This study's object is adaptive compression of general-form binary sequences based on binary binomial numbers.

The task addressed is to enable high compression speed of binary information based on binomial numbers under the condition of uncertainty in the characteristics of the binary sequences being compressed.

One of the factors that reduce the efficiency of binomial compression is uncontrolled transitions of the number of unit combinations to the region of inefficient use, the worst compression ratios.

In this regard, the work applies an adaptive approach to binomial compression, based on the choice of an encoding technique depending on the number of units of the processed sequence.

This approach yields the following result: a several-fold reduction in the amount of time spent processing binary combinations that are not compressible. Consequently, this leads to an increase in the average speed of binomial compression with a small, up to three to five percent, decrease in the compression ratio.

The adaptive compression process model includes the stages of comparing the calculated numbers of binary units with the compression conditions and selecting the coding technique based on binary binomial numbers. If the current value of the number of units goes beyond the compression conditions, the calculation of the number of units is stopped, and the processed sequence remains unchanged. This eliminates unnecessary time costs when the compression ratio becomes less than unity.

In practice, the adaptive approach to compression based on binary binomial numbers is effective in the case when the binary sequences being compressed have uncertain characteristics, and their preliminary evaluation is impossible or difficult

Keywords: adaptive compression, binomial numbers, coding selection, binomial-vector method, compression time

UDC 004.627

DOI: 10.15587/1729-4061.2025.335729

SPEEDING UP BINOMIAL COMPRESSION BASED ON BINARY BINOMIAL NUMBERS

Igor Kulyk
Corresponding author
PhD, Associate Professor*
E-mail: i.kulyk@ekt.sumdu.edu.ua

Maryna Shevchenko
PhD, Senior Lecturer*

Vitalii Grynenko
PhD, Associate Professor*

Maksym Hermes
PhD Student*

*Department of Electronics and Computer Technics
Sumy State University
Kharkivska str., 116, Sumy, Ukraine, 40007

Received 06.05.2025
Received in revised form 01.07.2025
Accepted date 18.07.2025
Published date 29.08.2025

How to Cite: *Kulyk, I., Shevchenko, M., Grynenko, V., Hermes, M. (2025). Speeding up binomial compression based on binary binomial numbers. Eastern-European Journal of Enterprise Technologies, 4 (9 (136)), 26–33. https://doi.org/10.15587/1729-4061.2025.335729*

1. Introduction

Data compression is an effective measure to increase the productivity of information systems [1, 2]. The use of compression methods and algorithms leads to the following:

1) an increase in the speed of information transmission over communication channels;

2) a decrease in the amount of data that needs to be stored, and, ultimately, a decrease in the number of memory accesses.

But the effectiveness of compression methods can significantly decrease if the statistical or structural characteristics of the processed information change. This also applies to binary compression methods. Being developed for one type of binary information, if the characteristics of binary sequences change, the methods can show mediocre or even negative results, not providing the stated degree or time of compression. Such situations in distributed information systems are common since the data flows circulating in them are characterized by the presence of various types of information – text, graphics, images, etc. This problem could be solved by devising and implementing adaptive compression methods that would adapt to the characteristics of changing binary sequences and the conditions of their transmission/storage. It should be added that such compression methods should have a small amount of hardware and software costs, be undemanding to computing resources, and have the possibility of application in machine arithmetic.

Thus, research on obtaining adaptive compression methods capable of minimizing losses in the degree or time of compression under the condition of uncertainty of the data being processed is relevant and practically significant.

2. Literature review and problem statement

Adaptive compression methods are widely used in various information systems. In [3, 4], adaptive Huffman codes are considered, which can adapt to changing symbol statistics, thereby maintaining a good degree of compression. As noted in [3], the disadvantage is that the adaptation technique used for Huffman codes is complex, requiring significant time and hardware and software resources. This is explained by the fact that each time a symbol appears, the code tree needs to be rebuilt, and variable-length codes are recalculated. Moreover, in [4, 5] it is added that Huffman coding is focused on compressing symbols with a limited binary length (because of this, text information is mainly compressed). In [3, 6] it is indicated that when implementing adaptive arithmetic coding, one also encounters the complexity of the adaptation mechanism for changing the statistical data model, which is a disadvantage of this compression method. To adapt to a change in the data model, it is necessary to recalculate the accumulated symbol frequencies each time, to sort the values of the frequency counters, and to perform operations with large-bit integers. Thus, the bulkiness of data models and the large number of probabilistic characteristics when using Huffman coding [4, 5] and arithmetic coding [3, 6] significantly complicate the development of adaptive compression methods.

