UDC 681.516

DOI: 10.29202/nvngu/2019-3/18

O. Syrotkina[1], Cand. Sc. (Tech.),
orcid.org/0000-0002-4069-6984,
M. Alekseyev[1], Dr. Sc. (Tech.), Prof.,
orcid.org/0000-0001-8726-7469,
V. Asotskyi[2], Cand. Sc. (Psychol.),
orcid.org/0000-0001-5403-3156,
I. Udovyk[1], Cand. Sc. (Tech.), Assoc. Prof.,
orcid.org/0000-0002-5190-841X

1 – Dnipro University of Technology, Dnipro, Ukraine, e mail: syrotkina.o.i@nmu.one
2 – National University of Civil Defence of Ukraine, Kharkiv, Ukraine, e-mail: asotskiy@nuczu.edu.ua

ANALYSIS OF HOW THE PROPERTIES OF STRUCTURED DATA CAN INFLUENCE THE WAY THESE DATA ARE PROCESSED

Purpose. The purpose of the article is to develop mathematical methods for processing "big data". This is based on the system analysis of properties for their structural organization. These methods allow us to optimize the basic characteristics of "big data". This includes increasing the search speed to process large volumes of fast incoming data while preserving their relevance.

Methodology. We suggested mathematical methods to work with a data structure "*m*-tuples based on ordered sets of arbitrary cardinality (OSAC)". We determined pairwise combinations of Boolean elements as operands of the operations investigated. The foregoing is based on the analysis of the data structure properties. We also calculated the dynamics of changes in the constituent pairwise combinations of the Boolean elements depending on the basis set cardinality for different groups of the given data structure.

Findings. We estimated the time needed to execute methods of working with the OSAC data structure as functional dependencies of the amount of data $O(f(n))$. We also determined the component of combinations for Boolean elements. For these elements, the execution of algorithms that implement the operation investigated is not required as the desired result is defined in the data structure property.

Originality. We further developed a mathematical method which allows us to forecast the result of performing certain operations on elements of ordered data structure. This takes into account the position of the elements in the structure without using the computational algorithm. For the first time, we obtained an analytical dependency to determine the component number for Boolean elements of length m_2. This includes an element represented by a tuple of smaller length m_1 in relation to the total number of Boolean elements of length m_2. For the first time we also obtained an analytical dependency to determine the minimum maxima of the functional dependency described above.

Practical value. The results obtained in this paper can be used to minimize the time and computational resources needed to process "big data" represented by *m*-tuples based on OSAC.

Keywords: *"big data", data organization structure, m-tuples, Boolean graph*

Introduction. At the present time, modern SCADA (Supervisory Control and Data Acquisition) systems are commonly distributed in the field of industrial automation. These systems are widely used throughout various industries and comprise multi-tasking with multi-user hardware and software [1]. There is an urgent need to develop methods for reliable and timely analysis of large flows of diagnostic information generated by the system in the event of a failure. This is necessary to diagnose the performance of these systems in real time [2].

The task of SCADA operation diagnostics refers to one of the most rapidly developing areas of modern information technology. Specifically, this is the area of creating and implementing a variety of methods and techniques as well as tools for processing, storing, analyzing and managing "big data" [3].

The main characteristics of "big data" include the volume of data, the velocity of their processing, and the diversity of data types [4]. Therefore, the relevant task is to maintain not only quantitative, but also the qualitative characteristics of "big data" when creating methods for their processing and analysis. The characteristics of "big data" in the methods created should meet the requirements for information technology solutions in this area.

Literature review. It is known that in order to perform automatic processing and analysis of "big data" in the conditions of temporal and informational limitations, three main methodologies are employed in the applied methods. These are mathematical, linguistic and heuristic.

In the mathematical approach [5], classification rules are derived within the framework of a certain mathematical formula, which also determines the causation and regularity of events.

The linguistic approach [6] is used for objects containing complex hierarchical structures where it is difficult to identify specific elements. In these cases, a set of rules is introduced which must be based on the system forming a "dictionary" of attributes while determining the boundaries of classes.

The heuristic approach [7] is based on intuition and many years of experience. It is used when working with hard-to-formalize knowledge for highly specialized tasks. The heuristic approach does not have strict formal justification. However, in most practical situations it provides an acceptable solution to the problem.

The methods which apply to the classification approach also include graphical and analytical methods for finding solutions in the space of their states. Examples of these diagnostic methods comprise the Fault Tree Analysis (FTA) and the Event Tree Analysis (ETA).

Fault Tree Analysis (FTA) [8] is a graphical and analytical method which is formed of a multi-level graphical and analytical structure. Causal relationships and chains of events (impacts) which lead to system failures are identified on the basis of this structure.

Event Tree Analysis (ETA) [9] is a graphical and analytical method for describing possible scenarios of events proceeding from the main event (i.e. emergency, failure). In addition, each element of the system represented by a node of the event tree can be in one of two states: operable or inoperable.

In work [8], the authors conducted an investigation into why the protective valves for a pressure vessel often failed. As a result, a reliability model for their system was developed. Reliability is formalized with a dynamic failure tree (DFT). The main disadvantage of models based on DFT is the high complexity of their structure which increases exponentially with every increase in the number of elements added to it. To solve this problem, the article proposes an approach that is based on the combination of DFT and Markov reliability models. In Markov model, the splitting of the space states was performed on the basis of tensor expressions. It takes into account arbitrary separation and memorization of the history of the operating time of system elements under load by using fictitious phases.

In work [9], the authors suggested a mathematical model to evaluate the reliability of the module as it pertains to keeping poultry. The main objective of the model is to maintain the microclimate parameters of the module within specified limits. They developed a homogeneous Markov model containing hundreds of states and transitions. This model is based on the structure of the failure-formed tree analysis and the reliability parameters of the module elements. The resulting graph was described using the Chapman-Kolmogorov system containing hundreds of differential equations where each equation describes one of the states of the system. They used the Dormand-Prince method to solve the system of equations with constant coefficients. It is included in the MATLAB mathematical package. Since all system states are absorbing, no computational problems were found. This model avoided the influence of "unreachable" states, which increases the accuracy of the results obtained. The authors performed an analysis and determined the probabilistic characteristics for the reasons of the module's inoperability based on the model developed.

