of work places connected by features is determined by some network at each time interval t_i (for example, the Petri net [5]). Changes in the WP network over time are determined by the function of changing the $F(x)$ network. Such a description of the system will be called a table of the functioning of the system. Graphically each operation d_j – performed on the WP α_i at the time t_k has coordinates (i, j, k).

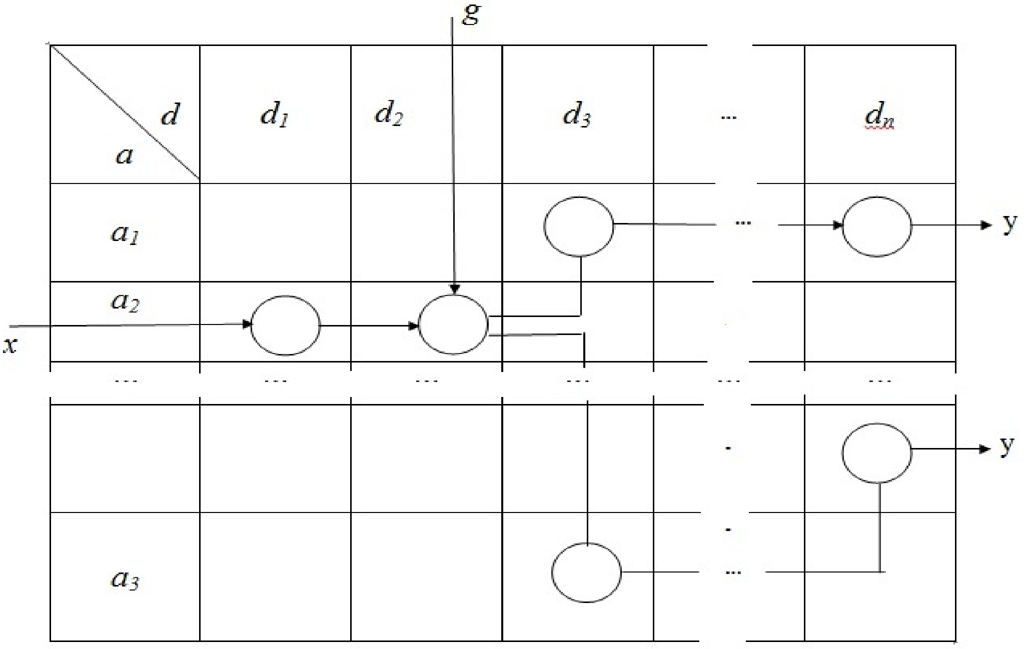

Figure 2. Function table hypothetical control object

Then the dynamic tables of functioning will be determined as follows:

$$T\Phi = \{P, D, I, O, A, T, \Delta, F\},$$

where P, D, I, O, A, T, Δ are the sets of positions (states), operations (transitions), input and output states of the WP, time intervals and coordinates of the WP system, respectively;

$F(x)$ is the function of changing the table of functioning in time t_i.

The control algorithm of complex systems in this case consists in finding an algorithm that allows for any complex control system to find a functional characteristic from its function table. In this sense, algorithmization is essentially a universal method for modeling the control process of complex systems based on function table. The task of synthesis of a control unit (monitor) and the process of synthesis of automata for the implementation of control operations are also formulated.

We introduce an algebra over function table and for formal operations on function table, the corresponding rules in matrix form are determined. In each time interval t_i, the description of the functioning tables is presented in the form of a labeled Petri net:

$$M = \{P, D, I, O, \mu\},$$

where μ function that displays the set of positions *P* in the set of *N* natural numbers $\mu : P \to N.$

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.