Specialized number systems, for example, Fibonacci [7], factorial, considered in [8], or binomial, which are discussed in [9], have great prospects for constructing methods of representing and processing information, in particular data compression. Belonging to structural positional number systems, they have properties that are inaccessible to the conventional binary number system. As indicated in [8, 9], such properties include the generation of various combinatorial objects, the ability to detect errors, and the economical representation of code sequences with given constraints.

In [10], an adaptive method for compressing binary information based on linear Fibonacci forms is proposed. As a drawback, it can be pointed out that the specified Fibonacci forms do not always provide a compact representation of binary blocks represented by integers. And the designed adaptation mechanism by selecting the length of byte chains and the bases of number systems for each block is cumbersome and requires certain computational resources. In addition, the considered adaptation is accompanied by a significant amount of service information.

In [11], it is stated that in the structure of any code sequence that satisfies a given constraint, a structural number corresponding to any structural positional number system can be found. In [12], equilibrium combinations are considered as code sequences, and the number of binary units is considered as a code constraint. There, binomial binary numbers are defined as structural numbers in the basis of equilibrium combinations. The disadvantage of studies in [12] is that binomial compression is considered only in a generalized form and only for equilibrium codes. The disadvantage of [13, 14] is that they do not investigate the issues and conditions of using binomial numbers for compression not only of equilibrium codes but also of code sequences of a general type. In [15, 16], numerical compression based on the binomial number system is studied, which, unlike compression based on binomial numbers, is much more expensive. In addition, an additional disadvantage of [16] is the vague definition of the adaptation mechanism when enumerating equilibrium combinations. Thus, coding based on binary binomial numbers in [13, 14] is oriented only to the fixed nature of the number of units in binary sequences, which does not provide for an adaptive approach to compression. The same applies to binomial number compression in works [15, 16].

In work [17], a method of binomial compression based on binary binomial numbers for codes with constant weight and general-form binary sequences is presented. The disadvantage of the devised method is that when the parameters of binary sequences go beyond the limits of its rational use, the compression ratio decreases noticeably, and time resources are wasted.

So, the task under consideration is to enable high compression speed of binary information based on binomial numbers under the condition of uncertainty in the characteristics of the binary sequences being compressed.

The solution to this problem is to build an adaptive compression model using binomial numbers, which would take into account the variability of the number of units in binary sequences and be characterized by high speed and undemanding to computing resources. This could prevent a decrease in the efficiency of binomial compression in the case of uncontrolled transitions of the number of units to the area of inefficient use, i.e., the worst obtained compression ratios, which is a topical issue.

3. The aim and objectives of the study

The aim of our study is to increase the speed of binomial compression based on binary binomial numbers by means of an adaptive approach to the choice of the encoding type. In general, this would make it possible to increase or maintain at a given level the productivity of information systems in the event of the appearance and processing of binary sequences that are not compressible.

To achieve this aim, the following objectives were accomplished:

– to build a model of binomial adaptive compression based on binary binomial numbers;

– to evaluate the compression and performance of the model of binomial adaptive compression based on binary binomial numbers.

4. The study materials and methods

The object of our study is the adaptive compression of general-form binary sequences based on binary binomial numbers.

The main hypothesis of the study assumes that the transition to vector coding in the area of inefficient compression based on binary binomial numbers could lead to a reduction in time costs for information transformations.

Assumptions adopted in our study: the number of units in binary sequences can take any values.

Simplifications accepted in the study: equal probability of the original binary sequences being compressed.

The problem statement is as follows. To achieve the goal of our study, it is necessary to minimize the average time T_b of compression of an array of binary n-bit sequences based on binomial numbers

$$T_b = \min_{0 \le k \le n}\left(t' + t''\right) = t' + \min_{M_b} t'',$$

where k is the number of binary n-bit sequences being compressed; t' is the average compression time in the region of efficient use of binomial numbers; t'' is the average compression time in region M_b of the inefficient use of binomial numbers. Region Mb is determined by the values of k that determine the worst compression ratio. According to the above objective function, the problem of minimizing binomial compression is reduced to reducing the amount t'' of time costs of information transformations in region M_b.

