Mathematical Methods for Optimizing Big Data Processing

Olena Syrotkina
Department of Software Engineering
Dnipro University of Technology
Dnipro, Ukraine
ORCID: 0000-0002-4069-6984

Mykhailo Aleksieiev
Department of Software Engineering
Dnipro University of Technology
Dnipro, Ukraine
ORCID: 0000-0001-8726-7469

Borys Moroz
Department of Software Engineering
Dnipro University of Technology line 4:
Dnipro, Ukraine
ORCID: 0000-0002-5625-0864

Serhii Matsiuk
Department of Information Security and Telecommunications
Dnipro University of Technology
Dnipro, Ukraine
ORCID: 0000-0001-6798-5500

Olga Shevtsova
Department of Software Engineering
Dnipro University of Technology
Dnipro, Ukraine
ORCID: 0000-0002-6421-8127

Andrii Kozlovskyi
Vinnytsia Educational and Scientific Institute of Economics
Ternopil National Economic University
Vinnytsia, Ukraine
ORCID: 0000-0001-9697-1511

***Abstract* — This paper addresses the creation and application of mathematical methods to optimize the main characteristics of Big Data. This involves reducing the amount of information processed as well as increasing search speed and processing data while maintaining their respective values and reliability. The foregoing can be achieved by applying the proposed data organization structure called "*m*-tuples based on ordered sets of arbitrary cardinality". This ordered structure describes a Boolean template in general. The Boolean template is an ordered set consisting of all subsets of an ordered basis set of arbitrary cardinality and any data type. We conducted a review of modern methodologies used in solving problems of this class. We also describe data organization properties which allow us to predetermine the result of performing certain operations in the structure by its elements using their location without executing a computational algorithm. The graphs represent the "operation of inclusion" when varying the length of the operand tuple. These graphs display the dynamics of changes in the fraction of operand combinations for which one tuple is a subset of the other. We obtained logical conclusions about the influence of the properties and mathematical methods of working with the structure. This allows us to minimize computational resources.**

Keywords — Big Data, Big Data reduction methods, data organization structure, ordered set of arbitrary cardinality, Boolean graph, minimization of time and computational resources.

I. Introduction

At the present time there is rapid growth of technology-intensive industries where information and telecommunication technologies form an integral part. There are stringent requirements for information and technological solutions related to the tasks of processing, storing, analysing and managing Big Data when time, computing and information restrictions apply. Therefore, an important problematic aspect in this area is to provide the requirements stated above for the development and application of mathematical methods, techniques, and tools when working with Big Data.

II. Analysis of Recent Publications

One of the most common problems in applying the methods of working with Big Data is their computational visibility (or immensity). When there is a significant increase in the amount of processed data, the number of operations grows exponentially. This phenomenon is called "combinatorial explosion" [1].

In order to solve this problem, we use methods to reduce the space of analysed states. In other words, the space is divided into sufficiently independent subspaces with characteristic incomplete solutions. Several methods have been developed in this area, including the use of symmetries in checking equivalence of states, abstractions based on dependency studies, predicate abstractions, imposing restrictions on the search space, directional search, and heuristic methods [2, 5, 8, 9].

Methods of analysing Big Data were also further developed. These include machine learning, artificial intelligence, artificial neural networks, distributed processing of flows and events as well as visual methods of data research [2 - 9].

In work [1], the author conducted an analysis based on recognized criteria regarding artificial intelligence methods containing "combinatorial explosion". Various search algorithms were also evaluated to determine the best solution to the "combinatorial explosion" problem.

In work [2], the authors conducted research on Big Data reduction methods. They examined network theory, Big Data compression, dimension reduction, redundancy elimination, data mining, and machine learning. It was concluded that, at the current time, there are no existing methods to solve the problem of Big Data complexity in terms of volume and variety.

In work [3], the authors considered the problem of Big Data processing. They examined the most efficient technique for dealing with vast and complex amounts of fast-growing information called Hadoop MapReduce. This article also studies relevant non-relational data models and modelling techniques currently available. It was concluded that, at present, there is no technique which can help in managing Big Data.

In work [4], the authors provided a review of feature-selection techniques to significantly reduce the complexity of Big Data. These include the following techniques: filter, wrapper and embedded methods as well as hybrid, ensemble and integrative methods. It was concluded that further attention is required on processing complex volumes of Big Data.

In work [5], the authors outlined the basic features characteristic of Big Data. They proposed a formal model of

Big Data and described its elements. The research also examined the most advanced methods to analyse large volumes of structured and unstructured data. These comprised artificial neural network predictive analytics, statistics and natural language processing methods. At the same time, the authors concluded that analysing and processing Big Data requires further research.