Says that the language – *L* is the language of the TC *L* – of the type if there is a transition room $\delta : T \to \sum$, and a initial marking μ and the finite set of final markings *F* are such that $L = \left\{\delta(B) \in \sum^*, \quad \tilde{B} \in T\right\}$ and $\delta\left(\mu, \tilde{B}\right) \in F$, where $\delta\left(\mu, \tilde{B}\right)$ is the transition function, i.e. for $\tilde{B}\left(t_{i_1}, t_{i_2}, ..., t_{i_k}\right)$ and marking function $\mu = \delta(\mu, \tilde{B})$ is the result of sequential start $\left(t_{i_1}, t_{i_2}, ..., t_{i_k}\right)$.

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:

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

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

$$L_1 \cup L_2 \cup, ..., \cup L_n = \bigcup_{i=1}^{n} L_i = \left\{x : x \in L_1 \; or \; x \in L_2 ... \text{or} \; x \in L_n\right\}.$$

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

$$\alpha_1 x_1 \| \alpha_2 x_2 \| ... \| \alpha_n x_n = \alpha\left(x_1\right) \| \alpha_2 x_2 ... \| ... \| \alpha_n x_n + \alpha_2\left(\alpha_1 x_1 \| x_2 \| ... \| \alpha_n x_n\right) + ... +$$
$$+ \alpha_n\left(\alpha_1 x_1 \| \alpha_2 x_2 \| ... \| x_n \| u\alpha \| \lambda\right) = \lambda \| \alpha = \alpha.$$

A parallel composition of two or more languages is:

$$L_1 \| L_2 \| \cdots \| L_n = \left\{x_1 \| x_2 \cdots \| x_n : x_1 \in L_1, \; x_2 \in L_2, ..., x_n \in L_n\right\}.$$

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

$$L_1 \cap L_2 = \left\{x : x \in L_1 \text{ and } x \in L_2\right\}.$$

The inverse operation of sentence x is a sentence whose characters 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 notice that TC languages are closed with respect to any finite number of union, intersection, inversion, parallel composition and concatenation operations performed in any order. A system consisting of a basic set of a set $\text{T}\Phi = \left\{\text{TC}_i\right\}$ 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 $\text{T}\Phi = \{\text{TC}_i\}$.

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.

Thus, the WP is taken as an indivisible element of the production system. The WP corresponds to α_i N.P.Buslenko aggregate of [2]. Each WP is assigned a certain number of operations *d*. The set of operations is denoted by *D*. In addition, they function in time and have spatial coordinates. The set WP are interconnected by arcs and forms a communication network with flows (meaning information flows, 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 WP 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 an *R*-network.

During the functioning of the system, the network structure may change over time: old arcs and vertices are canceled, and new ones are added. Such networks are called situational or RC networks. When solving a certain class of problems over time (t_1, t_2), one of the operations assigned to it is performed at each α.

Therefore, the construction of the network itself and the definition of the attributed operation is the main task of system research. At a certain period of time, the network can be depicted as a directed graph of constant structure (Fig. 2.).

This representation corresponds to the definition of the functioning table, and R-, RC networks are presented in the form of TF. On this network, you can fix the flow parameters and the network operation mode in time.

3. The task of synthesis of the control unit (monitor) and the process of synthesis of automata for the implementation of the control operation.

Consider the process of synthesis of automata for the implementation of the control operation. We introduce the concept of a monitor – a special unit that performs the control operation in *A*-systems. The monitor is an *A*-system unit or complex designed to control the units of this system. The monitor is responsible for the distribution of the relevant input information for the units of the system, and also controls the processes of functioning of the units. For example, in operating systems, there may be a main memory monitor and a central processor monitor, and in machining systems, monitors for local control of a group of computer numerical control machines, etc. The monitor has been granted the right to access information tables and other control structures of the *A*-system associated with the control unit.

We call a universal monitor such a monitor, which consists of a control unit that receives a set of instructions for execution as input. A model of a universal *A*-system monitor can be a state machine with store memory [9]. To obtain the final alphabet of store symbols, we use the synthesis technique of control automata according to graph-schemes of algorithms (GSA).

According to [5–7], the technique consists of two stages. The first stage consists in marking the GSA with symbols (labels) according to the following algorithm [5-7].

Step 1. The symbol α_1 marks the entrance of the vertex following the initial one, and the output of the final vertex.

