

3. Oppelt, U. Improvement on fire detectors by using multiple sensors [Text] / U. Oppelt // Fire & Safety. – 2006. – Available at: <http://www.securitysa.com/regular.aspx?pkregularid=2502>
4. Ding, Q. Multi-Sensor Building Fire Alarm System with Information Fusion Technology Based on D-S Evidence Theory [Text] / Q. Ding, Z. Peng, T. Liu, Q. Tong // Algorithms. – 2014. – Vol. 7, Issue 4. – P. 523–537. doi: 10.3390/a7040523
5. Cheng, C. One fire detection method using neural networks [Text] / C. Cheng, F. Sun, X. Zhou // Tsinghua Science and Technology. – 2011. – Vol. 16, Issue 1. – P. 31–35. doi: 10.1016/s1007-0214(11)70005-0
6. Cestari, L. A. Advanced fire detection algorithms using data from the home smoke detector project [Text] / L. A. Cestari, C. Worrell, J. A. Milke // Fire Safety Journal. – 2005. – Vol. 40, Issue 1. – P. 1–28. doi: 10.1016/j.firesaf.2004.07.004
7. Radonja, P. Generalized profile function model based on neural networks [Text] / P. Radonja, S. Stankovic // Serbian Journal of Electrical Engineering. – 2009. – Vol. 6, Issue 2. – P. 285–298. doi: 10.2298/sjee0902285r
8. Tsai, Y. C. The Design and Implementation of Early Fire Detection and Hierarchical Evacuation Alarm System, Master Thesis [Text] / Y. C. Tsai // Graduate Institute of Networking and Communication Engineering. – Taiwan, 2007.
9. Ristic, J. Decision algorithms in fire detection systems [Text] / J. Ristic, D. Radosavljevic // Serbian Journal of Electrical Engineering. – 2011. – Vol. 8, Issue 2. – P. 155–161. doi: 10.2298/sjee1102155r
10. Andronov, V. Increase of accuracy of definition of temperature by sensors of fire alarms in real conditions of fire on objects [Text] / V. Andronov, B. Pospelov, E. Rybka // Eastern-European Journal of Enterprise Technologies. – 2016. – Vol. 4, Issue 5 (82). – P. 38–44. doi: 10.15587/1729-4061.2016.75063
11. Acclimate intelligent multi-criteria sensor MIX-2251TMB [Electronic resource]. – Mircom. – Available at: http://www.mircom.com/media/datasheets/CAT-5919_MIX-2251TMB_ACCLIMATE_Intelligent_Multi-Criteria_Sensor.pdf

*Розглянуто питання підвищення ефективності діагностики працездатності автоматизованих систем за допомогою експертних діагностичних систем на основі розробленого методу пошуку діагнозу на конфліктній множині вхідних даних. Пропонується використання математичних методів роботи зі структурою типу «*t*-арні кортежі на основі впорядкованих множин». Отримано математичні та графічні залежності мінімізації часу пошуку діагнозу для розробленого методу*

Ключові слова: експертна діагностична система, діагностика відмов, структура організації даних, оцінка часу виконання алгоритму

*Рассмотрены вопросы повышения эффективности диагностики работоспособности автоматизированных систем при помощи экспертных диагностических систем на основе разработанного метода поиска диагноза на конфликтном множестве входных данных. Предлагается использование математических методов работы со структурой типа «*t*-арные кортежи на основе упорядоченных множеств». Получены математические и графические зависимости минимизации времени поиска диагноза для разработанного метода*

Ключевые слова: экспертная диагностическая система, диагностика отказов, структура организации данных, оценка времени исполнения алгоритма

1. Introduction

In recent years, the way people control various kinds of technology has radically changed. The leading place in this

process is taken by automated systems, SCADA (Supervisory Control and Data Acquisition) systems included. Modern SCADA systems are widely spread all over the world to control technological processes in different areas such as

UDC 681.516

DOI: 10.15587/1729-4061.2017.108454

EVALUATION TO DETERMINE THE EFFICIENCY FOR THE DIAGNOSIS SEARCH FORMATION METHOD OF FAILURES IN AUTOMATED SYSTEMS

O. Syrotkina

Assistant

Department of Software Engineering*

E-mail: syrotkina@hotmail.com

M. Alekseyev

Doctor of Technical Sciences, Professor*

E-mail: 1948keila@gmail.com

O. Aleksieiev

PhD, Associate Professor**

E-mail: aleksey.alekseev@gmail.com

*Department of Software Engineering***

Department of System Analysis and Management*

***National Mining University

Dmytra Yavornytskoho ave., 19, Dnipro, Ukraine, 49600

power engineering, the military, medicine, mining, metallurgy, natural gas, the environment, etc. The given crucial application fields show that these systems should meet strict requirements to ensure their fault-tolerance and reliability in the process of their work at mission-critical facilities.

It is known that most of the failures in the system operation occur due to errors of operational and maintenance personnel who have insufficient qualifications (and/or limited time resources with increased information flow) for supervisory control, quality assistance, and restoration of system operability. Therefore, the pressing problem is to develop highly reliable and fault-tolerant SCADA systems by improving the methods of their automatic self-diagnostics in real time with the possibility of auto-recovery of their operability after reversible failures.

2. Literature review and problem statement

The modern SCADA system is a distributed multilevel and multitasking hardware-software complex (HSC) operating in real time [1, 2]. It is a complex, dynamic and difficult-to-formalize diagnostic system with a changing structure and functionality in the process of its life cycle. The effectiveness of SCADA work and the reliability of the data at all levels of its hierarchy depend on the operability of backbone nodes, data transmission channels, peripheral equipment and the consistency of the software system in general [3].

Problems in SCADA maintenance due to low-level diagnostics and the limited ability to restore SCADA operability after reversible failures have a negative impact on the quality of control for the Technological Control Object (TCO).

Models and methods of failure diagnostics in automated systems are considered in the works of the authors [4–10].

The methodology used to determine the efficiency of technological diagnostic systems is considered in work [4]. It is based on the calculations of diagnostic object indicators as well as technical diagnostic tools. The author also uses the indicator of the diagnosis method effectiveness in terms of the accuracy of diagnostic operations and includes the length of diagnosis time.

The method of SCADA diagnostics is presented in work [5]. It is based on a specific, unique alarm (index analysis) generated by the system in the event of a malfunction. The main idea is that each alarm relates to a particular fault in the system. These alarms are always generated by the system when a fault occurs, but are never generated for any other fault. The alarm is activated when a fault is detected in the system. It is generated according to predetermined relevance indexes for a changing set of faults in different contexts. However, this method does not always make it possible to identify the required fault in a changing set. In this case, this fault is considered indistinguishable and cannot be reliably diagnosed.

The application of SCADA to monitor a power station in Amareleja (Spain) is described in work [6]. The system allows detection of not only current faults, but also allows human operators to conduct long-term forecasts of equipment status and abnormal trends in the operation of the facility. However, the system is aimed at diagnosing the operation of the TCO. It does not perform at all levels of the system hierarchy from the TCO to SCADA, which generally

reduces the reliability of diagnostics for the entire complex “object-system”.