Work [6] is focused on concepts and techniques devoted to Big Data processing. The authors examined several categories of Big Data such as human-generated or computer-generated data, structured data, unstructured data, and semi-structured data. They also reviewed Big Data technologies. One of the Big Data concepts they focused on is Apache Hadoop. The Hadoop framework consists of two main core components. These are the Hadoop Distributed File System and MapReduce.

In work [7], the authors proposed a workflow transformation for real-time Big Data processing. This was based on heterogeneous systems to minimize cost and defined different patterns as rules for a workflow transformation algorithm. The research experiments concluded that the proposed workflow transformation method can effectively reduce communication and computational costs.

In work [8], the authors proposed an approach to Big Data visualization based on clustering techniques. They also considered an approach to solving the problem of clustering, which is when only one solution can be shown converging to a local minimum. Therefore, the paper proposed a method to optimize the K-means algorithm which itself has multiple random starting points. This helps to streamline local minimums and visualize various solutions to the problem.

Work [9] represents a powerful method for visualization, cluster extraction and data mining. This method is called the self-organizing map (SOM). SOM has been used successfully for Big Data where traditional methods are often insufficient. The authors suggested an approach where data topology can be integrated into the SOM visualization. This can be achieved by introducing a weighted connectivity matrix and draping it over the SOM. A complementary approach is offered for exploiting data topology, thus allowing a graph model to be used in the data space.

III. Aim of Research

Described below are the following objectives of the study:

- Determining ways to minimize the time and computational resources involved in Big Data processing. This can be achieved by analysing the properties of the data organization structure called "*m*-tuples based on ordered sets of arbitrary cardinality (OSAC)".
- Developing mathematical methods for reducing the space of analysed states. These are based on the derivation of new functional dependencies between the given data structural elements according to their location in the ordered structure.

The selection in favour of "*m*-tuples based on OSAC" was made due to the fact that this ordered structure describes a Boolean template in general. It is an ordered set of all subsets within an ordered basis set of arbitrary cardinality for any data type. When representing a template class that describes the given data structure, it is possible to create an arbitrary algorithm for ordering the data of the basis set for a particular data type and for a specific task of working with these data.

Mathematical methods for working with this data structure are offered for use in solving problems associated with performing operations on a variety of combinations of parameters of a given basis set.

An example of the application of a certain data structure can be automation of analysis and processing of multiple streams of diagnostic information. This information is automatically generated by the hardware and software complex of an automated control system in the event of any malfunction. Typically, such flows in distributed multi-tasking, multi-user automated systems can contain thousands of messages per second.

To solidify the description and analysis of methods for working with this data structure, this paper uses the ordering of the basis data set in ascending order.

IV. Main Terms and Definitions of Given Data Organization Structure

The data organization structure called "*m*-tuples based on OSAC" includes the following main parameters and methods: $\boldsymbol{X}$ is an ascending ordered basis set; $\boldsymbol{I}$ is an ascending ordered basis set of indices of elements for set X; $\boldsymbol{x_i}$ is the i^{th} element of set X; $\boldsymbol{i}$ is an element of set I and is an element index of set X; $\boldsymbol{n}$ is the cardinality of sets X and I.

$$\begin{cases} I = \{1, \ 2, \ \ldots, \ n\} = \{i \mid 1 \le i \le n\}, \\ X = \{x_1, \ \ldots, \ x_i, x_{i+1}, \ \ldots, \ x_n\} = \\ = \{x_i \mid 1 \le i \le n, \ \ \forall i \in [1, n) \to x_i < x_{i+1}\}, \\ n = |X| = |I| \end{cases} \tag{1}$$

$\boldsymbol{2^X}$ is the Boolean of set X; $\boldsymbol{2^I}$ is the Boolean of set I.

$$\begin{cases} 2^X = \{\varnothing, Y_1^n, \ \ldots, \ Y_m^n, \ \ldots, \ Y_n^n\} = \\ = \{Y_m^n \mid 1 \le m \le n\}, \\ 2^I = \{\varnothing, I_1^n, \ \ldots, \ I_m^n, \ \ldots, \ I_n^n\} = \\ = \{I_m^n \mid 1 \le m \le n\} \end{cases} \tag{2}$$

Y_m^n is a subset of Boolean 2^X. Its elements are m-tuples $y_{m,j}^n$. They consist of the elements of set X. These elements are in ascending order by right-handed enumeration of indices i_η (for the example we consider in this paper). They are ordered from the lower boundary of a possible change in the index value $\alpha_{m,\eta}^n$ for each element of tuple x_{i_η} to the upper $\beta_{m,\eta}^n$.