Work [10] presents a method for reducing the amount of energy expended in coal mining using the help of the technological unit within the mining industry. The studies were conducted using a simulation model based on traditional methods for calculating the power used by the mining equipment. It also included a mathematical description of the processes occurring during energy conversion within the electric drive. All the results were obtained during the testing of mining and geological parameters of the mining operation. For the development of an energy-efficient algorithm for controlling the aggregate, an analytical relationship was derived between the maximum relative deviation of the selected numerical criterion from the average value and the volume of circulating ore. At the same time, the system periodically checks the value of the critical ore feed rate to ensure that the unit is operating at maximum capacity.

Unsolved aspects of the problem. We can consider a scheme when the flow of diagnostic information from SCADA is inputted into an expert diagnostic system (EDS) in order to establish a timely and reliable diagnosis of SCADA operability. The EDS has a database (DB) containing a set of SCADA software processes, a set of diagnostic codes (DC) generated by these processes, and a set of SCADA failures (SF). The knowledge base (KB) of EDS contains a set of rules and algorithms to establish the diagnosis based on a set of DCs. Different combinations of DCs coming to the EDS input correspond to different SCADA failures. It is necessary to create a method for processing and analyzing streams of diagnostic information by the expert system including all the possible (for a given expert system configuration) diagnostic code combinations which are in the input data set. The foregoing allows a reliable and timely solution to be found while minimizing computing resources when processing data in real time.

Purpose. The flow of diagnostic information is a complex data structure. It includes, for example, the following data: the moment of failure detection, the network address of the backbone node, the system diagnostic code, and the identifier of the software process that generated the diagnostic code, etc.

We define an element of diagnostic information flow which goes from SCADA to the EDS as a structure (c, q, p). This structure contains the diagnostic code c generated by SCADA as a result of passing/not passing SCADA execution process p through its control point q.

We define $X_{\Delta t}$ as the flow of diagnostic information from SCADA to the EDS over the time interval Δt. We can represent it as a certain set of SCADA diagnostic codes C_x generated at the control points Q_y by SCADA execution processes P_z. Thus, the EDS input data will be represented by a template basis set X which will be instantiated by the data structure (c, q, p) and initialized by the set $(C_x, Q_y, P_z)_{\Delta t}$

$$X_{\Delta t} = X\langle c, q, p\rangle (C_x, Q_y, P_z)_{\Delta t}.$$

Suppose we have a diagnostic code at a given point of time, t_i that belongs to the time interval Δt. This code is generated in a control point (CT) and belongs to some SCADA process.

$$x_{t_i} = (c,k,p)_{t_i} \in (C_x, K_y, P_z)_{\Delta t}.$$

Then we have

$$\begin{cases} X_{\Delta t} = \left\{ x_{t_1}, \ldots, x_{t_i}, \ldots, x_{t_E} \right\} \\ \Delta t = t_E - t_1 \\ \left| X_{\Delta t} \right| = n \end{cases}.$$

Imagine the structure of the EDS in the form

$$EDS = \langle DB, KB, IM \rangle,$$

where IM is an inference machine.

Let the database contain a set of DCs generated by SCADA when passing through its process control points, so that

$$(C_x, O_y, P_z)_{\Delta t} \in (C, Q, P).$$

Then the *EDS* database will include the template basis set X, instantiated by the data structure (c, q, p) and initialized by the set (C, Q, P)

$$X_{DB} = X\langle c, q, p\rangle(C, K, P).$$

Wherein

$$X_{\Delta t} \subset X_{DB}.$$

We denote $x_{\Delta Bi} = (c, q, p)_i \in (C, Q, P)$. Then

$$X_{DB} = \{x_{DB1}, \ldots, x_{DBi}, \ldots, x_{DBn(X_{DB})}\}.$$

Suppose the *EDS* database also contains many types of SCADA F failures. Suppose the *KB* of our expert system contains many G rules, algorithms and strategies that allow us to construct variants of the component structures for the database.

$$F \to G(X_{DB}).$$

In order to establish a diagnosis for SCADA operability using the *EDS* by the flow of diagnostic information for the time interval Δt, it is necessary to determine $F_{\Delta t} \to G(X_{\Delta t})$. We will do it with the help of the *EDS* inference machine in order to minimize time and computing resources when processing our data.

We have

$$X_{DB} = \{x_1, \ldots, x_i, \ldots, x_{n(X_{DB})}\};$$

$$F = \{f_1, \ldots, f_j, \ldots, f_{n(F)}\};$$

$$G = \{g_1, \ldots, g_k, \ldots, g_{n(G)}\};$$

$$F \to G(X_{DB}) \Rightarrow$$

$$\Rightarrow \begin{cases} f_j = g_k\left(x_1^{g_k}, \ldots, x_i^{g_k}, \ldots, x_{n(f_j)}^{g_k}\right) = g_k(X_{g_k}) \\ X_{DB} = \bigcup\limits_k X_{g_k} \end{cases}.$$

Required to find

$$\begin{cases} F_{\Delta t} \to G(X_{\Delta t}) \\ t_{proc.} \to t_{\min} \\ V_{st} \to V_{\min} \\ R_{calc} \to R_{\min} \end{cases},$$

where $F_{\Delta t}$ is diagnosis by the flow of diagnostic information $X_{\Delta t}$; t_{proc} is diagnostic information processing time $X_{\Delta t}$ to establish the diagnosis; V_{st} is the amount of data storage for storing and processing diagnostic information $X_{\Delta t}$; R_{calc} is the amount of computing resources involved in processing the data.

To complete the task, we must determine that

$$2^{X_{\Delta t}} \cap X_G \neq \varnothing,$$

where X_G is a set of diagnostic code combinations in the *EDS* database; $2^{X_{\Delta t}}$ is a set of input data combinations of the *EDS* for the time interval Δt.

This article discusses some of the properties and methods of working with the data structure "m-tuples based on ordered sets of arbitrary cardinality (OSAC)" in order to minimize time and computational resources when processing "big data" represented by the flow of diagnostic information from SCADA.

Methods. The description of the basic terms and definitions, as well as some properties and mathematical methods of working with the data structure "m-tuples based on OSAC" are given in [11].