Step 2. The inputs of all vertices following the operator are marked with symbols $\alpha_2, \alpha_3, ..., \alpha_n$.

Step 3. The inputs of various vertices, with the exception of the final one, are marked with different symbols.

At the first stage, we get the marked GSA. If now each $\alpha_1, \alpha_2, ..., \alpha_n$ symbol is associated with a vertex of the graph by automata or, if there is a transition path from α_m to α_3 on the GSA, then on the graph of the automaton the vertex α_m is connected by an arc with the vertex α_3 directed from α_m to α_3. At the beginning of the arc is written the conjunction $x(\alpha_m, \alpha_3)$ formed for this path; at the end of the arc, a subset of operations $y(\alpha_m, \alpha_3)$ is formed from the operator vertex through which the path passes [6].

The general algorithm for constructing monitors for the A-system is graphically presented in Fig. 3.

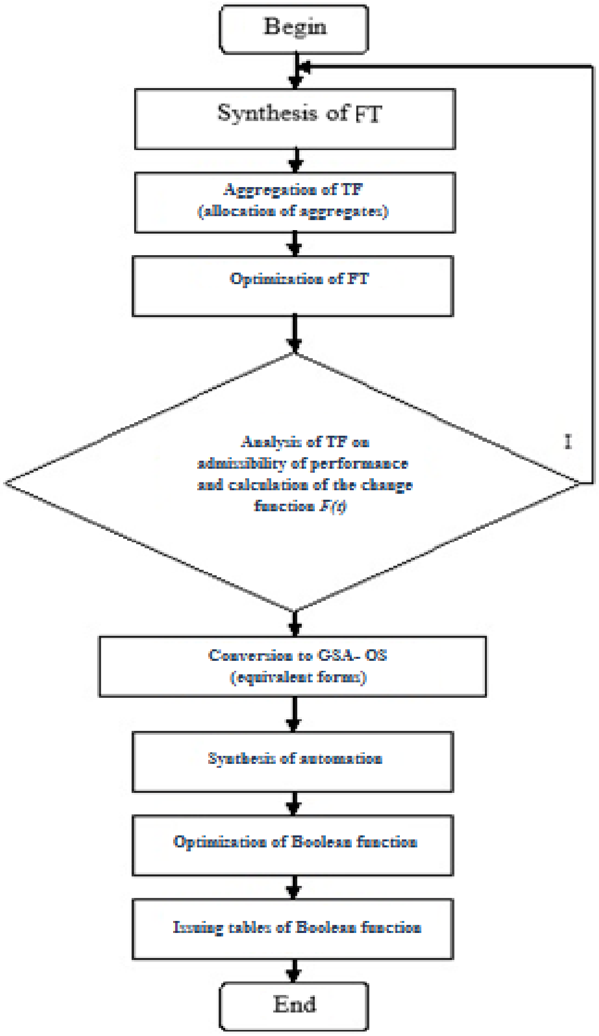

Figure 3. Scheme of building a monitor for A-system

A description of the process in the form of an OS or GSA often leads to significant duplication of the same operations when recording them. In this regard, the synthesis of an automaton for a monitor displays in a compact form the OS or GSA into a system of Boolean functions. Therefore, the Boolean function system is the control table for the implementation of A-system monitors in the form of automata with store memory.

4. Algorithmic method for managing complex systems based on function tables.

Algorithmization of control of complex systems in our case is to find an algorithm that allows for any system with a discrete nature of production on its functioning tables to find a functional characteristic. In this sense, algorithmization is, in essence, a universal method for modeling the process of managing complex systems with functioning tables.

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

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

The second level – local control based on stationary functioning tables – loads the corresponding shopping centers into control monitors, and then initiates the process of launching the WP in accordance with the reference stationary functioning tables and monitors the execution of operations on the specified functioning tables control monitor.