$$\begin{cases} Y_m^n = \{y_{m,j}^n \mid y_{m,j}^n = (x_{i_1}, \ \ldots, \ x_{i_\eta}, \ \ldots, \ x_{i_m})\}, \\ 1 \le \eta \le m, \\ \alpha_{m,\eta}^n \le i_\eta \le \beta_{m,\eta}^n, \\ \alpha_{m,\eta}^n = \eta, \\ \beta_{m,\eta}^n = n - m + \eta, \\ \forall \eta \in [1, m) \to x_{i_\eta} < x_{i_{\eta+1}} \end{cases} \tag{3}$$

$y_{m,j}^n$ is an m-tuple which is the j^{th} element of set Y_m^n consisting of elements from basis set X of cardinality n; $\boldsymbol{m}$ is

the length of tuple $y^n_{m,j}$; $\boldsymbol{j}$ is the index of tuple $y^n_{m,j}$ in ordered set Y^n_m. $\boldsymbol{I}^n_m$ is a subset of Boolean 2^I. Its elements are *m*-tuples $\boldsymbol{i}^n_{m,j}$ that correspond to tuples $y^n_{m,j}$.

$$\begin{cases} I^n_m = \{i^n_{m,1},\ i^n_{m,2},\ \ldots,\ i^n_{m,k^n_m}\} = \\ = \{i^n_{m,j} \mid i^n_{m,j} = (i_1,\ \ldots,\ i_\eta,\ \ldots,\ i_m),\ 1 \le j \le k^n_m\}, \\ i_\eta \in I, \\ 1 \le \eta \le m, \\ \eta \le i_\eta \le n - m + \eta, \\ \forall\, \eta \in [1, m) \to i_{\eta+1} - i_\eta \ge 1 \end{cases} \tag{4}$$

$\boldsymbol{\eta}$ is an element index number in tuple $y^n_{m,j}$; $\boldsymbol{i_\eta}$ is an index of the η^{th} element of tuple $y^n_{m,j}$ in basis set *X*; $\boldsymbol{i}^n_{m,j}$ is an *m*-tuple of indices of basis set elements corresponding to $y^n_{m,j}$; $\alpha^n_{m,j}$ is the lower boundary of index value change i_η in *m*-tuple $y^n_{m,j}$; $\beta^n_{m,j}$ is the upper boundary of index value change i_η in *m*-tuple $y^n_{m,j}$; k^n_m is the cardinality of the sets Y^n_m and I^n_m.

$$k^n_m = \left|Y^n_m\right| = \left|I^n_m\right| = \binom{n}{m} = \frac{n!}{(n-m)!\cdot m!} \tag{5}$$

K^n is a set with cardinalities k^n_m of the subsets Y^n_m and I^n_m for two Booleans 2^X and 2^I.

$$\begin{aligned} K^n = \{k^n_1 = n,\ \ k^n_2 = \frac{(n-1)*n}{2},\ \\ \ldots,\ \ k^n_m,\ \ \ldots, k^n_{n-1} = n,\ \ k^n_n = 1\} = \\ \{k^n_n \mid k^n_n = \binom{n}{m},\ \ 1 \le m \le n\} \end{aligned} \tag{6}$$

J^n_m is an ordered set of indices *j* for set elements Y^n_m and I^n_m.

$$J^n_m = \{1,2,\ \ldots,\ k^n_m\} = \{j \mid 1 \le j \le k^n_m\} \tag{7}$$

A more detailed description of the main terms and definitions as well as mathematical methods for working with the data structure called "*m*-tuples based on OSAC" is given in papers [10 - 12].

V. Problem Statement in General Terms

We have the given data structure called "*m*-tuples based on OSAC". For this data structure it is necessary to develop a mathematical method for solving system (8) for any correct combination of parameter values: *n*, m_1, m_2, j_1, j_2. The foregoing must be achieved in minimal time and using minimum computational resources for data processing and analysis.

$$\begin{cases} y^n_{m,j} = (y^n_{m_1,j_1}\ \ op\ \ y^n_{m_2,j_2}) \ne \varnothing, \\ op \in \{\subset, \cap, \cup\}, \\ 1 \le m_1 \le m_2 \le n, \\ 1 \le j_1 \le \binom{n}{m_1}, \\ 1 \le j_2 \le \binom{n}{m_2} \end{cases} \tag{8}$$

In system (8) we accept the following designations:

$y^n_{m,j}$ is an *m*-tuple and a Boolean element 2^X; $\boldsymbol{n}$ is the cardinality of an ordered basis set *X*; $\boldsymbol{m_1}$, $\boldsymbol{m_2}$ are operand tuple lengths; $\boldsymbol{j_1}$, $\boldsymbol{j_2}$ are indices (ordinal numbers) of operand *m*-tuples in ordered sets $Y^n_{m_1}$ and $Y^n_{m_2}$; $\boldsymbol{m}$, $\boldsymbol{j}$ are tuple indices for the result of operation *op*.