The method of fault diagnostics for oil transformers is given in work [7]. It is based on the application of the operational control method performed by SCADA which combines information from a set of data sources and uses neural networks with back propagation of error. This diagnostic method gives a high coefficient of the diagnostic accuracy (up to 93 %) because it uses flexible strategies applied by expert systems. However, the diagnostic method presented requires large data sets to be processed, which significantly affects the time duration to establish the diagnosis.

Methods of fault diagnostics for wind turbines controlled by SCADA are examined in work [8]. The authors provide a review of a number of diagnostic methods, including methods based on artificial neural networks, fuzzy-logic methods, and combined methods such as adaptive neuro-fuzzy inference. The selection of an optimal set of monitored parameters and quality of data (a noiseless signal from sensors) is very important to establish the diagnosis. The frequency of data interrogation by SCADA is critical too. If this frequency is measured in minutes, then the important information for diagnostics can be lost. Therefore, it is necessary to preprocess the raw data from SCADA to establish an accurate diagnosis, especially when time limits apply.

The method of fault detection for industrial automation processes is described in work [9]. The method takes into account the work of automated systems with insignificant noise of the data measured and the possibility of constant monitoring of a complete set of monitored parameters. The authors use neural networks and regression analysis methods to approximate the functional dependence of continuous variables for the technological processes over time. They also show the application of the HMM (Hidden Markov Model) to monitor continuing process variables as a sequence of hidden states of discrete variables under different system modes. The detection of current faults at the same time reaches 87 %, but the failure prediction error rates for various operating modes of the system remain high enough.

The concept of automated system diagnostics based on Latent Variable (LV) models is considered in work [10]. The authors define process parameters which exceed the monitoring statistics on the basis of the archived data array collected by SCADA. These parameters are related to the operation of the equipment and the behavior of the process or the malfunction in the system. The approach is based on the ability of the operational personnel of the enterprise to give an expert assessment of the current technological process and the operation of the equipment. Decision support systems based on knowledge can be applied to automate the process of forming an expert judgment. In this case, LV models must be unique and identifiable. It also requires processing large arrays of missing data, monitoring their integrity within the LV area which is set by the trainable sample. Restrictions on the use of LV models (built on the archive data of the system) consist of limiting the space of latent variables determined by these archival data. LV models also cannot be used to extrapolate TCO operation modes for which monitoring statistics are not available.

Having analyzed the literature in the field of automated system diagnostics, we can make the conclusion that there has been no adequate resolution for the following problems in the methods of diagnostics applied today:

- the intensity of the diagnostic information flows causes significant difficulties when it is necessary to process them operationally by the corresponding operational services of the enterprise;
- a large amount of data is considered low-level diagnostic information which requires the highly professional knowledge of technologists, IT specialists, and system integrators in order to process;
- for most automated control systems, the Expert System (ES) methodology is used only to diagnose the state of the Technological Control Object (TCO) and not the states of the system hierarchy levels from TCO to SCADA.

Thus, today the unsolved part of the problem is the development of methods for complex automatic high-level diagnostics of the operability at all levels of the system hierarchy in the automated system from TCO to SCADA with the minimization of the diagnosis time to real time.

3. The aim and objectives of the study

The aim of the research is to increase the efficiency of the automated control system for an industrial enterprise by providing automatic self-diagnostics for its components – SCADA software and hardware in real time.

It is necessary to solve the following tasks to achieve the set aim:

- formation of a mathematical model to diagnose the operability of an integrated complex TCO – SCADA using the methodology of Expert Diagnostic Systems;
- conducting a study of the patterns for the formation of the diagnostic search space with reference to the knowledge base of the Expert System for a conflicting set of input data – diagnostic codes generated by the SCADA system in the event of failures;
- development of a method to diagnose failures in an integrated complex TCO – SCADA with the minimization of the diagnosis search formation time;
- performing an analysis of the effectiveness for the diagnostic method developed.

4. Mathematical model to diagnose the operability of automated systems based on the methodology of expert systems

We wish to put forward a scheme of interaction between SCADA and the Expert System (see Fig.1) to solve the problem of automatic SCADA self-diagnostics using an Expert Diagnostic System as the diagnostic subsystem [11, 12].

The following designations are used to identify the above symbols in Fig. 1: *DRTES* – Diagnostic Real Time Expert System; *DB* – Database (basic components of the subject area, their properties and the relationships between

them); *KB* – Knowledge Base (rules to construct structure variants from database components); *IM* – Inference Machine; *SDC* – System Diagnostic Codes; *Human Machine Interface* (HMI) is the interface between processes and human operators.

SCADA subsystem based on the Expert System derives a conclusion about the SCADA technical state, its functional components and processes. The conclusion is based on the analysis of the input data of the Expert System diagnostic codes generated by SCADA in real time.

The diagnostic model of TCO and SCADA is shown in Fig. 2.

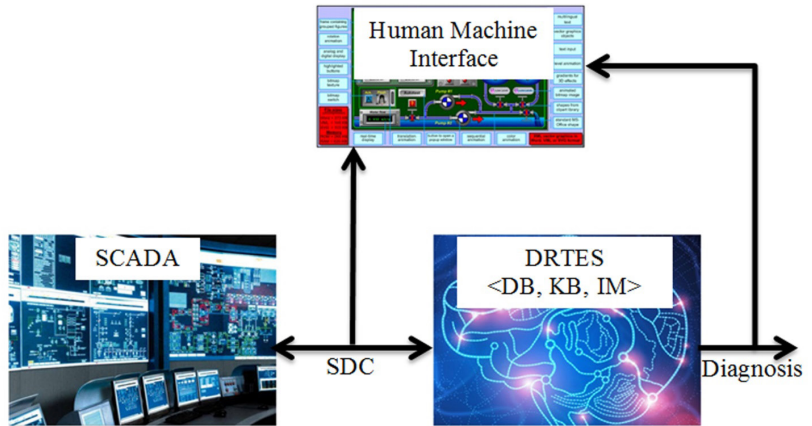


Fig. 1. Scheme of interaction between SCADA and the Expert System

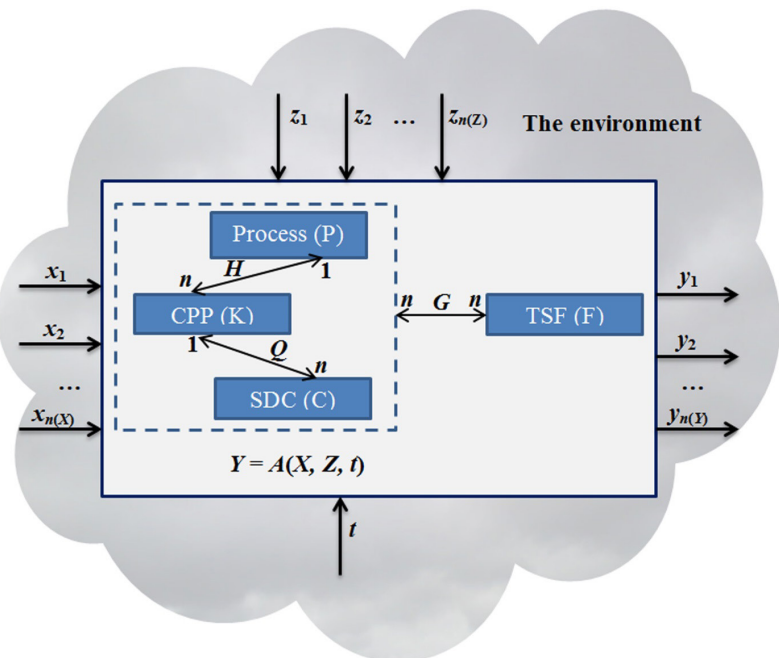


Fig. 2. Diagnostic model of TCO and SCADA

The following designations are used to identify the above symbols in Fig. 2: *X* – a set of input parameters (these include controllable parameters of TCO and SCADA); *Y* – a set of output parameters (this is a diagnosis of the TCO and SCADA state; localization and identification of failures in hardware and software of TCO and SCADA; recommendations for the recovery/auto recovery of their operability; recommendations for failure prevention, repair, hardware