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

Consider the management model of complex systems. The ratio of the form R: D, where R is the set of jobs, and V the set of parts of parts obtained as a result of building a dynamic functioning table is an integral part of the wider R: D: O ratio because of the transitivity of all relations, you can build the following functioning tables transformation:

$$R:D \to D:0 \to 0:\cup\left(\cup=\{CP\}\right).$$

Therefore, R: U. Such a sequential transformation is carried out at the upper level of management, when, when planning the production process, we get a functioning tables with a ratio of the form R: D, and stationary functioning tables with a relationship of the type D: O and O: U are obtained as a result of the workings of the technological preparation of production. When constructing dynamic functioning tables with the R: D ratio, we will use the scheme shown in Fig.2.

Analysis of the functioning tables on the admissibility of the solution is based on the resolution of two equations in the matrix form:

$$\overline{Y}_1 = Y_0 + A\cdot\overline{\chi},\ A\overline{W} = 0 \qquad (1)$$

Based on the analysis of the first equation, the existence of a nontrivial fundamental solution system for a functioning tables, given in the form of a matrix *A* is determined. The matrix *A* is obtained from the matrices A^+ and A^-, describing the functioning tables functioning tables and is related to them $A = A^+ - A^-$. Finding a system of vectors *W* means saving functioning tables when marking one of the vectors. Here, as the vector *Y*, the initial state of the functioning tables is used, *X* is a vector of operations performed, *A* is the functioning tables in the presented in the matrix form, $\overline{Y}_1$ final state of functioning tables.

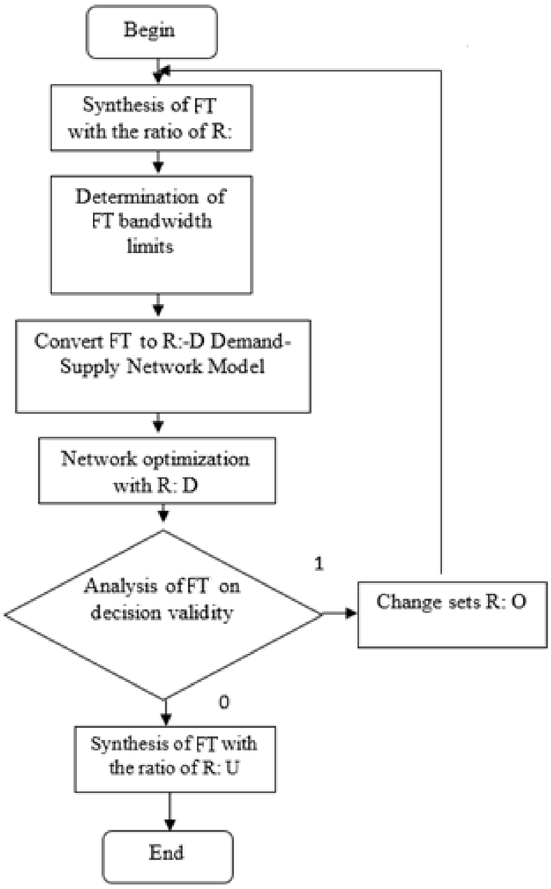
Begin
Synthesis of FT with the ratio of R:
Determination of FT bandwidth limits
Convert FT to R:-D Demand-Supply Network Model
Network optimization with R: D
Analysis of FT on decision validity
1
Change sets R: O
0
Synthesis of FT with the ratio of R: U
End

Figure 4. A scheme for constructing dynamic functioning tables with relations of the form R: O, R: U