The main idea of this method is to solve system (9) by deriving a set of functional dependencies:

$$F = \{f_\gamma(n, m_1, m_2, j_1, \eta)\}.$$

It is based on the analysis of the data structure properties under various initial conditions, i.e. for any possible valid combinations of argument values f_γ.

$$\begin{cases} j_2 = f_\gamma(n, m_1, m_2, j_1, \eta), \\ (j_2 \in J2^n_{m_2}(y^n_{m_1,j_1})) \to \\ (J2^n_{m_2}(y^n_{m_1,j_1}) = \{f_\gamma(n, m_1, m_2, j_1, \eta)\}), \\ J2^n(y^n_{m_1,j_1}) = \{J2^n_{m_2}(y^n_{m_1,j_1})\}, \\ m_1 < m_2 \le n \end{cases} \tag{9}$$

In system (9) we accept the following designations:

$\boldsymbol{\eta}$ is a location of element $y^n_{m_1,j_1}$ in tuple $y^n_{m_2,j_2}$; $J2^n_{m_2}(y^n_{m_1,j_1})$ is a set of indices j_2 determining elements $y^n_{m_2,j_2}$ from one of the subsets $Y^n_{m_2}$ of Boolean 2^X which meet the condition $m_1 < m_2 \le n$ and for which system (8) has a decision; $J2^n(y^n_{m_1,j_1})$ is a set of indices j_2 determining elements $y^n_{m_2,j_2}$ for all subsets of Boolean 2^X and for which system (8) has a decision.

VI. Materials and Methods

The ability to determine new properties of the ordered data structure and derive new functional dependencies between elements on their basis follows the rules for data structure formation and analysis of the location of elements. It is defined by a pair of indices (*j*, *m*).

Boolean 2^X is presented graphically, in general form, in Fig. 1 and Fig. 2.

Each point in the graphs shown in Fig. 1 and Fig. 2 corresponds to a unique *m*-tuple $y^n_{m,j}$. These are formed from the elements of an ordered basis set *X* with cardinality *n* and can be uniquely determined by a pair of indices (*j*, *m*). Closed polylines shown in the graphs outline all the elements of Boolean 2^X.

As a result of the research conducted regarding the properties and functional relationships between data structure elements of "*m*-tuples based on OSAC", we proved certain

properties of the given data structure and have described them below.

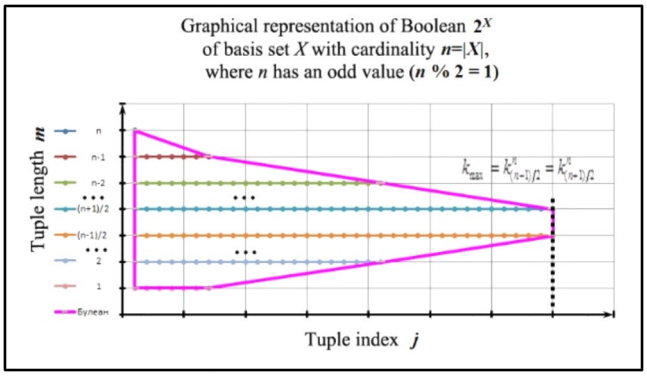

Fig. 1. Graphical representation of Boolean 2^X with an odd cardinality value of the basis set

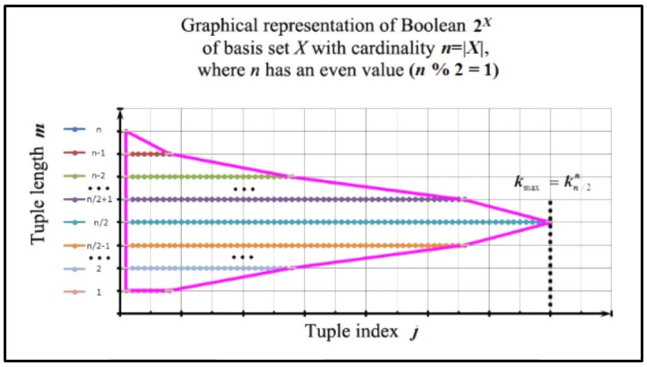

Fig. 2. Graphical representation of Boolean 2^X with an even cardinality value of the basis set