In this paper, we consider the possibility of minimizing the time and computational resources in data processing for the methods $A_7: i^n_{m,j} = i^n_{m_1,j_1} \cup i^n_{m_2,j_2}$ and $A_8: y^n_{m,j} = y^n_{m_1,j_1} \cup i^n_{m_2,j_2}$. These methods implement the union of Boolean elements 2^I and 2^X as part of the given data structure.

We define m-tuples as ordered, ascending subsets of the basis sets

$$i^n_{m_1,j_1} = \left\{i_1^{j_1}, \ldots, i_{\eta_1}^{j_1}, \ldots, i_{m_1}^{j_1}\right\} \subseteq I;$$

$$i^n_{m_2,j_2} = \left\{i_1^{j_2}, \ldots, i_{\eta_2}^{j_2}, \ldots, i_{m_2}^{j_2}\right\} \subseteq I;$$

$$y^n_{m_1,j_1} = \left\{x_{i_1}^{j_1}, \ldots, x_{i_{\eta_1}}^{j_1}, \ldots, x_{i_{m_1}}^{j_1}\right\} \subseteq X;$$

$$y^n_{m_2,j_2} = \left\{x_{i_1}^{j_2}, \ldots, x_{i_{\eta_2}}^{j_2}, \ldots, x_{i_{m_2}}^{j_2}\right\} \subseteq X.$$

Therefore, under the union of the Boolean elements, we accept the logical operation of the union of the sets represented by the operands according to the corresponding basis sets.

$$i^n_{m,j} = \left\{i_1^{j_1}, \ldots, i_{\eta_1}^{j_1}, \ldots, i_{m_1}^{j_1}\right\} \cup \left\{i_1^{j_2}, \ldots, i_{\eta_2}^{j_2}, \ldots, i_{m_2}^{j_2}\right\};$$

$$y^n_{m,j} = \left\{x_{i_1}^{j_1}, \ldots, x_{i_{\eta_1}}^{j_1}, \ldots, x_{i_{m_1}}^{j_1}\right\} \cup;$$

$$\cup \left\{x_{i_1}^{j_2}, \ldots, x_{i_{\eta_2}}^{j_2}, \ldots, x_{i_{m_2}}^{j_2}\right\}.$$

The result of this operation is a tuple consisting of the ordered values of the elements that were part of at least one of the operands.

Define formal rules for the execution of a method

$$A_7: i^n_{m,j} = i^n_{m_1,j_1} \cup i^n_{m_2,j_2}.$$

We have

$$n, m_1, m_2, j_1, j_2;$$

$$i^n_{m_1,j_1} = \left(i_1^{j_1}, \ldots, i_{\eta_1}^{j_1}, \ldots, i_{m_1}^{j_1}\right);$$

$$i^n_{m_2,j_2} = \left(i_1^{j_2}, \ldots, i_{\eta_2}^{j_2}, \ldots, i_{m_2}^{j_2}\right).$$

Required to find: $m, j, i^n_{m,j} = i^n_{m_1,j_1} \cup i^n_{m_2,j_2}$.

Decision:

1. Perform a validation check of the input data:

$$n \geq 1; \quad 1 \leq m_1 \leq n; \quad 1 \leq m_2 \leq n;$$

$$1 \leq j_1 \leq k^n_{m_1}; \quad 1 \leq j_2 \leq k^n_{m_2}.$$

2. Form Table 1 with dimension $(\max(m_1, m_2) + 2) \times 6$ to calculate the parameters $\alpha^n_{m_1,\eta_1}$, $\beta^n_{m_1,\eta_1}$ and $\alpha^n_{m_2,\eta_2}$, $\beta^n_{m_2,\eta_2}$.

3. Perform input validation.

$$i^n_{m1,j1} = \left(i^{j1}_1, \ldots, i^{j1}_{\eta 1}, \ldots, i^{j1}_{m1}\right),$$

and

$$i^n_{m2,j2} = \left(i^{j2}_1, \ldots, i^{j2}_{\eta 2}, \ldots, i^{j2}_{m2}\right).$$

Ensure that $i^n_{m_1,j_1}$ and $i^n_{m_2,j_2}$ are set correctly

$$\forall i^{j_1}_{\eta_1} \in [1, m_1] \to \left(\alpha^n_{m_1,\eta_1} \leq i^{j_1}_{\eta_1} \leq \beta^n_{m_1,\eta_1}\right);$$

$$\forall i^{j_1}_{\eta_1} \in (1, m_1] \to \left(i^{j_1}_{\eta_1} - i^{j_1}_{\eta_1 - 1} \geq 1\right);$$

$$\forall i^{j_2}_{\eta_2} \in [1, m_2] \to \left(\alpha^n_{m_2,\eta_2} \leq i^{j_2}_{\eta_2} \leq \beta^n_{m_2,\eta_2}\right);$$

$$\forall i^{j_2}_{\eta_2} \in (1, m_2] \to \left(i^{j_2}_{\eta_2} - i^{j_2}_{\eta_2 - 1} \geq 1\right).$$

4. Perform an algorithm for combining two Boolean elements via the basis set

$$\eta := 1; \quad \eta_1 := 1; \quad \eta_2 := 1;$$

$$\Big[\left((\eta \leq \min(m_1 + m_2, n)) \wedge (\eta_1 \leq m_1) \wedge (\eta_2 \leq m_2)\right)?$$

$$(((i^{j_1}_{\eta_1} > i^{j_2}_{\eta_2})?(i_\eta := i^{j_2}_{\eta_2}, \eta_2 ++):$$

$$\left(i_\eta := i^{j_1}_{\eta_1}, \quad \eta_1 ++, ((i^{j_1}_{\eta_1} = i^{j2}_{\eta_2})?(\eta_2 ++)))\right), \quad \eta ++\Big],$$

$$\Big[\left((\eta \leq \min(m_1 + m_2, n)) \wedge (\eta_1 \leq m_1) \wedge (\eta_2 \leq m_2)\right)?$$

$$\left(i_\eta := i^{j_1}_{\eta_1}, \quad \eta ++, \quad \eta_1 ++\right)\Big],$$

$$\Big[\left((\eta \leq \min(m_1 + m_2, n)) \wedge (\eta_1 > m_1) \wedge (\eta_2 \leq m_2)\right)?$$

$$\left(i_\eta := i^{j_2}_{\eta_2}, \quad \eta ++, \quad \eta_2 ++\right)\Big],$$

$$m := \eta - 1,$$

$$j := A_2(i_1, \ldots, i_\eta, \ldots, i_m).$$

Table 1

Scopes of index definitions i_η

$i^n_{m_1,j_1}$			$i^n_{m_2,j_2}$		
η_1	$\alpha^n_{m_1,\eta_1}$	$\beta^n_{m_1,\eta_1}$	η_2	$\alpha^n_{m_2,\eta_2}$	$\beta^n_{m_2,\eta_2}$
1			1		
...	η_1	$n - m_1 + \eta_1$	...	η_2	$n - m_2 + \eta_2$
m_1			m_2		

Answer: as a result of the method execution A_7 using method A_2 [12] we derived m-tuple $i^n_{m,j} = (i_1, \ldots, i_\eta, \ldots, i_m)$.