Binomial compression is based on the use of binary binomial numbers $X_j = x_1x_2...x_i...x_r$, which are generated by binary (n, k)-binomial number systems, where n and k are the parameters of the number system. In this study, we consider the sets $X[n, k]$ of binary (n, k)-binomial numbers X_j, $X_j \in X[n, k]$; $Y[n, k]$ of binary equilibrium combinations Y_j, which have length n and number k of units, $Y_j \in Y[n,k]$; $A = \{0,1\}^n$ binary n-bit sequences A_j of general form, $A_j \in A$. In this case, the cardinalities of sets $X[n, k]$ and $Y[n, k]$ are the number of combinations C_n^k, and the cardinality of set A is 2^n.

The basis of binomial compression is that length r of numbers X_j can be much less than length n of the equilibrium combinations Y_j being compressed, the structure of which they constitute. The next important idea of compression is to extend the application of binomial numbers X_j to binary sequences A_j of general form. In this case, the fact is taken into consideration that there is a partition of the set

$$A=\bigcup_{0\le k\le n} A_k,\ A_k=Y[n,k], A_c\cap A_d=\varnothing, 0\le c,\ d\le n,$$

into equivalence classes A_k by the number k of binary units, which represent the classes of equilibrium combinations $Y[n, k]$. Then, for the purpose of binomial compression, the number of k units is calculated for each A_j, and the ordered samples A_j are compared $M_j=(k, Y_j)$, where $Y_j=A_j$. Thus, function $M_j=f_w(A_j)$ is used, which defines a mapping of the form

$$f_w\colon A\to M, M=\{(k, Y_j)/\ 0\le k\le n, Y_j\in Y[n,k]\}, M_j\in M. \quad (1)$$

In [17, 18], the mappings $f_b{:}Y[n,k]\to X[n,k]$ and $f_b^{-1}{:}X[n,k]\to Y[n,k]$ denote, respectively, the compression and restoration of equilibrium combinations Y_j based on binary binomial numbers. The mappings $f_{bg}{:}A\to Z$ and $f_{bg}^{-1}{:}Z\to A$, where Z is the set of compressed combinations, represent generalized binomial compressions and restorations of binary sequences $A_j\in A$ of the general form using binary binomial numbers. The set Z consists of combinations $Z_j=(\mathrm{Bin}k, X_j)$ and $Z_j=\mathrm{Bin}k$.

As analysis of the degree of compression reveals, the generalized method f_{bg} has an area of inefficient use, which is a consequence of the need to use Bin$k$ in the record of the compressed combination for its unique restoration. Fig. 1 shows a plot of change in length $L(f_{bg})$ of the compressed combination depending on k units, as well as regions L_b and H_b of the f_{bg} effective use, when the initial A_j has $n=64$ bits. The region of inefficient use f_{bg}, denoted as M_b, for a given n is located between the limits $\alpha_b=7$ and $\beta_b=57$.

Boundaries α_b and β_b of regions L_b and H_b, when the compression ratio exceeds unity, are defined in [18].

Let us consider the vector coding function f_v, which converts the equilibrium combination Y_j into itself: $Y_j=f_v(Y_j)$ or $\mathrm{id}_Y(Y_j)=Y_j$. In order to increase the speed of methods f_{bg} and f_{bg}^{-1}, it is proposed to introduce a procedure for selecting the encoding f_b or f_v for the initial n-bit sequence $A_j\in A=\{0,1\}^n$.

When compressing A_j, it is necessary to use function f_w (1), which matches the original sequence A_j with sample (k, Y_j), where $Y_j=A_j$. Further, if the obtained value k satisfies the system of inequalities of the form

$$\begin{cases}0<k<\alpha_b,\\ \beta_b<k<n,\end{cases} \quad (2)$$

then to compress sequence A_j, f_b coding based on binary binomial numbers is used. If the inequality of the form $\alpha_b\le k\le\beta_b$ is satisfied for value k, then the vector coding method f_v is implemented. In both cases, a binary number of k units Bin$k$ is added to the combinations being coded for unambiguous recovery, i.e., additional coding f_k is performed. If value k satisfies the system of equalities $(k=0)\vee(k=n)$, then the resulting combination will consist only of Bin$k$, i.e., a single coding method f_k is used.

The following theorem gives the properties of the compressive mapping f_{bv} and the technique of its practical implementation ("++" is the concatenation symbol).