A. Certain properties of two adjacent elements of given data structure

Property A.1. For two adjacent elements $i^n_{m,j}$ and $i^n_{m,j+1}$ of set I^n_m ordered by right-handed enumeration of elements of basis set I with cardinality n, the following condition is fulfilled:

$$(\exists i^j_{\eta'} \in i^n_{m,j}) \ \ \& \ \ (\exists i^{j+1}_{\eta'} \in i^n_{m,j+1}) \rightarrow$$
$$\rightarrow \begin{cases} i^{j+1}_{\eta'} - i^j_{\eta'} = 1, \\ (\eta' > 1) \ \ ? \ \ (1 \le \eta_2 = \eta_1 < \eta', \quad i^{j+1}_{\eta_2} = i^j_{\eta_1}) \end{cases}$$

Property A.2. (Opposite). Elements i^n_{m,j_1} and i^n_{m,j_2} of set I^n_m ordered by a right-handed search of elements from basis set I with cardinality n are adjacent $|j_2 - j_1| = 1$ if the following condition is fulfilled:

$$\left(\eta \in [1,m], \ \ i^{j_1}_{\eta} \in i^n_{m,j_1}, \ \ i^{j_2}_{\eta} \in i^n_{m,j_2}, \ \ \Delta i_{\eta} := \left|i^{j_1}_{\eta} - i^{j_2}_{\eta}\right|\right) \rightarrow$$
$$\rightarrow \left((\Delta i_1 = 1) \vee (\underset{\eta=1}{\overset{\eta^*}{\&}} (\Delta i_{\eta} = 0) \,\&\, (\Delta i_{\eta^*+1} = 1)), \ \ 1 \le \eta^* < m \right)$$

At the same time, the order of elements in set I^n_m is defined as follows:

$$((\Delta i_{\eta^*+1} = i^{j_1}_{\eta^*+1} - i^{j_2}_{\eta^*+1}) > 0) \ \ ? \ \ (j_1 > j_2) : (j_2 > j_1)$$

B. Certain properties of complements of data structure elements to basis set

Property B.1. Complement to basis set X to element $y^n_{m,j} \in Y^n_m$ is an element that belongs to set Y^n_{n-m}.

$$\overline{y^n_{m,j}} \in Y^n_{n-m}$$

Property B.2. Complement to basis set I to element $i^n_{m,j} \in I^n_m$ is an element that belongs to set I^n_{n-m}.

$$\overline{i^n_{m,j}} \in I^n_{n-m}$$

Property B.3. Complement to basis set X to element $y^n_{m,j} \in Y^n_m$ is an element that can be defined as follows:

$$\overline{y^n_{m,j}} = y^n_{n-m,k^n_m-(j-1)}$$

C. Certain properties of intersections and unions for data structure elements

We have:

$$\begin{cases} y^n_{m,j} = (y^n_{m_1,j_1} \quad op \quad y^n_{m_2,j_2}), \\ i^n_{m,j} = (i^n_{m_1,j_1} \quad op \quad i^n_{m_2,j_2}), \\ op \in \{\cap, \cup\} \end{cases} \qquad (10)$$

For the expressions in system (10), all of the following properties are applicable:

Property C.1.1. (Property of annulment for the operation of intersection). If one of the operands of intersection is a subset of another operand, the result of the operation of the intersection will be the first operand.

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

where NOP – no operation.

Consequence from property C.1.1. If at least one of the operands of the intersection of the Boolean elements along the basis set with cardinality n is an n-tuple, then their intersection will be the second operand.

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

Property C.1.2. (Property of annulment for the operation of union). If one of the operands of union is a subset of another operand, the result of the operation of union will be the second operand.

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

Consequence 1 from property C.1.2. If at least one of the operands of the union of Boolean elements along the basis set with cardinality *n* is an *n*-tuple, then their union will be an *n*- tuple.

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

Consequence 2 from property C.1.2. If operands of the union are the first two or respectively the last two elements in the subsets $Y^n_{m_1}$ and $Y^n_{m_2}$ with different tuple lengths, then an element with a shorter tuple length is a subset of the element with the longer tuple length.

$$((m_1 \neq m_2) \wedge ((j_1 = j_2 = 1) \vee ((j_1 = k^n_{m_1}) \wedge (j_2 = k^n_{m_2})))) \quad ?$$
$$((m_1 < m_2) \quad ? \quad (y^n_{m_1,j_1} \subset y^n_{m_2,j_2}) \quad : \quad (y^n_{m_1,j_1} \supset y^n_{m_2,j_2}))$$

Property C.2. (Property of idempotency of intersection/ union operations). The intersection/union of a Boolean element with itself along the basis set is the same *m*-tuple.