upgrades, software updates and recovery/auto recovery included); Z – a set of factors which can impede the system’s reliability (for example, a power failure, a hardware failure, a DoS attack, external electromagnetic effects on system equipment, etc.); A – the operator of the model represented by a set of algorithms and/or functions; P – processes. A process is an executable program module that runs on a backbone node. A set of processes form SCADA software; K – Control Process Points (CPP). A Control Process Point is a virtual location with a given instruction in the executable process code. When a failure occurs, it is possible to give the result based on the diagnostic code if the process passed/didn’t pass its control point. In other words, it is a diagnostic code about the process execution error. The set of diagnostic codes allows us to determine failures in the system; C – System Diagnostic Codes (SDC). These include software error codes, exceptions of classes and functions, process return codes, states of uncertainty (i. e. the partial or complete absence of information after the timeout is expired, etc.). Diagnostic Codes are generated in the process of SCADA operation; F – Types of SCADA Failures (TSF); H – the relationship between a Control Process Point and Process. Q – the relationship between System Diagnostic Codes and Control Process Points as a disjunction of a set with possible values of the SDC for a given CPP; G – the relationship between a given set of SDC determined at the CPP to the relevant TSP.

Thus, the relationship G determines the TSF in SCADA using the set of SDC coming at the input of the ES.

$G_1 \subset G$ – the relationship between a set of SDC and TSP which is completely determined at the set of SDC.

Suppose the SCADA subsystem based on the ES receives a set of input data $X(\Delta t)$ ordered by the time of generation for SDC in SCADA:

$$X(\Delta t) = (C_x, K_y, P_z)(\Delta t) = \{(C_x, K_y, P_z)(t_1), \dots, (C_x, K_y, P_z)(t_i), \dots, (C_x, K_y, P_z)(t_k)\}, \quad (1)$$

where $X(\Delta t)$ – a subset SDC C_x , generated by SCADA in the event of off-nominal passing /non-passing a subset CPP K_y which belongs to a subset of processes P_z ; $\Delta t = (t_k - t_1)$ – the time interval to generate the SDC in SCADA for the given input data presented.

Suppose the Knowledge Base (KB) of the ES for diagnosing a failure $f \in F$ contains an alternative set of rules $G_f \subset G$ for the logical derivation of the diagnosis, determined at the set $X_f = (C_x, K_y, P_z)_f$.

$$f = G_f(X_f), \quad (2)$$

$$X_f = \{x_1, x_2, \dots, x_i, \dots, x_n\}_f, \quad (3)$$

$$G_f = \{g_1, g_2, \dots, g_j, \dots, g_m\}_f. \quad (4)$$

Skolem standard form for the formula of the deductive derivation rule in the Expert System Knowledge Base is as follows [13, 14]:

$$\vdash \forall_{i=1}^n x_i (\omega(x_1, x_2, \dots, x_n)) \leftarrow \exists_{i=1}^n (\bigwedge_{j=1}^m g_j(x_1, x_2, \dots, x_n)), \quad (5)$$

which is interpreted as follows: “it is required to prove the truth of deducibility for the conclusion from a set of true premises”.

The following designations are used in formula (5): x_i – a propositional variable corresponding to a certain SDC in the Expert System Database determined at the CPP; g_j – a condition (premise), corresponding to accepted rules of failure determination; ω – conclusion; n – the number of propositional variables; m – the number of conditions.

5. Formation of the diagnosis search space for failures in the automated system at the conflicting set of input data

We can determine the diagnosis search space as follows:

$$F_u = \{f_1, f_2, \dots, f_{n(F_u)}\} = \{f \mid f \in G_f(X_f), X_f \cap X(\Delta t) \neq \emptyset\}. \quad (6)$$

Consider an example of the diagnosis search space formation [13].

For instance:

X – a set of SDC determined at CPP in the Expert System Database;

G_1 – a system of rules in the Expert System Knowledge Base to determine a failure in the system;

$X(\Delta t)$ – a set of input data in the Expert System for a time interval Δt .

We have:

$$X = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}, x_{11}, x_{12}, x_{13}, x_{14}, x_{15}\};$$

$$G_1 = \{g_1, g_2, g_3, g_4, g_5, g_6, g_7, g_8, g_9, g_{10}\};$$

$$f_1 = g_1(x_1, x_4, x_6, x_{11}) = G_{f_1}(X_{f_1});$$

$$f_2 = g_2(x_2, x_5, x_9, x_{11}, x_{15}) = G_{f_2}(X_{f_2});$$

$$f_3 = g_3(x_2, x_3, x_7, x_{10}) = G_{f_3}(X_{f_3});$$

$$f_4 = g_4(x_1, x_3, x_7) = G_{f_4}(X_{f_4});$$

$$f_5 = g_5(x_7, x_{13}, x_{14}, x_{15}) = G_{f_5}(X_{f_5});$$

$$f_6 = g_6(x_4, x_8, x_{12}, x_{13}) = G_{f_6}(X_{f_6});$$

$$f_7 = g_7(x_5, x_8, x_{15}) = G_{f_7}(X_{f_7});$$

$$f_8 = g_8(x_2, x_3, x_8, x_{14}) = G_{f_8}(X_{f_8});$$

$$f_9 = g_9(x_6, x_{11}, x_{12}, x_{15}) = G_{f_9}(X_{f_9});$$

$$f_{10} = g_{10}(x_9, x_{10}, x_{13}) = G_{f_{10}}(X_{f_{10}});$$

$$X(\Delta t) = \{x_1, x_3, x_7, x_{12}\}.$$

Required to find: the diagnosis search space for a set of input data $X(\Delta t)$. To do this, it is necessary to form subsets Y_m^n on the basis of the set $2^{X(\Delta t)}$. The elements of the given set are represented by m -tuples, where: $n = |X(\Delta t)|$ – the number of elements in the set of input data $X(\Delta t)$; m – the length of the tuple.

We consolidate the results of the formation of Y_m^n based on the set of input data $X(\Delta t)$ into Table 1.