We define formal rules for the method execution $A_8 : y^n_{m,j} = y^n_{m_1,j_1} \cup y^n_{m_2,j_2}$.

We have

$$n, m_1, m_2, j_1, j_2;$$

$$y^n_{m_1,j_1} = \left\{x^{j_1}_{i_1}, \ldots, x^{j_1}_{i_{\eta_1}}, \ldots, x^{j_1}_{i_{m_1}}\right\};$$

$$y^n_{m_2,j_2} = \left\{x^{j_2}_{i_1}, \ldots, x^{j_2}_{i_{\eta_2}}, \ldots, x^{j_2}_{i_{m_2}}\right\}.$$

Required to find: $m, j, y^n_{m,j} = y^n_{m_1,j_1} \cup y^n_{m_2,j_2}$.

Decision:

1. Execute point 1 described in method A_7.

2. In addition to point 1 from method A_7 we will check the validity of the input data

$$\forall \eta_1 \in [1, m_1) \to x^{j_1}_{i_{\eta_1}} < x^{j_1}_{i_{\eta_1 + 1}};$$

$$\forall \eta_2 \in [1, m_2) \to x^{j_2}_{i_{\eta_2}} < x^{j_2}_{i_{\eta_2 + 1}}.$$

3. Form m-tuples $i^n_{m_1,j_1}$ and $i^n_{m_2,j_2}$ using method A_1 [11]

$$i^n_{m_1,j_1} = A_1\left(j_1\right);$$

$$i^n_{m_2,j_2} = A_1\left(j_2\right).$$

4. Perform the method described above

$$A_7 : i^n_{m,j} = i^n_{m_1,j_1} \cup i^n_{m_2,j_2}.$$

5. We find corresponding tuple $y^n_{m,j}$ for the resulting m-tuple $i^n_{m,j} = (i_1, \ldots, i_\eta, \ldots, i_m)$.

Answer: as a result of the execution of method A_8 we derived an m-tuple

$$y^n_{m,j} = (x_{i_1}, \ldots, x_{i_\eta}, \ldots, x_{i_m}).$$

When calculating estimates of the execution time for methods A_7 and A_8 we obtained the following results:

- obtaining the resulting m-tuple $(i_1, \ldots, i_\eta, \ldots, i_m)$ using method A_7 without determining its location in the given data structure. It corresponds to the linear execution time algorithm

$$O'_7(f'_7(n)) = O'_7(n);$$

- obtaining the resulting m-tuple using method A_7 by determining its location in the given data structure $i^n_{m,j}$. It corresponds to the algorithm of cubic execution time

$$O_7(f_7(n)) = O_7(n^3);$$

- obtaining the resulting m-tuple $y^n_{m,j}$ using method A_8 corresponds to the algorithm of cubic execution time

$$O_8(f_8(n)) = O_8(n^3).$$

We can define some properties of intersections / unions of m-tuples.

Property 1. If at least one of the operands of intersection of Boolean elements in a basis set with cardinality n is an n-tuple, then their intersection will be the second operand

$$((m_1 = n) \wedge (m_2 \neq n))?(m := m_2, j := j_2):$$
$$(((m_1 \neq n) \wedge (m_2 = n))?(m := m_1, j := j_1) : NOP),$$

where NOP is "no operation".

If at least one of the operands of the union of Boolean elements in a basis set with cardinality n is an n-tuple, then their union will be an n-tuple

$$((m_1 = n) \vee (m_2 = n))?(m := n, j := 1).$$

Property 2. The intersection or union of a Boolean element in a basis set with itself is the same m-tuple

$$((m_1 = m_2) \wedge (j_1 = j_2))?(m := m_1, j := j_1).$$

Property 3. The intersection of two different $(n-1)$-tuples in the basis set with cardinality n, will be $(n-2)$-tuple

$$((m_1 = n-1) \wedge (m_2 = n-1) \wedge (j_1 \neq j_2))?(m := n-2).$$

Combining two different $(n-1)$-tuples in the basis set with cardinality n, will be an n-tuple.

$$((m_1 = n-1) \wedge (m_2 = n-1) \wedge (j_1 \neq j_2))?(m := n, j := 1).$$

Property 4. If one of the operands of the intersection of Boolean elements in the basis set with cardinality n is the complement of the other operand to the basis set, then the result of the operation is $\varnothing$.

$$((m_2 = n - m_1) \wedge (j_2 = k^n_{m_1} - j_1 + 1))?(\varnothing).$$

If one of the operands of the union of Boolean elements in the basis set with cardinality n is the complement of the other operand to the basis set, then the result of the operation is an n-tuple.

$$((m_2 = n - m_1) \wedge (j_2 = k^n_{m_1} - j_1 + 1))?(m := n, j := 1).$$

Property 5. If one of the intersection operands is a subset of the other one, then the result of their intersection will be the first operand.

$$(y^n_{m_1,j_1} \subset y^n_{m_2,j_2})? y^n_{m,j} = y^n_{m_1,j_1}.$$

If one of the union operands is a subset of the other one, then the result of their union will be the second operand.

$$(y^n_{m_1,j_1} \subset y^n_{m_2,j_2})? y^n_{m,j} = y^n_{m_2,j_2}.$$

We developed a software application called "Cortege" in the Borland C++ Builder environment to carry out experimental studies. In this software application we created a template class.

template ⟨Class T⟩ class Cortege.

In this class we implemented a data structure called "m-tuples based on OSAC" described in our paper as well as the methods we use to work with it.

In order to determine (specialize) the Cortege template class, it must be instantiated by the data type we examine.