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

Property C.3.1. The intersection of two different $(n-1)$ - tuples along the basis set with cardinality *n* is an $(n-2)$-tuple.

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

Property C.3.2. The union of two different $(n-1)$-tuples along the basis set with cardinality *n* is an *n*- tuple.

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

Property C.4.1. (*Property of complement for intersections*). If one of the operands of the intersection of Boolean elements along the basis set with cardinality *n* is the complement to the other operand to the basis set, then the result of the operation will be *Ø*.

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

Property C.4.2. (*Property of complement* for unions). If one of the operands of the union of Boolean elements along the basis set with cardinality *n* is the complement to the other operand to the basis set, then the result of the operation will be an *n*-tuple.

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

Property C.5. Valid range for changing the length of the resulting *m*-tuple after the operation of union is as follows:

$$\max(m_1, m_2) \le m \le \min(m_1 + m_2, n)$$

D. Certain properties of the inclusion of data structure elements

We have:

$$\begin{cases} y^n_{m,j} = (y^n_{m_1,j_1} \subset y^n_{m_2,j_2}), \\ i^n_{m,j} = (i^n_{m_1,j_1} \subset i^n_{m_2,j_2}), \\ 1 \le m_1 < m_2 \le n \end{cases} \tag{11}$$

For expressions in system (11), all the properties listed below are applicable:

Property D.1. Expression (11) is true for $\binom{n-m_1}{m_2-m_1}$ elements of set $Y^n_{m_2}$.

Consequence from property D.1. If $j_1 = 1$, then expression (1) is true for the first $\binom{n-m_1}{m_2-m_1}$ elements of set $Y^n_{m_2}$.

Property D.2. Expression (11) is true for $N_{m_1} = \sum_{m_2=m_1+1}^{n} \binom{n-m_1}{m_2-m_1}$ elements of Boolean 2^X.

A more detailed description of the properties and mathematical methods to work with the given data structure "*m*-tuples based on OSAC" is given in [10−12]. In the papers [10−12], there were also mathematical methods considered for deriving a set of functional dependencies $f_\gamma(n, m_1, m_2, j_1, \eta)$ and determining the truth of system (9) under given initial conditions: m_1=1; j_1= {1, 2, 3}.

VII. Results and Discussion

For Boolean elements 2^X with cardinality 2^n and tuple lengths $[m_1; m_2]$ which are applied as operands $y^n_{m_1,j_1} \subset y^n_{m_2,j_2}$, we can define the proportion of operand combinations for which the condition is fulfilled:

$$(y^n_{m_1,j_1} \subset y^n_{m_2,j_2}) = true$$

Define the total number of combinations of the operation operands $y^n_{m_1,j_1} \subset y^n_{m_2,j_2}$, where $m_2 = m_1$.

$$N^n_{m_2=m_1} = N^n_{m_1} = k^n_{m_1} + \binom{k^n_{m_1}}{2}$$

Define the number of operand combinations for the condition $(y^n_{m_1,j_1} \subset y^n_{m_2,j_2}) = true$ where $m_2 = m_1$.

$$Nt^n_{m_2=m_1} = Nt^n_{m_1} = k^n_{m_1}$$

The proportion of operand combinations represented by Boolean elements with tuple lengths $[m_1; m_2 = m_1]$ for the condition $(y^n_{m_1,j_1} \subset y^n_{m_2,j_2}) = true$ in relation to the total number of operand combinations with tuple lengths $[m_1; m_2 = m_1]$ is calculated as follows:

$$\Delta^n_{m_2=m_1} = \Delta^n_{m_1} = \frac{Nt^n_{m_1}}{N^n_{m_1}} \cdot 100\% = \frac{k^n_{m_1}}{k^n_{m_1} + \binom{k^n_{m_1}}{2}} \cdot 100\%$$

Define the total number of combinations of operands for the condition $y^n_{m_1,j_1} \subset y^n_{m_2,j_2}$, where $m_2 > m_1$.

$$N^n_{m_2>m_1} = N^n_{m_1,m_2} = k^n_{m_1} * k^n_{m_2}$$

Define the total number of combinations of operands for the condition $(y^n_{m_1,j_1} \subset y^n_{m_2,j_2}) = true$ where $m_2 > m_1$.

$$Nt^n_{m_2>m_1} = Nt^n_{m_1,m_2} = k^n_{m_1} * \binom{n-m_1}{m_2-m_1}$$

The proportion of operand combinations represented by Boolean elements with tuple lengths $[m_1; m_2 > m_1]$ which meet the condition $(y^n_{m_1,j_1} \subset y^n_{m_2,j_2}) = true$ in relation to the total number of operand combinations with tuple lengths $[m_1; m_2 > m_1]$ is calculated as follows:

$$\Delta^n_{m_2>m_1} = \Delta^n_{m_1,m_2} = \frac{Nt^n_{m_1,m_2}}{N^n_{m_1,m_2}} * 100\% = \frac{k^n_{m_1} * \binom{n-m_1}{m_2-m_1}}{k^n_{m_1} * k^n_{m_2}} * 100\%$$

Table 5 shows calculation results for $\Delta^5_{m_1,m_2}$.