There is a set of rules in the Expert System Knowledge Base $G_{m,j}(y_{m,j}^n)$ which is relevant to each element $y_{m,j}^n \in Y_m^n \subset 2^{X(\Delta t)}$ formed at the set of input data in the Expert System $X(\Delta t)$. These input data form the diagnosis search space $F_{m,j}$:

$$X_{f_i} \cap 2^{X(\Delta t)} = y_{m,j}^n, \quad (7)$$

$$X_{f_i}^- = X_{f_i} \setminus y_{m,j}^n, \quad (8)$$

$$G_{m,j} : y_{m,j}^n \rightarrow F_{m,j}, \quad (9)$$

$$F_{m,j} = \{f_1, \dots, f_N\}_{m,j} = \{G_{f_1}(X_{f_1}), \dots, G_{f_N}(X_{f_N})\}_{m,j}, \quad (10)$$

where $X_{f_i}^-$ – the set of SDC to derive a diagnosis f_i and the values of its elements are not determined at the set of the input data $X(\Delta t)$.

Table 1

Representation of m -tuples on the basis of input data $2^{X(\Delta t)}$

| m | Y_m^n | $ Y_m^n $ |
|-----|---|-----------|
| 1 | $Y_1^4 = \{x_1, x_3, x_7, x_{12}\}$ | 4 |
| 2 | $Y_2^4 = \{(x_1, x_3), (x_1, x_7), (x_1, x_{12}), (x_3, x_7), (x_3, x_{12}), (x_7, x_{12})\}$ | 6 |
| 3 | $Y_3^4 = \{(x_1, x_3, x_7), (x_1, x_3, x_{12}), (x_1, x_7, x_{12}), (x_3, x_7, x_{12})\}$ | 4 |
| 4 | $Y_4^4 = \{(x_1, x_3, x_7, x_{12})\}$ | 1 |

If $X_{f_i}^- = \emptyset$, then the set of rules G_{f_i} is completely determined at the set of input parameters $X(\Delta t)$ for the current state of the Expert System Knowledge Base. Otherwise, it is partly determined.

We consolidate the results of the alternative search space formation into Table 2.

Table 2

Alternative spaces for diagnosis search

| No. | $y_{m,j}^n$ | F_u | $X_{f_i}^-$ |
|-----|---|-------------|--|
| 1 | $y_{1,1}^4 = X_{f_1} \cap 2^{X(\Delta t)} = x_1$ | f_1 | $X_{f_1}^- = \{x_4, x_6, x_{11}\}$ |
| 2 | $y_{1,2}^4 = X_{f_8} \cap 2^{X(\Delta t)} = x_3$ | f_8 | $X_{f_8}^- = \{x_2, x_8, x_{14}\}$ |
| 3 | $y_{1,3}^4 = X_{f_5} \cap 2^{X(\Delta t)} = x_7$ | f_5 | $X_{f_5}^- = \{x_{13}, x_{14}, x_{15}\}$ |
| 4 | $y_{1,4}^4 = (X_{f_6} \cap 2^{X(\Delta t)}) \vee (X_{f_9} \cap 2^{X(\Delta t)}) = x_{12}$ | f_6 | $X_{f_6}^- = \{x_4, x_8, x_{13}\}$ |
| | | f_9 | $X_{f_9}^- = \{x_6, x_{11}, x_{15}\}$ |
| 5 | $y_{2,4}^4 = (X_{f_3} \cap 2^{X(\Delta t)}) = (x_3, x_7)$ | f_3 | $X_{f_3}^- = \{x_2, x_{10}\}$ |
| 6 | $y_{3,1}^4 = (X_{f_4} \cap 2^{X(\Delta t)}) = (x_1, x_3, x_7)$ | f_4 | \emptyset |
| 7 | $y_{2,1}^4 = (x_1, x_3)$ | \emptyset | \emptyset |
| | $y_{2,2}^4 = (x_1, x_7)$ | | |
| | $y_{2,3}^4 = (x_1, x_{12})$ | | |
| | $y_{2,5}^4 = (x_3, x_{12})$ | | |
| | $y_{2,6}^4 = (x_7, x_{12})$ | | |
| | $y_{3,2}^4 = (x_1, x_3, x_{12})$ | | |
| | $y_{3,3}^4 = (x_1, x_7, x_{12})$ | | |
| | $y_{3,4}^4 = (x_3, x_7, x_{12})$ | | |
| | $y_{4,1}^4 = (x_1, x_3, x_7, x_{12})$ | | |

According to Table 2, the diagnosis search space is defined as follows:

$$y_{3,1}^4 \rightarrow F_{3,1} = f_4, \quad (11)$$

$$y_{1,4}^4 \rightarrow F_{1,4} = \{f_6, f_9\}, \quad (12)$$

$$F_u = F_{3,1} \cup F_{1,4} = \{f_4, f_6, f_9\}. \quad (13)$$

There are priority rules with absolutely determined parameters for the elements of Boolean $2^{X(\Delta t)}$ with maximum cardinality to establish the diagnosis.

Formation of the diagnosis search space using the rules from the Expert System Knowledge Base with partially defined parameters is carried out by finding a complement to the Boolean element $2^{X(\Delta t)}$ for a subset of an ordered basis set of arbitrary cardinality.

6. Application of data structure m -tuples in the method for failure diagnostics of an automated system

Consider the properties of the input data set in the Expert System and methods to work with them in detail [15]. The set is based on the elements of m -tuples Y_m^n as part of Boolean $2^{X(\Delta t)}$.

Suppose we have a finite ordered increasing basis set:

$$X = \{x_1, \dots, x_i, x_{i+1}, \dots, x_n\} = \{x_i \mid 1 \leq i \leq n, \forall i \in [1, n] \rightarrow x_i < x_{i+1}\}, \quad (14)$$

where i – the ordinal number of the element x in the set X ; $n=|X|$ – the number of elements or the cardinality of the set X ; “ $<$ ” – comparison operator for the elements of the set X , defined individually, depending on the type of elements of the set X ; 2^X – Boolean of the set X ; $|2^X|=2^n$ – cardinality of Boolean.

$$2^X = \{\emptyset, Y_1^n, \dots, Y_m^n, \dots, Y_n^n\} = \left\{ Y_m^n \mid 1 \leq m \leq n, k_m^n = |Y_m^n| = \frac{n!}{(n-m)! \cdot m!} \right\}, \quad (15)$$

where Y_m^n – a subset of Boolean. The elements of the given subset are m -tuples. They consist of the elements from the set X . The set X is ordered by a right-hand search of its indexes in m -tuples from the lower boundary to the upper. k_m^n – cardinality of the subset Y_m^n .

$$Y_m^n = \{y_{m,j}^n \mid y_{m,j}^n = (x_{i_1}, \dots, x_{i_m}), 1 \leq \eta \leq m, \alpha_{m,\eta}^n \leq i_\eta \leq \beta_{m,\eta}^n, \alpha_{m,\eta}^n = \eta, \beta_{m,\eta}^n = n - m + \eta, \forall \eta \in [1, m] \rightarrow x_{i_\eta} < x_{i_{\eta+1}}\}, \quad (16)$$

where $y_{m,j}^n$ – m -tuple is an element of the set Y_m^n and consists of the elements from the basis set X ; m – the length of the tuple; j – an index (sequence number) of a tuple in the set of tuples Y_m^n ; η – an element index in the tuple $y_{m,j}^n$; i_η – an index of a tuple element $y_{m,j}^n$ in the basis set X ; $\alpha_{m,\eta}^n$ – lower boundary of change i_η in m -tuple $y_{m,j}^n$; $\beta_{m,\eta}^n$ – upper boundary of change i_η in m -tuple $y_{m,j}^n$.