The solution of the equations is based on the methods of linear algebra described in [6]. In determining the constraints on the capacity of the functioning tables, we rationing, calculate the upper and lower bounds of the positions that determine the material flow in the functioning tables: t_k , $\alpha_{ij} \le \chi_{ij} \le t_k$, here χ_{ij} is the normalized position value in the t_k – th time interval; α_{ij} – the value of the *i*-th position for *j*-th operation in functioning tables. Conversion to the network model is as follows:

$$\sum_{k=1}^{m}\sum_{j=1}^{n}\left[\left(c_{2j}^{k}-c_{1j}^{k}\right)x_{2j}^{k}+c_{1j}^{k}b_{j}^{k}\right]\to \min ,$$

where

c_{2j}^k – demand value of the k -th product for the j-th node;

c_{1j}^k – cost of the offer of the *k*-th product of the j-th node;

x_{ij}^k – flow of the *k*-th product from the *i*-th node to the *j* -th node;

b_j^k – *j* - th the node demand for the *k* -th product, provided that

$$\sum_{k=1}^{m}\chi_{2j}^{k}+S_{1j}=U_{2j},\, j=\overline{1,n},\; -\sum_{j=1}^{n}\chi_{1j}^{k}=a_{2}^{k}, k=\overline{1,n},\; -\sum_{j=1}^{n}S_{2j}=-\sum_{j=1}^{n}U_{2j}+\sum_{k=1}^{m}a_{2}^{k},$$

where a_2^k *j* – th node clauses the for *k* – th product

$$\chi_{2j}^{k}+\chi_{1j}^{k}=b_{j}^{k},\; S_{2j}+S_{1j}=U_{2j}+U_{1j}-\sum_{k=1}^{m}b_{j}^{k},$$

$$\chi_{ij}^{k},\; S_{ij}\ge 0,\; i=\overline{1,\,2},\; j=\overline{1,\,n},\; k=\overline{1,\infty}\text{ or }\chi_{2j}\le \chi_{ij}^{k},\; S_{2j}\le U_{2j}+U_{1j}-\sum_{k=1}^{m}b_{j}^{k}.$$

The solution of this network model can be obtained by one of the methods [7, 8] of solving the maximum flow problem.

Thus, the functioning tables with the R: D and R: O ratios give us the basic initial technological information necessary to control the software equipment. At the local control level, dynamic functioning tables with the R: U ratio are loaded into the control monitors in the order prescribed by the stationary functioning tables with an R: U. Control over the execution of processing operations is carried out by the stationary functioning tables with R: 0 minimum values of the functions defined in formula (1) [9-11].

5. Conclusion. The article explores the problem of control algorithmization of complex systems based on their standard description in the form of jobs and job complexes based on functioning tables. A description of the process of solving the object control problem based on an algorithmic approach is proposed, where a study of the algorithmic method of converting functioning tables for managing complex systems is proposed. Given:

– a unified standard description of the models of functioning of classes of cybernetic objects based on a unified scheme of representing objects using functioning tables;

– construction of control monitors based on state machines;

– development of an algorithmic method for constructing an complex systems control model based on their standard and unified description.

References

[1] Glushkov V and other 1987 *Logical design of discrete devices (*Kiev: Naukova Dumka)

[2] Buslenko N 1978 *Modeling of complex systems* (Moscow: Science)

[3] Zhuravlev Yu Finkelstein Yu 1977 *The scope of discrete programming methods (*Moscow: Economics)

[4] Kabulov V Kabulov A and Normatov I 2017 *Algorithmization in the theory control systems* (Tashkent: Navruz)
[5] Kabulov A V Normatov I H and Ashurov A O 2019 *Computational methods of minimization of multiple function*s Journal of Physics **1260**
[6] Kabulov A V and Normatov I H 2019 *About problems of decoding and searching for the maximum upper zero of discrete monotone functions* Journal of Physics **1260**
[7] Normatov I H 2018 *Principle of independence of continuation of functions multivalued logic from coding* Journal of Physics **1210**
[8] James L. Peterson 1981 *Petri net theory and the modeling of systems* (Prentice-Hall, Inc., Englewood Cliffs, N.J.)
[9] Baranov S Sklyarov V 1986 *Digital devices on programmable LSI with matrix structure (*Moscow: Radio and Communications)
[10] Ford L, Fulkerson D and Robert G. Bland 2010 *Flows in Networks* (Princeton University Press; Revised ed. edition)
[11] Hira N Ahuja 1976 *Construction Performance Control by Networks (*New York: United States)