Results. Consider an example of instantiating the Cortege class with a character data type of the basis set with cardinality $n = 3$.

$$X = \{a, b, c\}.$$

We define a non-empty subset of the Boolean $2^X \backslash \varnothing$ as a set of m-tuples in Table 2.

For our basis set X with cardinality $n = 3$, we have the cardinality of a non-empty subset of the Boolean

$$|2X \backslash \varnothing| = 2^n - 1 = 7.$$

The number of possible combinations of Boolean elements N_Σ, as method operands A_7 and A_8, consists of the sum of two components:

1) the number of the Boolean element associations with itself, i. e. the number of combinations of $2^n - 1$ by 1;

2) the number of associations of various elements of the Boolean, i.e. the number of combinations of $2^n - 1$ by 2.

$$N_\Sigma = \begin{pmatrix} 2^n - 1 \\ 1 \end{pmatrix} + \begin{pmatrix} 2^n - 1 \\ 2 \end{pmatrix} = 7 + 21 = 28.$$

The number of possible combinations of Boolean elements satisfying property 1

$$N_1 = 2^n - 1 = 7.$$

The number of possible combinations of Boolean elements satisfying property 2

$$N_2 = 2^n - 1 = 7.$$

However, since merging element $y^3_{3,1}$ with itself satisfies both the first and second properties, then we will take into account this combination in N_1. Then we believe that

$$N_2 = 2^n - 2 = 6.\ N_2 = 2^n - 2 = 6.$$

The number of possible combinations of Boolean elements satisfying property 3

$$N_3 = \begin{pmatrix} k^n_{n-1} \\ 2 \end{pmatrix} = \begin{pmatrix} 3 \\ 2 \end{pmatrix} = 3.$$

Table 2

Definition of m-tuples

№	m	j	$y^n_{m,j}$	$(x_{i_1}, \ldots, x_{i_m})$	Y^n_m
1	1	1	$y^3_{1,1}$	a	Y^3_1
2		2	$y^3_{1,2}$	b	
3		3	$y^3_{1,3}$	c	
4	2	1	$y^3_{2,1}$	(a, b)	Y^3_2
5		2	$y^3_{2,2}$	(a, c)	
6		3	$y^3_{2,3}$	(b, c)	
7	3	1	$y^3_{3,1}$	(a, b, c)	Y^3_3

The number of possible combinations of Boolean elements satisfying property 4

$$N_4 = (2^n - 2)/2 = 3.$$

The total number of combinations of elements that satisfy the above properties of associations of Boolean elements

$$\sum_{i=1}^{4} N_i = (2^n - 1) + (2^n - 2) + \binom{k_{n-1}^n}{2} + (2^n - 2)/2;$$

$$\sum_{i=1}^{4} N_i = 2.5 * 2^n + \binom{n}{2} - 4 = 19;$$

$$\sum_{i=1}^{4} N_i \Big/ N_\Sigma \times 100\,\% \approx 68\,\%.$$

Thus, for our basis set X with cardinality $n = 3$, we find that 68 % of the combinations of Boolean elements satisfy the first four intersection/union properties of the elements described above and do not require the execution of algorithms implementing intersection/union operations since the desired result is already obtained from these properties.

We calculate the number of possible combinations of Boolean elements satisfying property 5 without taking into account combinations with n-tuples (Property 1) which are defined in N_1

$$N_5 := 0; \quad m : 1,$$

$$[(1 \le m \le n - 1)] ?$$

$$\left(k_m^n = \frac{n!}{(n-m)! \cdot m!}, \quad N_5 += k_m^n * (2^{n-m} - 2), \quad m++ \right].$$

So for $n = 3$, $N_5 = 6$.

$$\sum_{i=1}^{5} N_i = 25;$$

$$\sum_{i=1}^{5} N_i \Big/ N_\Sigma \times 100\,\% \approx 89\,\%;$$

$$\Delta = 89\,\% - 68\,\% = 21\,\%.$$

For $n = 3$, we find that using Property 5 for intersection / union of Boolean elements allows a 21 % increase in the component of combinations of Boolean elements. This does not require the execution of algorithms that implement the operation. The desired result is already obtained from the property.

Table 3 was created to determine the number of combinations of Boolean elements. These combinations should satisfy the above described properties of intersection / union of Boolean elements depending on our basis set with cardinality n.

The graph depicted in Fig. 1 is of the components from the total number of combinations of Boolean elements with the previously calculated results achieved without performing methods A_7 and A_8. This graph also shows the results of the intersection/union operations obtained on the basis of the properties of the elements.

We can analyze some additional properties of the given data structure called "m-tuples based on OSAC" which follow from Property 5.

To do this, we represent our data structure in the form of a graph where each vertex of the graph corresponds to element $y_{m,j}^n$ of the data structure. This graph is shown in Fig. 2.

Numerical characteristics that determine the graph and its vertices are shown in Table 4.

Table 3

Components of combinations of Boolean elements satisfying the intersection / union of properties

n	N_Σ	$\sum_{i=1}^{4} N_i$	$\%_1$	$\sum_{i=1}^{5} N_i$	$\%_2$	Δ
1	2	3	4 = 3/2, %	5	6 = 5/2, %	7 = 6 – 4
3	28	19	68	6	89	21
4	120	42	35	36	65	30
5	496	86	17	150	47.6	30.6
6	1 989	171	8 6	540	35.7	27.1
7	8 128	337	4 1	1 806	26.4	22.3
8	32 640	664	2	5 782	19.7	17.7
9	130 816	1 312	1	18 150	14.9	13.9
10	523 776	2 601	0 5	55 980	11.2	10.7
11	2 096 128	5 171	0 25	171 006	8.4	8.15
12	8 386 560	10 266	0 12	529 422	6.3	6.18
13	33 550 336	20 554	0 06	1 590 304	4.74	4.68
14	134 209 536	41 047	0 03	4 774 867	3.56	3.53
15	536 854 528	82 021	0 015	14 332 627	2.67	2.655