Suppose I – an ordered increasing set of indexes which are elements of the set X :

$$I = \{1, 2, \dots, n\} = \{i \mid 1 \leq i \leq n\}. \quad (17)$$

J_m^n – an ordered increasing set of indexes which are elements of the set Y_m^n :

$$J_m^n = \{1, 2, \dots, k_m^n\} = \{j \mid 1 \leq j \leq k_m^n\}. \tag{18}$$

I_m^n – a set of tuples which contains indexes of basic elements for the tuples of the set Y_m^n :

$$I_m^n = \{i_{m,1}^n, i_{m,2}^n, \dots, i_{m,k_m^n}^n\} = \{i_{m,j}^n \mid i_{m,j}^n = (i_1, \dots, i_\eta, \dots, i_m), 1 \leq j \leq k_m^n, i_\eta \in I, 1 \leq \eta \leq m, \eta \leq i_\eta \leq n - m + \eta, \forall \eta \in [1, m] \rightarrow i_{\eta+1} - i_\eta \geq 1\}. \tag{19}$$

$i_{m,j}^n$ – a tuple of indexes of basic elements in the tuple $y_{m,j}^n$.

We can specify the following dependencies between the ordered sets given above:

– indexing elements from the ordered basis set X :

$$\phi_1 : X \rightarrow I; \tag{20}$$

– searching for elements of the basis set X using its index:

$$\phi_1^{-1} : X \leftarrow I; \tag{21}$$

– forming ordered sets of m -tuples Y_m^n from the elements of the ordered basis set X with cardinality n :

$$\phi_2 : X \rightarrow Y_m^n; \tag{22}$$

– projecting ordered sets of m -tuples Y_m^n onto elements of the basis set X with cardinality n :

$$\phi_2^{-1} : X \leftarrow Y_m^n; \tag{23}$$

– forming ordered sets of m -tuples I_m^n from the elements of the ordered basis set I with cardinality n :

$$\phi_3 : I \rightarrow I_m^n; \tag{24}$$

– projecting ordered sets of m -tuples I_m^n onto elements of the basis set I with cardinality n :

$$\phi_3^{-1} : I \leftarrow I_m^n; \tag{25}$$

– indexing the elements of ordered sets Y_m^n , which consist of m -tuples $y_{m,j}^n$:

$$\phi_4 : Y_m^n \rightarrow J_m^n; \tag{26}$$

– searching for elements of the set Y_m^n using its index:

$$\phi_4^{-1} : Y_m^n \leftarrow J_m^n; \tag{27}$$

– indexing the elements of ordered sets of m -tuples Y_m^n using the elements of ordered sets of m -tuples I_m^n :

$$\phi_5 : Y_m^n \rightarrow I_m^n; \tag{28}$$

– searching for elements of ordered sets of m -tuples Y_m^n using the elements of ordered sets of m -tuples I_m^n :

$$\phi_5^{-1} : Y_m^n \leftarrow I_m^n. \tag{29}$$

The following methods are proposed for consideration to carry out the search and projection of elements in ordered sets:

– determining the values in the tuple of indexes $i_{m,j}^n$ using the ordinal number j for the element of the ordered set Y_m^n [15]:

$$A_1 : j \rightarrow i_{m,j}^n; \tag{30}$$

– determining the ordinal number j for the element of the ordered set Y_m^n using the tuple of indexes i_m^n :

$$A_2 : j \leftarrow i_{m,j}^n; \tag{31}$$

– determining the complement to the tuple of indexes $i_{m,j}^n$ to the basis set I :

$$A_3 : \overline{i_{m,j}^n} \leftarrow i_{m,j}^n; \tag{32}$$

– determining the complement to the element $y_{m,j}^n$ to the basis set X :

$$A_4 : \overline{y_{m,j}^n} \leftarrow y_{m,j}^n. \tag{33}$$

Consider formal rules to determine the dependency $j \leftarrow i_{m,j}^n$ based on the method A_2 .

We have:

$$X = \{x_i \mid n = |X|, 1 \leq i \leq n, \forall i \in [1, n] \rightarrow x_i < x_{i+1}\},$$

$$m, i_{m,j}^n = (i_1^j, \dots, i_\eta^j, \dots, i_m^j),$$

Required to find: $j \leftarrow i_{m,j}^n$.

Solution:

1. We verify that our input data are correct: X and m

$$\forall i \in [1, n] \rightarrow x_i < x_{i+1}, 1 \leq m \leq n. \tag{34}$$

2. We form Table 3 with the dimension $(m+1) \times 4$ to calculate the parameters $\alpha_{m,\eta}^n, \beta_{m,\eta}^n, \mu_{m,\eta}^n$:

$$\forall \eta \in [1, m] \rightarrow (\alpha_{m,\eta}^n = \eta, \beta_{m,\eta}^n = n - m + \eta, \mu_{m,\eta}^n = m - \eta). \tag{35}$$

Table 3

Areas of index definitions i_h

| η | $\alpha_{m,\eta}^n$ | $\beta_{m,\eta}^n$ | $\mu_{m,\eta}^n$ | } m |
|--------|---------------------|--------------------|------------------|-------|
| 1 | 1 | $n - m + 1$ | $m - 1$ | |
| ... | η | $n - m + \eta$ | $m - \eta$ | |
| m | m | N | 0 | |

3. We verify that input data are correct:

$$i_{m,j}^n = (i_1^j, \dots, i_\eta^j, \dots, i_m^j). \tag{36}$$

The condition of correctness $i_{m,j}^n$:

$$\forall i_\eta^j \in [1, m] \rightarrow (\alpha_{m,\eta}^n \leq i_\eta^j \leq \beta_{m,\eta}^n),$$

$$\forall i_\eta^j \in [1, m] \rightarrow (i_\eta^j - i_{\eta-1}^j \geq 1). \tag{37}$$

4. We form Table 4 with the dimension $(\beta_{m,1}^n - \alpha_{m,1}^n + 4) \times (2m + 1)$ to calculate parameters $C_{n-i_\eta}^{m-\eta}$ which determine the number of elements in the set Y_m^n . These elements are obtained by performing an ordered right-hand search of indexes beginning with i_η in the given area:

$$(\forall \eta \in [1, m]) \& (\forall i_\eta \in [\alpha_{m,\eta}^n, \beta_{m,\eta}^n]) \rightarrow$$

$$\rightarrow C_{n-i_\eta}^{m-\eta} = \frac{(n - i_\eta)!}{((n - i_\eta) - (m - \eta))! \times (m - \eta)!}. \tag{38}$$