Theorem. Any binary sequence $A_j=a_1a_2\ldots a_i\ldots a_n\in A$ can be put into a one-to-one correspondence using the mappings f_{bv} and f_{bv}^{-1} binary combination $Z_j\in Z$ of the following form:

1) if $(0<k<\alpha_b)\vee(\beta_b<k<n)$, then

$$Z_j=\mathrm{Bin}k{+}{+}X_j, \quad (3)$$

where $k=\sum_{i=1}^{n}a_i$ и $X_j\in X[n,k]$;

2) if $\alpha_b\le k\le\beta_b$, then

$$Z_j=\mathrm{Bin}k{+}{+}Y_j, \quad (4)$$

then $Y_j\in Y[n,k]$, $Y_j=A_j$;

3) otherwise, if $(k=0)\vee(k=n)$, then

$$Z_j=\mathrm{Bin}k. \quad (5)$$

Corollary. The mapping $f_{bv}{:}A\to Z$ is bijective.

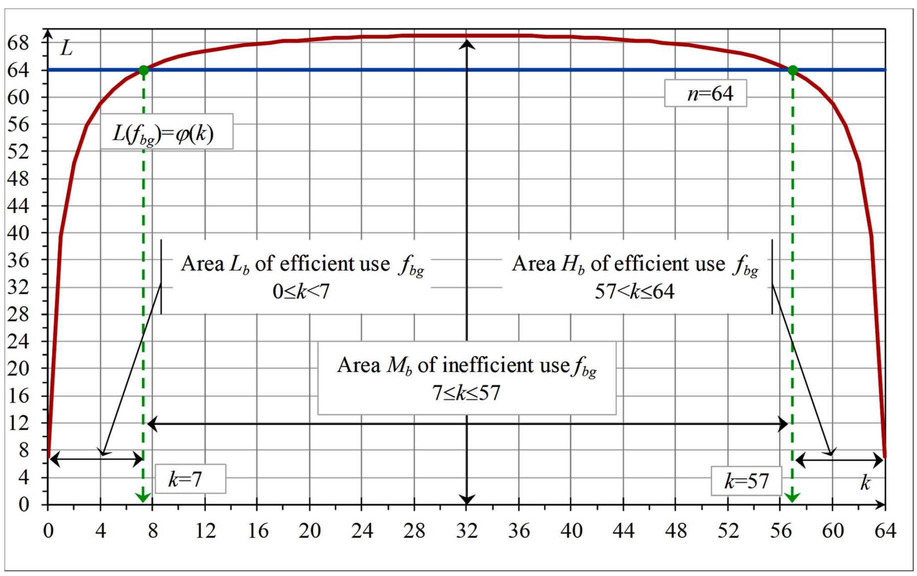

Fig. 1. Plots $L(f_{bg})=\varphi(k)$ and $\varphi(k)=64$ at specified $n=64$

The theorem for the direct f_{bv} and the inverse mapping f_{bv}^{-1} is proved from the opposite of each in three cases. For case (3), we take into account the existence of a one-to-one correspondence between X_j and Y_j [17] and the fact that a unique A_j cannot have two different Bin$k$'s. In the proof of case (4), we take into account the uniqueness of A_j under the condition $Y_j=A_j$, and also, as for case (3), the impossibility for A_j to have two Bin$k$'s. The validity of case (5) is determined by the uniqueness of combination A_j, which has only one binary zero or one.

Combining into one system of equalities (3) to (5) taking into account the conditions of their application, for the resulting combination Z_j we obtain

$$Z_j = \begin{cases} \text{Bin}\,k, & (k=0)\vee(k=n), \\ \text{Bin}\,k{+}{+}X_j, & (0<k<\alpha_b)\vee(\beta_b<k<n), \\ \text{Bin}\,k{+}{+}Y_j, & \alpha_b \le k \le \beta_b. \end{cases} \quad (6)$$

Definition. The mapping $f_{bv}{:}A \to Z$, which is given by the following complex function of the form

$$f_{bv} = \begin{cases} f_k \circ f_w, & (k=0)\vee(k=n), \\ f_k \circ f_b \circ f_w, & (0<k<\alpha_b)\vee(\beta_b<k<n), \\ f_k \circ f_v \circ f_w, & \alpha_b \le k \le \beta_b, \end{cases} \quad (7)$$

is defined as the binomial-vector compression method f_{bv} of binary data.