TABLE I. RESULTS FOR $\Delta^5_{m_1,m_2}$

m_1	m_2	N	Nt	Δ, %
1	**2**	**3**	**4**	**5**
1	1	15	5	33
	2	50	20	40
	3	50	30	60
	4	25	20	80
	5	5	5	100
2	2	55	10	18
	3	100	30	30
	4	50	30	60
	5	10	10	100
3	3	55	10	18
	4	50	20	40
	5	10	10	100
4	4	15	5	33
	5	5	5	100
5	5	1	1	100

The graph of $\Delta^5 = f_5(m_1, m_2)$ is shown in Fig. 3. In order to compare how $\Delta^n_{m_1,m_2}$ changes when n is increased, Fig. 4 shows the graph of $\Delta^{15} = f_{15}(m_1, m_2)$.

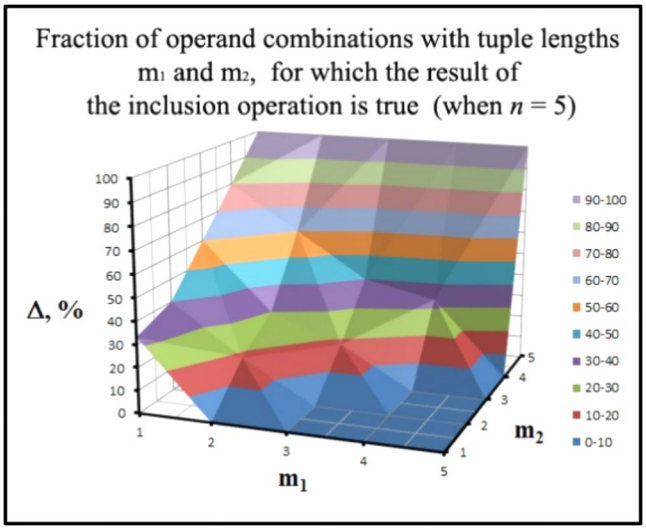

Fig. 3. Graphical representation of $\Delta^n_{m_1,m_2}$ for operands represented by m-tuples when $n = 5$

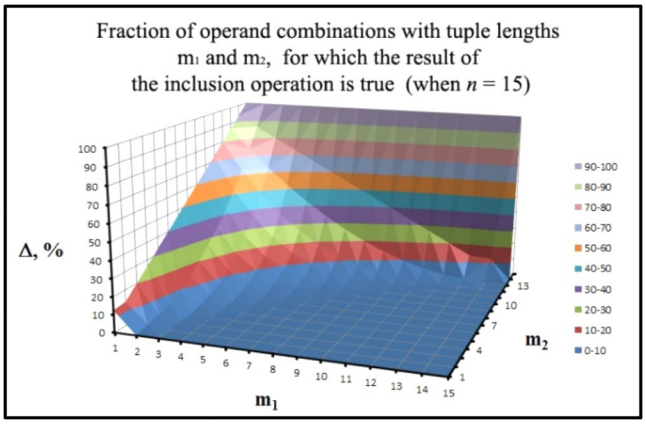

Fig. 4. Graphical representation of $\Delta^n_{m_1,m_2}$ for operands represented by m-tuples when $n = 15$

From these graphs we can see that an increase in n for operands with the same tuple lengths m_1 and m_2, the proportion of operand combinations decreases. At the same time, one of them is a subset of the other. The value of the fraction of operand combinations analysed starts increasing at the last values of scale m_2. Therefore, the fraction of operand combinations, where one of them is a subset of the other, increases with an increase in the difference $m_2 - m_1$.

VIII. PRACTICAL APPLICATION OF METHODS FOR WORKING WITH THE GIVEN DATA STRUCTURE

In general, there are certain principles to bear in mind when accessing Boolean elements generated on the basis of ordered enumeration of the basis data set. These include sequential data access algorithms with exponential execution time of the algorithm.

There are certain mathematical methods used in this paper for working with the data structure based on the properties of "m-tuples". These relate to methods of direct access to data with a constant, linear or cubic execution time of the algorithm.