Table 4

Calculation of parameters $C_{n-i_\eta}^{m-\eta}$

| | | | | | |
|---------------------|------------------|------------------------------|------------------|------------------------------|-----|
| η | 1 | 2 | ... | M | |
| $\mu_{m,\eta}^n$ | $m-1$ | $m-2$ | ... | 0 | 1 |
| i_η | i_1 | $C_{n-i_1}^{m-\eta}$ | i_2 | $C_{n-i_2}^{m-\eta}$ | ... |
| $\alpha_{m,\eta}^n$ | $\alpha_{m,1}^n$ | $C_{n-\alpha_{m,1}^n}^{m-1}$ | $\alpha_{m,2}^n$ | $C_{n-\alpha_{m,2}^n}^{m-2}$ | ... |
| ... | ... | ... | ... | ... | ... |
| $\beta_{m,\eta}^n$ | $\beta_{m,1}^n$ | $C_{n-\beta_{m,1}^n}^{m-1}$ | $\beta_{m,2}^n$ | $C_{n-\beta_{m,2}^n}^{m-2}$ | ... |

$\beta_{m,1}^n - \alpha_{m,1}^n + 1$

5. We form Table 5 with the dimension $(\beta_{m,1}^n - \alpha_{m,1}^n + 2) \times 2m$ to calculate $J_{m,\eta}^n(i_\eta)$:

Table 5

Calculation of $J_{m,\eta}^n(i_\eta)$

| | | | | | | | |
|------------------|--------------------|------------------|--------------------|-----|------------------|--------------------|---|
| i_η | $J_{m,1}^n(i_1)$ | i_2 | $J_{m,2}^n(i_2)$ | ... | i_m | $J_{m,m}^n(i_m)$ | |
| $\alpha_{m,1}^n$ | Formulas (39)–(42) | $\alpha_{m,2}^n$ | Formulas (39)–(42) | ... | $\alpha_{m,m}^n$ | Formulas (39)–(42) | } |
| ... | Formulas (39)–(42) | ... | Formulas (39)–(42) | ... | ... | Formulas (39)–(42) | |
| $\beta_{m,1}^n$ | Formulas (39)–(42) | $\beta_{m,2}^n$ | Formulas (39)–(42) | ... | $\beta_{m,m}^n$ | Formulas (39)–(42) | |

$\beta_{m,1}^n - \alpha_{m,1}^n + 1$

The calculation of $J_{m,\eta}^n(i_\eta)$ to fill in Table 5 is carried out by using formulas (39), (41).

$$J_{m,1}^n(i_1) := \sum_{i_1=\alpha_{m,1}^n}^{i_1^j} C_{n-i_1}^{m-1} = \begin{cases} J_{m,1}^n(i_1) := C_{n-\alpha_{m,1}^n}^{m-1}, & i_1 = \alpha_{m,1}^n, \\ J_{m,1}^n(i_1) := J_{m,1}^n(i_1 - 1) + C_{n-i_1}^{m-1}, & i_1 \in (\alpha_{m,1}^n, i_1^j]. \end{cases} \quad (39)$$

We calculate j_{\min} using formula (40).

$$(\eta \in (1, m]) \rightarrow ((i_{\eta-1}^j > \alpha_{m,\eta-1}^n) ? (i_\eta := i_{\eta-1}^j, j_{\min} := J_{m,\eta}^n(i_\eta) := J_{m,\eta-1}^n(i_{\eta-1}^j - 1)) : (j_{\min} := 0)), \quad (40)$$

$$(i_\eta \in (i_{\eta-1}^j, i_\eta^j]) \rightarrow J_{m,\eta}^n(i_\eta) := j_{\min} + \sum_{i_\eta=i_{\eta-1}^j+1}^{i_\eta^j} C_{n-i_\eta}^{m-1} = \begin{cases} J_{m,\eta}^n(i_\eta) := j_{\min} + C_{n-i_\eta}^{m-\eta}, & i_\eta = i_{\eta-1}^j + 1, \\ J_{m,\eta}^n(i_\eta) := J_{m,\eta}^n(i_\eta - 1) + C_{n-i_\eta}^{m-\eta}, & i_\eta \in (i_{\eta-1}^j + 1, i_\eta^j]. \end{cases} \quad (41)$$

The final rule for method A_2 :

$$((\eta^* < m) \& (i_{\eta^*+h} := \beta_{m,\eta^*+h}^n, h \in [1, m - \eta^*])) ? (j := J_{m,\eta^*}^n(i_{\eta^*}^j)) : (j := J_{m,m}^n(i_m^j)). \quad (42)$$

Answer: $(j := J_{m,\eta^*}^n(i_{\eta^*}^j)) \vee (j := J_{m,m}^n(i_m^j)).$

7. Comparative evaluation of the performance of algorithms with sequential access to the Boolean of input data and usage of method A_2

We will use the asymptotic estimate $O(f(n))$ of the increased rate in the number of operations $f(n)$ with increasing n to estimate the productivity of the methods. $O(f(n))$ is calculated for the worst case, when the input data of the length n require the maximum time for the execution of the

algorithm. Thus, $O(f(n))$ is a “truncated” estimate of the execution time of the algorithm which shows the asymptotic changes of $f(n)$ when $n \rightarrow \infty$.

The data structure “ m -tuples based on ordinary sets of arbitrary cardinality n ” is a list with sequential access containing 2^n elements. Therefore, when searching for an element at the end of the list (the worst case), the function to calculate the number of operations is as follows:

$$f(n) = 2^n. \quad (43)$$

Accordingly, the estimated time of the algorithm execution “Searching for an element in a list with sequential access” is as follows:

$$O(f(n)) = O(2^n). \quad (44)$$

Thus, the given algorithm refers to algorithms with the exponential time of execution.

Method A_2 allows us to transform a list with sequential access to a list with direct access when using the data structure “ m -tuples based on ordinary sets of arbitrary cardinality n .”

We calculate the estimation of the execution time for method A_2 .

The function to calculate the number of operations for method A_2 is as follows:

$$f_2(n) = f_{t3}(n) + f_{t4}(n) + f_{t5}(n), \quad (45)$$

where $f_{t3}(n)$, $f_{t4}(n)$, $f_{t5}(n)$ – functions to calculate the number of operations when filling in Tables 3, 4 and 5, respectively.

When filling in Table 3, for the worst case ($m=n$) we have:

$$f_{t3}(n) = 3 \cdot n. \quad (46)$$

Accordingly, the time estimation when filling in Table 3:

$$O_{t3}(f_{t3}(n)) = O_{t3}(n). \quad (47)$$

We will calculate the function of the number of operations for an element from Table 4 $f_C(n)$ to calculate $f_{t4}(n)$. According to this, an element from Table 4 in general can be calculated using the following formula:

$$C_n^m = \frac{n!}{m!(n-m)!}, \quad (48)$$

for $m < n/2$ we have:

$$C_n^m = \frac{n!}{m!(n-m)!} = \frac{\overbrace{(n-m+1) \times \dots \times n}^{m \text{ multipliers}}}{m!}, \quad (49)$$

$$f_C(m) = 2 \cdot (m-1) + 1. \quad (50)$$

For $m > n/2$ we have:

$$C_n^m = \frac{n!}{m!(n-m)!} = \frac{\overbrace{(m+1) \times \dots \times n}^{m \text{ multipliers}}}{(n-m)!}, \quad (51)$$

$$f_c(n-m) = 2 \cdot (n-m-1) + 1. \tag{52}$$

Thus, in the worst case for $m=n/2$ we have:

$$f_c(n) = 2 \cdot (n/2 - 1) + 1 = n - 1. \tag{53}$$

Accordingly, the estimated time to calculate an element from Table 4 is as follows:

$$O_C(f_c(n)) = O_C(n). \tag{54}$$

The function to calculate the number of operations when filling in Table 4 is as follows:

$$f_{t4}(n) = m \cdot (n-m+1) \cdot f_c(n). \tag{55}$$

In the worst case for $m=n/2$ we have:

$$f_{t4}(n) = n/2 \cdot (n/2 + 1) \cdot f_c(n) = n^3/4 + n^2/2, \tag{56}$$

$$O_{t4}(f_{t4}(n)) = O_{t4}(n^3). \tag{57}$$

We will calculate the function of the number of operations for an element from Table 5 $f_j(n)$ to calculate $f_{t5}(n)$. In general, the element from Table 5 is calculated using formula (41) and consists of maximum $n-m$ summands:

$$f_j(n) = n - m. \tag{58}$$

For the worst case ($m=1$):

$$f_j(n) = n - 1. \tag{59}$$

Based on (59), the estimated time to calculate an element from Table 5 is as follows:

$$O_J(f_j(n)) = O_J(n). \tag{60}$$

The function to calculate the number of operations when filling in Table 5 is as follows:

$$f_{t5}(n) = m \cdot (n-m+1) \cdot f_j(n). \tag{61}$$

In the worst case for $m=n/2$ we have:

$$f_{t5}(n) = n/2 \cdot (n/2 + 1) \cdot f_j(n) = n^3/4 + n^2/2, \tag{62}$$

$$O_{t5}(f_{t5}(n)) = O_{t5}(n^3). \tag{63}$$

Therefore, the estimated time for method A_2 is as follows:

$$O_2(f_2(n)) = O_2(n^3). \tag{64}$$

Thus, A_2 is an algorithm of cubic execution time.

8. The results to evaluate the performance of algorithms with sequential access to the Boolean of input data and using method A_2

We form a comparative table of changes in the number of operations when n is increasing for the cubic (method A_2) and exponential (sequence access) time of execution (Table 6).

Table 6

The number of operations for methods with cubic and exponential time of execution

| N | n^3 | 2^n | $\Delta=2^n-n^3$ |
|-----|---------------------|------------------------|------------------|
| 1 | 1 | 1 | 0 |
| 5 | 125 | 32 | -93 |
| 10 | 1 000 | 1 024 | 24 |
| 15 | 3.375×10^3 | 3.277×10^4 | $\sim 10^4$ |
| 20 | 8.0×10^3 | 1.049×10^6 | $\sim 10^6$ |
| 25 | 1.563×10^4 | 3.355×10^7 | $\sim 10^7$ |
| 30 | 2.70×10^4 | 1.074×10^9 | $\sim 10^9$ |
| 35 | 4.288×10^4 | 3.436×10^9 | $\sim 10^9$ |
| 40 | 6.40×10^4 | 1.1×10^{12} | $\sim 10^{12}$ |
| 45 | 9.113×10^4 | 3.518×10^{13} | $\sim 10^{13}$ |
| 50 | 1.250×10^5 | 1.126×10^{15} | $\sim 10^{15}$ |

Graphs of functions of increasing numbers of operations $f(n), f_2(n)$ for sequential access to elements of the given data structure and using algorithm A_2 are shown in Fig. 3.

The Speed of Increasing Numbers of Operations for Sequential Access and Direct Access to the Given Data Structure

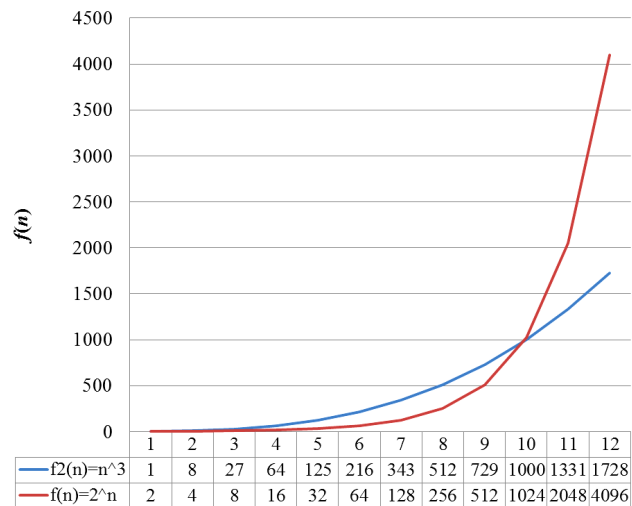


Fig. 3. Graphs of the speed of increasing numbers of operations for methods with sequential and direct access to the given data structure

Taking into account that modern CPUs have a clock frequency more than 3 GHz and an average MAC operation (MAC – Multiplier/Accumulator is an operation called fused multiply – accumulate $Y=Y+A \times B$) is performed in 4 clock cycles, we can set the following execution time for one operation in the given methods:

$$\tau = 4 / (3 \cdot 10^9 \text{ Hz}) = 1,33 \cdot 10^{-9} \text{ s} = 1,33 \text{ ns}. \tag{65}$$

We perform a comparative analysis to estimate the time of execution for the given methods (Table 7).

According to Table 7, we can see that if we work with the data structure “ m -tuples based on ordinary sets of arbitrary cardinality n ” as a list with sequential access, it is quicker to get access to elements of this data structure unlike using method A_2 when $n < 10$.

If $n=10$, the execution time of the methods which work with the data structure as a list with sequential access as well as method A_2 is approximately the same.

If $n \geq 40$, method A_2 of direct access to the elements of the data structure is several orders of magnitude faster than the methods of sequential access to them. Algorithms that implement the methods of sequential access with exponential execution time are generally extensive and refer to the complexity class *NP-hard* (non-deterministic polynomial). The application of method A_2 takes the problem to the complexity class *P* (polynomial), that is, the algorithm to implement A_2 is executed in polynomial time. According to Cobham's thesis, complexity class *P* belongs to the fast-executing complexity class.

Table 7

Estimation of the execution time for methods developed to work with given data structure

| N | $O(n)$, s | $O(n^3)$, s | $O(2^n)$, s |
|----|------------------------|------------------------|---------------------------------------|
| 1 | 1.33×10^{-9} | 1.33×10^{-9} | 1.33×10^{-9} |
| 5 | 6.65×10^{-9} | 1.663×10^{-7} | 4.256×10^{-8} |
| 10 | 1.33×10^{-8} | 1.33×10^{-6} | 1.362×10^{-6} |
| 15 | 1.995×10^{-8} | 4.489×10^{-6} | 4.358×10^{-5} |
| 20 | 2.66×10^{-8} | 1.064×10^{-5} | 1.395×10^{-3} |
| 25 | 3.325×10^{-8} | 2.078×10^{-5} | 4.463×10^{-2} |
| 30 | 3.99×10^{-8} | 3.591×10^{-5} | 1.428 |
| 35 | 4.655×10^{-8} | 5.702×10^{-5} | 45.70 |
| 40 | 5.32×10^{-8} | 8.512×10^{-5} | $1.462 \times 10^3 \approx 25$ min |
| 45 | 5.985×10^{-8} | 1.212×10^{-4} | $4.680 \times 10^4 \approx 13$ h |
| 50 | 6.65×10^{-8} | 1.663×10^{-4} | $1.498 \times 10^6 \approx 17$ days |
| 55 | 7.315×10^{-8} | 2.213×10^{-4} | $4.792 \times 10^7 \approx 18$ months |