The set Z is the union of subsets $Z = Z_o \cup Z_b \cup Z_v$, each of which, taking into account $Q = \{\text{Bin}k/0 \le k \le n\}$, is expressed as:

$Z_o = Q\times\varnothing = \{Z_j \,/\, Z_j = \text{Bin}k,\ (k = 0)^\vee(k = n)\}$,

$Z_b = Q\times X[n,k] = \{Z_j \,/\, Z_j = (\text{Bin}k, X_j)$,

$(0 < k < \alpha_b)^\vee(\beta_b < k < n)\}$,

$Z_v = Q\times Y[n,k] = \{Z_j \,/\, Z_j = (\text{Bin}k, Y_j),\ \alpha_b \le k \le \beta_b\}$.

In turn, the inverse mapping $f_{bv}^{-1}{:}\ Z \to A$, which is given by the inverse complex function (taking into account $f_v^{-1} = f_v$)

$$f_{bv}^{-1} = \begin{cases} f_w^{-1} \circ f_k^{-1}, & (k=0)\vee(k=n), \\ f_w^{-1} \circ f_b^{-1} \circ f_k^{-1}, & (0<k<\alpha_b)\vee(\beta_b<k<n), \\ f_w^{-1} \circ f_v \circ f_k^{-1}, & \alpha_b \le k \le \beta_b, \end{cases} \quad (8)$$

is a restoration taking into account the available value k of original binary sequences A_j. In the case $(k = 0)^\vee(k = n)$, the restoration of A_j is carried out based on Bin$k$ only. If $(0 < k < \alpha_b)^\vee(\beta_b < k < n)$, then the reverse transition to sequence A_j is carried out based on Bin$k$ and binary binomial numbers $X_j \in X[n, k]$. In the case $\alpha_b \le k \le \beta_b$, the restoration A_j is an identical mapping of A_j into itself. We term this type of restoration a binomial-vector restoration.

5. Results of investigating the binomial adaptive compression model

5. 1. Implementation of the binomial adaptive compression model based on binary binomial numbers

The modeling of compression process f_{bv} of sequences $A_j = a_1a_2...a_i...a_n$, $A_j \in A = \{0,1\}^n$ is carried out on the basis of the above theorem and the transformation of equilibrium combinations Y_j into binary binomial numbers X_j, considered in [17]. The f_{bv} compression model consists of the following stages:

Stage 1. The number s of bits is determined to represent Bin$k$ of the number k units, $0 \le k \le n$, of the original sequence $A_j = a_1a_2...a_i...a_n$: $s = \lceil\log_2(n + 1)\rceil$.

Stage 2. The number k binary units in the original $A_j = a_1a_2...a_i...a_n$ is calculated, thereby implementing function $f_w(A_j) = (k,Y_j)$. When inequality $\alpha_b \le k \le \beta_b$ is satisfied, the variable $k = \alpha_b$.

Stage 3. The number k of units is converted to its binary form Bin$k$, which consists of s bits.

Stage 4. If number k satisfies the system of equalities $(k = 0)^\vee(k = n)$, then the resulting combination will take the form $Z_j = Bink$, $Z_j \in Z_o$. Otherwise, the transition to the next stage is performed.

Stage 5. If number k satisfies the system of $(0 < k < \alpha_b)^\vee(\beta_b < k < n)$, then the available value n and the calculated value k are parameters of the binary (n,k)-binomial number system. The transition to the following stages is performed to implement the encoding $f_k(f_b(Y_j)) = Z_j$, $Z_j \in Z_o$. Otherwise, the transition to stage 9 is performed to implement the encoding of the form $f_k(f_v(Y_j)) = Z_j$, $Z_j \in Z_v$.

Stage 6. Determine $Y_j = y_1y_2...y_i...y_n$ in the equilibrium combination, which has the number k of units, the value of the last digit y_n.