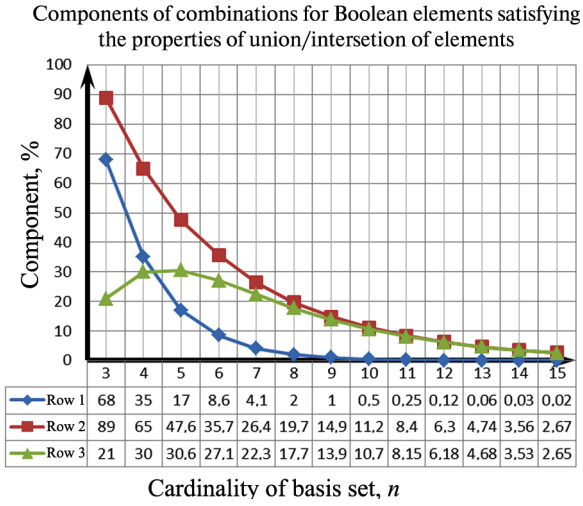

Fig. 1. Graph of constituent combinations of Boolean elements satisfying the properties of intersection / union of elements

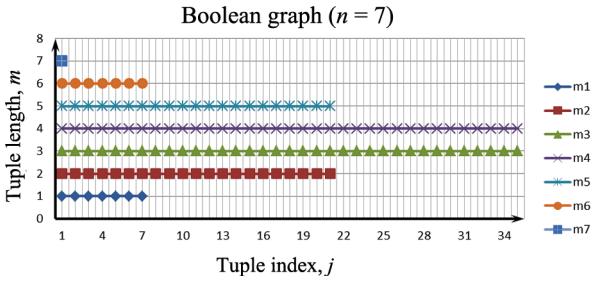

Fig. 2. Graph to represent the data structure "m-tuples"

Property 5a. Every Boolean element $y^n_{m_1,j_1}$ which does not have the maximum length ($m_1 < n$) is a subset of the number $N_{S_{m1}}$ elements $y^n_{m_2,j_2}$ represented by longer tuples ($m_2 > m_1$), where

$$N_{S_{m1}} = \sum_{m_2=m_1+1}^{n} \binom{n-m_1}{m_2-m_1}.$$

We can analyze Property 5a using the following example.

We have: Boolean of the basis set with cardinality $n = 7$. It is necessary to define a set of elements for each first element of an ordered subset of a Boolean with the given length $m_1 < n$. The set we need to define is represented by longer tuples ($m_2 > m_1$), for which element $y^n_{m_1,1}$ is a subset of each element of this set. We need to determine the number of set elements $N_{S_{m1}}$.

The solution to the problem is presented in Fig. 3.

The closed polylines $S_1, \ldots, S_5$ delineate the vertices corresponding to the desired sets of Boolean elements. The first vertices of segments that are outside the polylines $S_1, \ldots, S_5$ correspond to the elements $y^7_{m_1,1}$.

For vertex $(1, m_1)$, we define the total number of vertices belonging to longer tuples.

$$N_{>m_1} = 2^n - 1 - \sum_{m=1}^{m_1} k^n_m.$$

Table 4

Numerical characteristics of the graph to represent the data structure "m-tuples"

Numerical characteristics of the graph		
n	The number of parallel segments of the graph corresponding to the number of elements of basis set X	
m	The sequence number of the segment that corresponds to the length of data structure element $y^n_{m,j}$	$1 \le m \le n$
j	The ordinal number of the vertex of the graph that corresponds to the ordinal number of data structure element $y^n_{m,j}$. This element is located inside ordered subset Y^n_m of the same lengths	$1 \le j \le k^n_m$
k^n_m	The number of vertices on the m^{th} segment of the graph corresponding to the cardinality of subset Y^n_m	$k^n_m = \dfrac{n!}{(n-m)!\cdot m!}$
(j, m)	Coordinates of the graph vertex	

For the example shown in Fig. 3, we create a table for the definition of component Δ_{m_1}.

Table 5 shows data for a Boolean having a basis set with length $n = 7$.

Table 6 shows the results of calculations $\Delta_{m_1}(n,m)$ when $5 \le n \le 15$.

The component of Boolean elements with length $m_2 > m_1$, for which element $y^n_{m_1,j_1} \subset y^n_{m_2,j_2}$ relates to the total number of Boolean elements with length $m_2 > m_1$ is calculated as

$$\Delta_{m_1} = \frac{N_{S_{m1}}}{N_{>m_1}} * 100\,\%.$$

Dependency graph $\Delta_{m_1}(n,m)$ corresponding to Table 6 is shown in Fig 4.

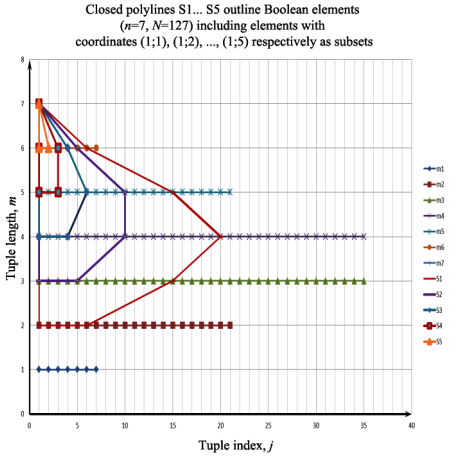

Fig. 3. Graph to represent the data structure "m-tuples" with closed polylines

Table 5

Component of elements with length $m_2 > m_1$, for which $y^7_{m_1,j_1} \subset y^7_{m_2,j_2}$

m_1	$N_{>m_1}$	$N_{S_{m_1}}$	Δ_{m_1}, %
1	120	63	52.5
2	99	31	31.3
3	64	15	23.4
4	29	7	24.1
5	8	3	37.5

Table 6

Rounded calculations $\Delta_{m_1}(n,m)$

m \ n	5	6	7	8	9	10	11	12	13	14	15
1	57	54	53	51	51	50	50	50	50	50	50
2	44	36	31	29	27	26	26	25	25	25	25
3	50	32	23	19	16	15	14	13	13	13	13
4	–	43	24	16	12	10	9	8	7	7	7
5	–	–	38	19	12	8	6	5	4	4	4
6	–	–	–	33	15	9	6	4	3	3	2
7	–	–	–	–	30	13	6	4	3	2	2
8	–	–	–	–	–	27	10	5	3	2	1
9	–	–	–	–	–	–	25	9	4	2	1
10	–	–	–	–	–	–	–	23	8	3	2
11	–	–	–	–	–	–	–	–	21	7	3
12	–	–	–	–	–	–	–	–	–	20	8
13	–	–	–	–	–	–	–	–	–	–	19