9. Discussion of the research results to investigate the effectiveness of the diagnostic method developed by the criterion of the diagnostic time duration

1. We developed a diagnostic mathematical model using the methodology of Expert Systems to study the processes of functional diagnostics for the automated system operability in order to minimize the time needed to establish the diagnosis. The distinctive features of this model are as follows:

- complete diagnostics of the entire integrated complex TCO – SCADA. This can be achieved by processing a complete set of input data in the diagnostic model – diagnostic codes generated by SCADA at all levels of the hierarchy (hardware platform – operating system – software development and execution environment – SCADA configuration software – run-time SCADA software – software to control and monitor TCO);

- minimization of the time for the diagnosis search formation space when working with a conflicting set of input

data in the model developed – system diagnostic codes. Minimization of the diagnostic time is achieved due to the fact that we apply methods to work with the data structure “ m -tuples based on ordinary sets of arbitrary cardinality n ” instead of methods with sequential access to the elements of the Boolean of input data.

2. Investigation of the patterns for the diagnosis search formation space for the Boolean of input data showed that the access time to the elements of the Boolean of input data increases depending on the number of input data n :

- using the exponential functional dependence 2^n for sequential access to the elements of the data structure;
- using the cubic functional dependence n^3 for direct access to the elements of the data structure.

3. A distinctive feature of the method developed to form the diagnosis search space for the integrated complex TCO – SCADA is the minimization of the access time to the elements of the Boolean of input data $2^{X(\Delta t)}$ as well as the minimization of the time to obtain the complement to the element of Boolean $2^{X(\Delta t)}$ to a subset of the ordered basis set. This allows us to quickly determine conflicting groups of input data.

4. The effectiveness analysis of the diagnostic method developed by the criterion of the diagnosis search duration showed that the algorithm realizing this method allows us to segue from the tasks of the *NP-hard* complexity class in general to the tasks of the *P* complexity class, unlike the diagnostic method with sequential access to the elements. This approach allows us to minimize the time to establish the diagnosis to real time. This has a significant impact when processing a set of input data of arbitrary $n \geq 40$ (Table 7).

10. Conclusions

1. The method developed for the diagnosis search formation space changes the functional dependency of the execution time estimation for the algorithm in accordance with the number of its input data n from exponential $O(2^n)$ to cubic $O(n^3)$.

2. The effectiveness of the method developed is confirmed by the fact that the formation time of the diagnosis search space is minimized to real-time, which is especially noticeable for the number of input data $n \geq 40$.

3. The method developed for automatic self-diagnostics for the integrated complex TCO – SCADA allows creating methods and algorithms for SCADA automatic self-recovery after reversible failures in real time.

4. The method developed to diagnose the operability of an integrated complex TCO – SCADA is a prerequisite for creating a methodology and a mathematical basis for the design and implementation of maintenance-free hardware-software complexes.

References

1. Ponomarev, O. P. Naladka i ekspluatatsiya sredstv avtomatizatsyi. SCADA-sistemy. Promyshlennyye shiny i interfeysy. Obshchie svedeniya o programmiruemykh logicheskikh kontrollerah i odnoplattnykh komp'yuterah [Text]; ucheb. pos. / O. P. Ponomarev. – Kaliningrad: Izd-vo in-ta «KVSHU», 2006. – 80 p.
2. Shopin, A. G. Evolyutsiya SCADA i informatsionnykh sistem proizvodstva [Text] / A. G. Shopin, I. V. Zanin // Avtomatizatsiya v promyshlennosti. – 2012. – Issue 1. – P. 18–21.
3. Berry, B. SCADA Tutorial: A Fast Introduction to SCADA Fundamentals and Implementation [Text] / B. Berry. – DPS Telecom, USA, 2011. – 12 p.

4. Goryainov, A. N. Opredelenie effektivnosti sistem diagnostirovaniya v teorii transportnoy diagnostiki [Text] / A. N. Goryainov // Vestnik NTU «KhPI». – 2012. – Issue 1. – P. 64–70.
5. Charbonnier, S. Analysis of fault diagnosability from SCADA alarms signatures using relevance indices [Text] / S. Charbonnier, N. Bouchair, P. Gayet // 2014 IEEE International Conference on Systems, Man, and Cybernetics (SMC). – 2014. doi: 10.1109/smc.2014.6974342
6. Munoz, M. A tool for the performance evaluation and failure detection of Amareleja PV plant (ACCIONA) from SCADA [Text] / M. Munoz, I. de la Parra, M. Garcia, J. Marcos, M. Perez // 2015 17th European Conference on Power Electronics and Applications (EPE'15 ECCE-Europe). – 2015. doi: 10.1109/epe.2015.7311722
7. Wang, X. Research on Transformer Fault Diagnosis based on Multi-source Information Fusion [Text] / X. Wang, K. Wu, Y. Xu // International Journal of Control and Automation. – 2014. – Vol. 7, Issue 2. – P. 197–208. doi: 10.14257/ijca.2014.7.2.19
8. Wang, K.-S. SCADA data based condition monitoring of wind turbines [Text] / K.-S. Wang, V. S. Sharma, Z.-Y. Zhang // Advances in Manufacturing. – 2014. – Vol. 2, Issue 1. – P. 61–69. doi: 10.1007/s40436-014-0067-0
9. Windmann, S. Efficient fault detection for industrial automation processes with observable process variables [Text] / S. Windmann, O. Niggemann // 2015 IEEE 13th International Conference on Industrial Informatics (INDIN). – 2015. doi: 10.1109/indin.2015.7281721
10. MacGregor, J. Monitoring, fault diagnosis, fault-tolerant control and optimization: Data driven methods [Text] / J. MacGregor, A. Cinar // Computers & Chemical Engineering. – 2012. – Vol. 47. – P. 111–120. doi: 10.1016/j.compchemeng.2012.06.017
11. Dzhekson, P. Vvedenie v ekspertnye sistemy [Text]: ucheb. pos. / P. Dzhekson. – Moscow: Vil'yams, 2001. – 624 p.
12. Syrotkina, E. I. Formirovanie ob'ektno-klassifikatsionnoy modeli diagnostiki raboty SCADA sistemy [Text] / E. I. Syrotkina, M. A. Alekseev // Problemy nedropol'zovaniya. – Sankt-Peterburg, 2013. – P. 256–258.
13. Syrotkina, O. Formation of the classification space of the expert system knowledge base for SCADA failure diagnostics [Text] / O. Syrotkina // Power Engineering, Control and Information Technologies in Geotechnical Systems. – 2015. – P. 179–184. doi: 10.1201/b18475-25
14. Ponomarev, V. F. Matematicheskaya logika [Text]: ucheb. pos. / V. F. Ponomarev. – 2-e izd., ispr. i dop. – Kaliningrad: izd-vo KGTU, 2005. – 201 p.
15. Syrotkina, O. The application of specialized data structures for SCADA diagnostics [Text] / O. Syrotkina // Sistemnye tekhnologii. – 2015. – Issue 4. – P. 72–81.