Stage 7. If $y_n = 0$, then

$X_j = Y_j/00...0 = y_1y_2...y_i...y_{n-1}0/00...0 = x_1x_2...x_i...x_{r-1}1$,

that is, from combination $Y_j = y_1y_2...y_i...y_{n-1}0$ all zero digits are discarded, starting from $y_n = 0$, until the first binary unit $y_r = 1$ appears, which will represent the value of the last digit $x_r = y_r = 1$ of the desired (n,k)-binomial number $X_j = x_1x_2...x_i...x_{r-1}1$. Otherwise

$X_j = Y_j/11...1 = y_1y_2...y_i...y_{n-1}1/11...1 = x_1x_2...x_i...x_{r-1}0$,

i.e., from combination $Y_j = y_1y_2...y_i...y_{n-1}1$ all single digits are discarded, starting from $y_n = 1$, until the appearance of the first binary zero $y_r = 0$, which will represent the value of the last digit $x_r = y_r = 0$ of the desired (n,k)-binomial number $X_j = x_1x_2...x_i...x_{r-1}0$. In both cases, the values of the other digits remain unchanged: $x_1 = y_1$, $x_2 = y_2$,..., $x_{r-1} = y_{r-1}$.

Stage 8. The concatenation $Z_j = Bink{+}{+}X_j$ is performed, i.e., the encoding of the form $f_k(X_j) = Z_j$, for the case $(0 < k < \alpha_b)^\vee(\beta_b < k < n)$, thereby obtaining the resulting combination $Z_j \in Z_b$.

Stage 9. The concatenation $Z_j = Bink{+}{+}Y_j$ is performed, i. e., the encoding of the form $f_k(f_v(Y_j)) = Z_j$, for the case $\alpha_b \le k \le \beta_b$, thereby obtaining the resulting combination $Z_j \in Z_v$.

The considered binomial-vector compression f_{bv}, which is adaptive to the binary sequence $A_j = a_1a_2...a_i...a_n$, being compressed, is a mapping of the form $f_{bv}{:}A \to Z$, which is given by complex function $Z_j = f_{bv}(A_j)$ in the form of (7), where $A_j \in A = \{0,1\}^n$, and $Z_j \in Z$ take the form $Z_j = (\text{Bin}k,X_j)$, $Z_j = (\text{Bin}k,Y_j)$ or $Z_j = \text{Bin}k$, $j = 1,2,...,2^n$. The binomial vector reconstruction f_{bv}^{-1} of compressed images Z_j is an inverse mapping of the form $f_{bv}^{-1}{:}Z \to A$, which is given by the inverse complex function (8).

The practical implementation techniques of mappings f_{bv} and f_{bv}^{-1}, which use switching between encoding f_b based on binary binomial numbers and vector encoding f_v, can be different. The selected approaches to constructing f_{bv} and f_{bv}^{-1}, the coding procedures formed for them, affect the speed and amount of hardware and software costs of the proposed adaptive approach to binomial compression.

The techniques for implementing functions (7) and (8) on the subdomain of definition, when $(k = 0)^\vee(k = n)$ and $(0 < k < \alpha_b)^\vee(\beta_b < k < n)$, for the binomial-vector compression method f_{bv} are similar to the techniques of constructing functions for the generalized compression method f_{bg} [17]. The techniques for implementing functions (7) and (8) on the subdomain of definition, when $\alpha_b \le k \le \beta_b$, are based on fairly simple operations of calculating k units, concatenation and decatenation of Bin$k$, and mapping the original A_j into itself.

The constructed compression model f_{bv} consists of fairly simple operations, the main ones of which are determining the number of bits s for forming the service word Bin$k$, counting the number of k units in the original sequence A_j, checking the compression condition according to (2), forming a binary binomial number X_j and the operation of concatenating the obtained binomial number X_j or the equilibrium combination $Y_j = A_j$ with the Bin$k$ record. It should be emphasized that the proposed adaptive method f_{bv} differs from the generalized compression method f_{bg} [17] only in the presence of the operation of checking the compression condition (2), a simplified and faster procedure for counting the number of k units, and concatenation $Y_j = A_j$ with the service word Bin$k$.

Thus, the practical implementation of adaptive compression f_{bv} based on binary binomial numbers is characterized by being undemanding to computational and hardware resources.