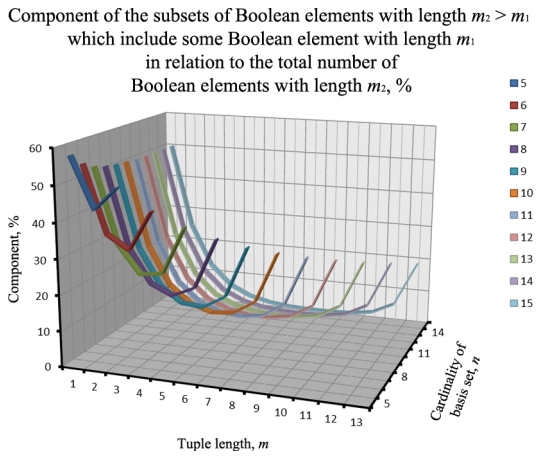

Fig. 4. Graph $\Delta_{m_1}(n,m)$

As can be seen from Table 6 and Fig. 4, the maximum values $\Delta_{m_1}(n,m)$ refer to tuples of minimal length $m_1 = 1$. We created Table 7 according to the minimum values $\Delta_{m_1}(n,m)$ from the length of tuple m_1.

There is certain regularity in determining the length of tuple m_1 for which dependence $\Delta_{m_1}(n,m)$ will have the minimum value.

Table 7

Minimum values $\Delta_{m_1}(n,m)$

n	$m_2$max	$m_1$min	Δmin
5	3	2	43.75
6	4	3	31.82
7	5	3	23.44
8	6	4	16.13
9	7	5	11.54
10	8	5	8.03
11	9	6	5.52
12	10	7	3.9
13	11	7	2.65
14	12	8	1.81
15	13	9	1.27

$$x = 2 * \left\lfloor \frac{m_2 \max}{3} \right\rfloor;$$

$$m_1 \min = (m_2 \max \%3)\ ?\ (x+1) : x,$$

where $\lfloor\ \rfloor$ is the integer part of number; % is the remainder of the division.

Conclusions. The given data structure "m-tuples based on OSAC" is a Boolean. In our case this Boolean is ordered by right-hand enumeration of the basis set elements with cardinality n. It is ordered from the lower boundary of the possible change in the index value for each element of the tuple to the upper boundary.

This data structure is essentially a sequential access list containing 2^n elements. Accordingly, it has an exponential functional dependence of the time of access to the elements of the data structure $O(2^n)$ depending on the number of input data n.

As shown in the article, the methods of working with the data structure "m-tuples based on OSAC" allow us to convert the list with sequential access into the list with direct access. In this case, the functional dependence of the estimate for the method execution time depending on the number of input data n will be changed from the exponential $O(2^n)$ to cubic $O(n^3)$. In fact, the methods considered can significantly (by several orders of magnitude) reduce the data processing time.

The applied methods of working with "m-tuples based on OSAC" can significantly speed up data processing, since:

- instead of storing, searching and processing large data arrays, it becomes possible to generate and process m-tuples according to a certain sequence of formal rules;
- the size of the memory used for storing the given data structure has decreased by $(2^n - n) \cdot$ size of(T), where T is the type of element of the basis set;
- the complexity of the data structure T for the specialization of the data structure template does not affect the speed of data processing. Instead of cumbersome operations on elements of large arrays with a complex structure of organization, these methods work with a set

of integers which are a set of indices of elements of basis set *X*.

The data structure properties described in the article allow us to determine interdependencies between *m*-tuples by their location in the structure. These properties follow the formation rules of the data structure elements in ascending order defined by a pair of indices (*j*, *m*) without execution of computational algorithms.

Analysis of the research results led to the conclusion that the use of methods for working with "*m*-tuples based on OSAC" using the data structure properties minimizes the time and computational resources in data processing to real-time.

Acknowledgements. This article is dedicated to the memory of my grandfather – Alexander Erpert (1937–2017) who was a talented scientist, a Professional with a capital letter, an excellent teacher, and an amazing lecturer. He was the person who owned the art of igniting interest in the subjects he taught and holding his audience. A kind and sympathetic person, he was a professor at Dnipropetrovsk Mining Institute, fully devoting himself to his work in this institution for more than half a century (1963–2016). Olena Syrotkina.

References.

1. Hunzinger, R. (2016). SCADA Fundamentals and Applications in the IoT. *Internet of Things and Data Analytics Handbook,* 283-293. DOI: 10.1002/9781119173601.ch17.

2. Charbonnier, S., Bouchair, N., & Gayet, P. (2014). Analysis of Fault Diagnosability from SCADA Alarms Signatures Using Relevance Indices. *IEEE International Conference on Systems, Man, and Cybernetics (SMC)* (pp. 2739-2744). DOI: 10.1109/SMC.2014.6974342.

3. Min Chen, Shiwen Mao, Yin Zhang, & Victor C. M. Leung (2014). *Big Data. Related Technologies, Challenges, and Future Prospects*. Spinger. DOI: 10.1007/s40558-015-0027-y.

4. Shi-Nash, A., & Hardoon, D. R. (2016). Data Analytics and Predictive Analytics in the Era of Big Data. *Internet of Things and Data Analytics Handbook*, 329-345. DOI: 10.1002/9781119173601.ch19.

5. Volkova, V. N., Kozlov, V. N., Mager, V. E., & Chernenkaya, L. V. (2017). Classification of Methods and Models in System Analysis. *XX IEEE International Conference on Soft Computing and Measurements (SCM)*. (pp. 183-186). DOI: 10.1109/scm.2017.7970533.

6. Massanet, S., Riera, J. V., Torrens, J., & Herrera-Viedma, E. (2015). A Consensus Model for Group Decision-Making Problems with Subjective Linguistic Preference Relations. *IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)* (pp.1-8). DOI: 10.1109/fuzz-ieee.2015.7337886.

7. Chang, L. L., Zhou, Z. J., Chen, Y. W., Liao, T. J., Hu, Y., & Yang, L. H. (2018). Belief Rule Base Structure and Parameter Joint Optimization under Disjunctive Assumption for Nonlinear Complex System Modeling. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, *48*(9), 1542-1554. DOI: 10.1109/tsmc.2017.2678607.