Taking into account that modern processors operate with a clock frequency of more than 3 GHz and that, on average, an arithmetic operation with accumulation of type is

performed in 4 cycles, we take the following estimated time of execution of one operation for the methods considered:

$$\tau = \frac{4}{3 \cdot 10^9 HZ} = 1.33 \cdot 10^{-9} s = 1.33 ns \qquad (12)$$

Estimates of the execution time of the methods are summarised in Table II.

TABLE II. ESTIMATES OF THE EXECUTION TIME FOR WORKING WITH THE GIVEN DATA STRUCTURE METHODS

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	$6.65*10^{-9}$	$1.663*10^{-7}$	$4.256*10^{-8}$
25	$3.325*10^{-8}$	$2.078*10^{-5}$	$4.463*10^{-2}$
30	$3.99*10^{-8}$	$3.591*10^{-5}$	1.428
40	$5.32*10^{-8}$	$8.512*10^{-5}$	$1.462*10^3 \approx 25$ *min*
45	$5.985*10^{-8}$	$1.212*10^{-4}$	$4.680*10^4 \approx 13$ *h*
50	$6.65*10^{-8}$	$1.663*10^{-4}$	$1.498*10^6 \approx 17$ days

Table 2 shows that with an increase in n, methods of direct access to the data structure elements are several orders of magnitude faster than sequential access methods. The difference in the estimation of execution time of the methods is especially noticeable at $n \geqslant 40$.

IX. CONCLUSIONS

In this paper we considered the methods of working with the data organization structure called "*m*-tuples based on OSAC". Comparing these methods with other well-known methods and algorithms for working with Big Data [1−9], they allow us to obtain results when working with data structure elements without executing complicated computational algorithms. These depend on the location of the operands in the structure defined by a pair of indices (*j*, *m*) using the considered data structure properties and a set of derived functions $f_\gamma(n, m_1, m_2, j_1, \eta)$ [10−12].

Time estimates for obtaining the result changes from cubic $O(n^3)$ to linear $O(n)$ [10−12]. This approach allows the time and computational resources involved in processing to be minimized to a real-time scale when processing the given data organization structure.

REFERENCES

[1] A. Gaur, "Search techniques to contain combinatorial explosion in artificial intelligence," International Journal of Engineering Research & Technology, vol. 1, issue 7, pp. 1−7, September 2012.

[2] S. Yadav, A. Phulre, M. Pradesh, "A literature review on Big Data reduction methods," International Journal of Electrical, Electronics and Computer Engineering, pp. 92−99, June 2017.

[3] H. Hashem, D. Ranc, "An integrative modeling of Big Data processing," International Journal of Computer Science and Applications, ©Technomathematics Research Foundation, vol. 12, pp. 1–15, January 2015.

[4] K. Tadist, S. Najah, N. Nikolov, F. Mrabti, A. Zahi, "Feature selection methods and genomic Big Data: a systematic review," Journal of Big Data, pp. 1–24, August 2019.

[5] N. Shakhovska, O. Veres, M. Hirnyak, "Generalized formal model of Big Data," Econtechmod. An International Quarterly Journal, vol. 5, pp. 33–38, February 2016.

[6] B. Suvarnamukhi, M. Seshashayee, "Big Data concepts and techniques in data processing," International Journal of Computer Sciences and Engineering, vol. 6, Issue-10, pp. 712–714, Oct 2018.

[7] Y. Ishizuka, W. Chen, I. Paik, "Workflow transformation for real-time Big Data processing," IEEE International Congress on Big Data, pp. 31–318, 2016.

[8] I. Bifulco, S.Cirillo, "Discovery multiple data structures in Big Data through global optimization and clustering Methods," IEEE 22nd International Conference Information Visualization, pp. 117–121, 2018.

[9] K. Tasdemir, E. Merenyi, "Exploiting data topology in visualization and clustering of self-organizing maps," IEEE Transactions on Neural Networks, vol. 20, pp. 549–562, April 2009.

[10] O. Syrotkina, M. Alekseyev, V. Asotskyi, and I. Udovyk, "Analysis of how the properties of structured data can influence the way these data are processed," Naukovyi Visnyk NHU, Dnipro, vol. 3 (171), 2019, pp. 119–129.

[11] O. Syrotkina, M. Alekseyev, L. Meshcheriakov, and B. Moroz, "Methods of working with "big data" based on the application of "m-tuple" theory,." Computer-Integrated Technologies: Education, Science, Production, Lutsk, vol. 36, 2019, pp. 140–152.

[12] O. Syrotkina, M. Alekseyev, and I. Udovyk "Graphical and analytical methods for processing "Big Data" based on the analysis of their properties," System Technologies, Dnipro, vol. 3(122), 2019. pp. 78–90.