5. 2. Compression and performance evaluation of the binomial adaptive compression model based on binary binomial numbers

Some results of the mappings f_{bv}:$A \rightarrow Z$, f_{bv}^{-1}:$Z \rightarrow A$ for $n = 54$, where $A = \{0,1\}^{54}$, $j = 1,2,\ldots 2^{54}$, are given in Table 1. The systems of conditions (2) for the implementation of coding functions $f_k \circ f_w$, $f_k \circ f_b \circ f_w$ i $f_k \circ f_v \circ f_w$ for this case are given in Table 2. In Table 1 in the lines numbered 1.1, 2.1, 3.1, 4.1 and 5.1, the original binary sequences $A_j \in \{0,1\}^{54}$ are indicated, and in lines 1.2, 2.2, 3.2, 4.2 and 5.2 the corresponding compressed combinations Z_j are shown. Line 1.2 contains the result of encoding $f_k \circ f_w$; lines 2.2, 3.2 and 4.2 – the result of applying function $f_k \circ f_b \circ f_w$, and line 5.2 – the result of applying function $f_k \circ f_v \circ f_w$. The binary representation of the number k units in combinations Z_j is conditionally separated by a dot from the rest of X_j or Y_j. The ratio of the lengths of binary representation A_j and Z_j for this table varies from 0.9 to 9, and their average value for the entire Table 1 is approximately 3.49.

When analyzing time T_b of compression f_{bv}, the number of machine cycles required to count the number of units and subsequently convert the equilibrium combinations into binary binomial numbers is considered. In the case of choosing f_b coding for compression, the machine cycles for calculating the number of units and converting the equilibrium combinations into binomial numbers are summed. In another case, when the choice is made in favor of vector coding f_v, only the number of cycles for determining the number k units is taken into account.

Table 1

Correspondence between certain binaries A_j and Z_j at $n = 54$

1.1	000000000000000000000000000000000000000000000000000000
1.2	000000
2.1	001100100000000000000000000000000000000000000000000000
2.2	000011.0011001
3.1	100000000000000000100000000000100000000000000000000000
3.2	000011.10000000000000000001000000000001
4.1	001111111110011111111011111111111111111111111111111111
4.2	110001.0011111111100111111110
5.1	101111000111001110010101111010111011101111100111011100
5.2	100011.101111000111001110010101111010111011101111100111011100

Table 2

Conditions for applying coding functions $f_k \circ f_w$, $f_k \circ f_b \circ f_w$ and $f_k \circ f_v \circ f_w$ at $n = 54$

$f_k \circ f_w$	$f_k \circ f_b \circ f_w$	$f_k \circ f_v \circ f_w$
$(k = 0) \vee (k = 54)$	$(0 < k < 6) \vee (48 < k < 54)$	$6 \leq k \leq 48$

The calculation of the number of units in binary sequences is carried out based on the technique using operation $S \leftarrow S \wedge (S - 1)$ [19]. The time for calculating the number k units will be exactly k machine cycles. However, for use in the f_{bv} compression method, this calculation technique is modified. First, operation $A_j \leftarrow A_j \wedge (A_j - 1)$ is simultaneously used for the inverse image $\bar{A}_j$, and, second, when the number of units of values α_b or β_b is exceeded, the calculation is interrupted, and the template value α_b is assigned to the value k. This is explained by the fact that the exact value of the number of units is needed only for region $(0 < k < \alpha_b)^{\vee}(\beta_b < k < n)$. As for the compression time of equilibrium combinations Y_j, its calculation is determined by the average value of length L_{cp} of binary binomial numbers [20]. The difference $(n - L_{cp})$ will represent the average number of machine cycles required to discard the bits of combinations Y_j in order to obtain the corresponding binomial numbers X_j.

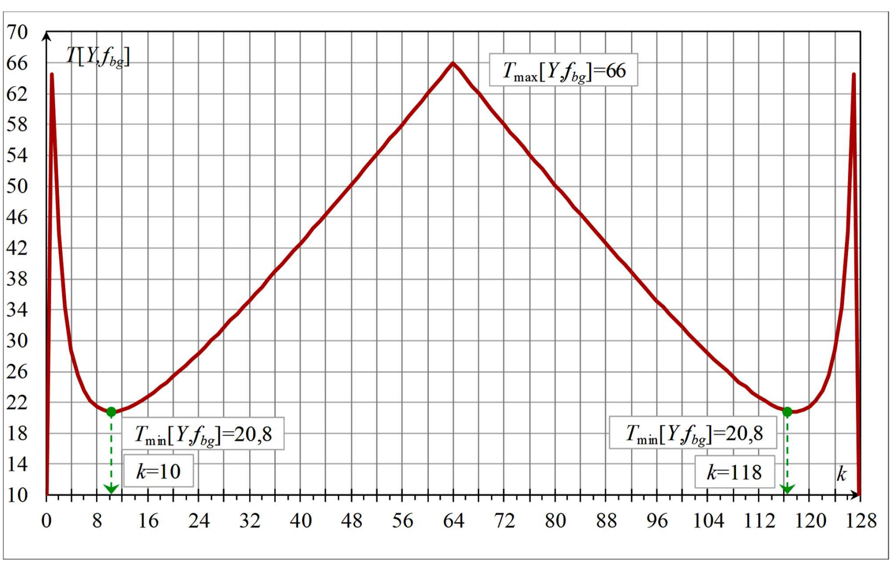

Fig. 2. Plot of average time $T[Y, f_{bg}]$ of the generalized binomial compression f_{bg} at $n = 128$ for different values $0 \leq k \leq 128$