8. Stefanovych, T., Shcherbovskykh, S., & Drożdziel, P. (2015). The reliability model for failure cause analysis of pressure vessel protective fittings with taking into account load-sharing effect between valves. *Diagnostyka*. *16*(4), 17-24.

9. Shcherbovskykh, S., Spodyniuk, N., Zhelykh, V., Stefanovych, T., & Shepitchak, V. (2016). Development of a reliability model to analyse the causes of a poultry module failure. *Easter-European Journal of Enterprise Technologies, 4*(3(82)), 4-9. DOI: 10.15587/1729-4061.2016.73354.

10. Tkachov, V., Bublikov, A., & Isakova, M. (2013). Control automation of shearers in terms of auger gumming criterion. *Energy Efficiency Improvement of Geotechnical Systems. Proceedings of the International Forum on Energy Efficiency*, 137-145. DOI: 10.1201/b16355-19.

11. Syrotkina, O. (2015). The Application of Specialized Data Structures for SCADA Diagnostics. *System technologies. Regional interuniversity collection of scientific papers*, *4*, 72-81.

12. Syrotkina, O., Alekseyev, M., & Aleksieiev, O. (2017). Evaluation to Determine the Efficiency for the Diagnosis Search Formation Method of Failures in Automated Systems. *Eastern-European Journal of Enterprise Technologies*, *4*(9(88)), 59-68. DOI: 10.15587/1729-4061.2017.108454.

Аналіз впливу властивостей структурованих даних на оптимізацію процесів їх обробки

О. І. Сироткіна[1], *М. О. Алексєєв*[1], *В. В. Асоцький*[2], *І. М. Удовик*[1]

1 – Національний технічний університет „Дніпровська політехніка", м. Дніпро, Україна, e-mail: syrotkina.o.i@nmu.one

2 – Національний університет цивільного захисту України, м. Харків, Україна, e-mail: asotskiy@nuczu.edu.ua

Мета. Розробка математичних методів обробки „великих даних" на основі системного аналізу властивостей їх структурної організації для оптимізації основних характеристик „великих даних": збільшення швидкості обробки великих обсягів даних, що невпинно надходять зі збереженням їх актуальності.

Методика. Пропонуються математичні методи роботи зі структурою організації даних (СОД) типу „*m*-арні кортежі на основі впорядкованих множин довільної потужності". На основі аналізу властивостей СОД визначені складові попарних поєднань елементів булеана, як операндів досліджуваних операцій. Розрахована динаміка зміни складових попарних поєднань елементів булеана в залежності від потужності базової множини для різних груп СОД.

Результати. Розраховані оцінки часу виконання методів роботи із СОД типу „*m*-арні кортежі на основі впорядкованих множин довільної потужності", як функціональних залежностей від кількості даних $O(f(n))$. Визначена складова поєднань елементів булеана, для яких не потрібне виконання алгоритмів, що реалізують досліджувану операцію, оскільки цей результат визначений у самій властивості СОД.

Наукова новизна. Отримав подальший розвиток математичний метод, що дозволяє прогнозувати результат виконання деякої операції над елементами впорядкованої СОД за їх розташуванням у структурі без виконання обчислювального алгоритму. Уперше отримана аналітична залежність визначення складової кількості елементів булеана довжини m_2, що включають у себе певний елемент, представлений кортежем меншої довжини m_1, за відношенням до загальної кількості елементів булеана довжини m_2. Уперше також отримана аналітична залежність визначення мінімального екстремуму описаної вище функціональної залежності.

Практична значимість. Отримані в роботі результати можуть бути використані для мінімізації тимчасових та обчислювальних ресурсів при обробці „великих даних“, що мають впорядковану структурну організацію типу „*m*-арних кортежів на основі впорядкованих множин довільної потужності“.

Ключові слова: *„великі дані“, структура організації даних, „m-арні кортежі“, граф булеана*

Анализ влияния свойств структурированных данных на оптимизацию процессов их обработки

Е. И. Сироткина[1], *М. А. Алексеев*[1], *В. В. Асоцкий*[2], *И. М. Удовик*[1]

1 – Национальный технический университет „Днепровская политехника“, г. Днепр, Украина, e-mail: syrotkina.o.i@nmu.one

2 – Национальный университет гражданской защиты Украины, г. Харьков, Украина, e-mail: asotskiy@nuczu.edu.ua

Цель. Разработка математических методов обработки „больших данных“ на основе системного анализа свойств их структурной организации для оптимизации основных характеристик „больших данных“: увеличения скорости обработки больших объемов быстро поступающих данных с сохранением их актуальности.

Методика. Предлагаются математические методы работы с СОД типа „*m*-арные кортежи на основе упорядоченных множеств произвольной мощности“. На основе анализа свойств структуры организации данных (СОД) определены **с**оставляющие попарных сочетаний элементов булеана, как операндов исследуемых операций. Рассчитана динамика изменения составляющих попарных сочетаний элементов булеана в зависимости от мощности базового множества для различных групп СОД.

Результаты. Рассчитаны оценки времени выполнения методов работы с СОД типа „*m*-арные кортежи на основе упорядоченных множеств произвольной мощности“, как функциональных зависимостей от количества данных $O(f(n))$. Определена составляющая сочетаний элементов булеана, для которых не требуется выполнение алгоритмов, реализующих исследуемую операцию, т. к. искомый результат определен в самом свойстве СОД.

Научная новизна. Получил дальнейшее развитие математический метод, позволяющий прогнозировать результат выполнения некоторой операции над элементами упорядоченной СОД по их месторасположению в структуре без исполнения вычислительного алгоритма. Впервые получена аналитическая зависимость определения составляющей количества элементов булеана длины m_2, включающих в себя некоторый элемент, представленный кортежем меньшей длины m_1, по отношению к общему количеству элементов булеана длины m_2. Впервые также получена аналитическая зависимость определения минимального экстремума описанной выше функциональной зависимости.

Практическая значимость. Полученные в работе результаты могут быть использованы для минимизации временных и вычислительных ресурсов при обработке „больших данных“, имеющих упорядоченную структурную организацию типа „*m*-арных кортежей на основе упорядоченных множеств произвольной мощности“.

Ключевые слова: *„большие данные“, структура организации данных, „m-арные кортежи“, граф булеана*

Рекомендовано до публікації докт. техн. наук Б. І. Морозом. Дата надходження рукопису 04.02.18.