Fig. 2, 3 show the results of analyzing the time of generalized f_{bg} and adaptive binomial-vector f_{bv} compression, respectively.

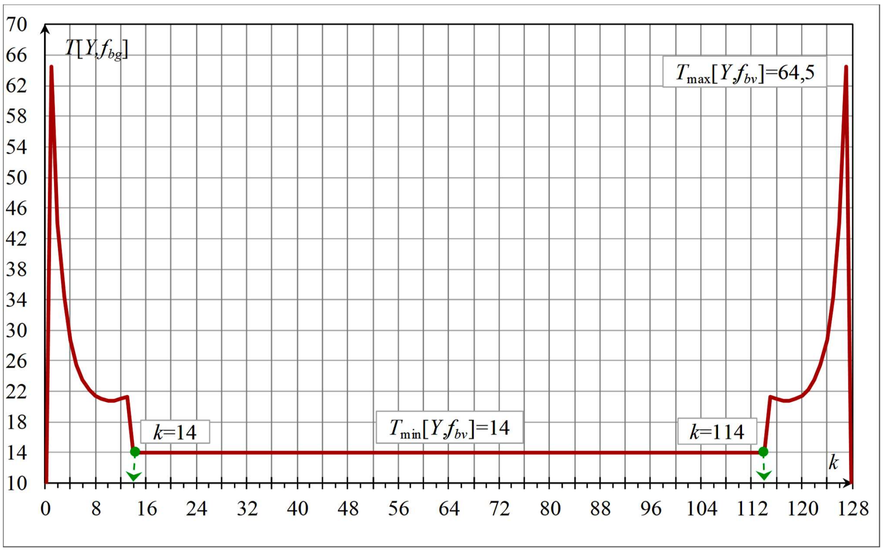

Fig. 3. Plot of average time $T[Y,f_{bv}]$ of adaptive binomial vector compression f_{bv} at $n = 128$ for different values $0 \leq k \leq 128$

Comparing the plots (Figs. 2, 3), it can be observed that in region $14 \leq k \leq 114$, when f_v coding is used, the number of clock cycles used for adaptive f_{bv} compression is generally much smaller than for generalized f_{bg} compression. The value $T[Y,f_{bv}] = 14$ is the machine clock cycle required only to count 14 or more ones (zeros) with a compressed sequence length of $n = 128$. The switching condition to f_b coding based on binomial numbers takes the form $(0 < k < 14)^{\vee}(114 < k < 128)$ at $n = 128$.

6. Discussion of results of investigating the binomial adaptive compression based on binary binomial numbers

Considering the constructed model, adaptive compression based on binary binomial numbers is distinguished by its simplicity of practical implementation in contrast to the models considered in [15, 16]. This becomes possible due to the set of simple operations used in the implementation of the adaptive model of binomial-vector compression. Unlike the model used in the generalized method of compression based on binary binomial numbers in work [17], our model additionally contains only the procedure for selecting the encoding and the operation of concatenating the original combination with the service word Binα_b. But this is precisely what, firstly, allows us to avoid unnecessary time spent when moving to the area of inefficient application of compression (Fig. 1), and, secondly, to simplify the procedure for calculating the number of units in sequences.

Indeed, in stage 5 of our adaptive model, according to the system of inequalities (2), the membership of the processed sequence Y_j within region M_b of the inefficient use of binary binomial numbers is checked. If such membership is established, then immediately, bypassing the time-consuming stage 7, as well as stages 6 and 8, stage 9 is performed. At stage 9, sequence Y_j remains unchanged and the template value Binα_b is added to it. Therefore, on average, instead of $(n - L_{cp})$ machine cycles, where L_{cp} is the average length of binomial numbers [20], in the case of binomial coding for sequences unfavorable for compression, no machine cycle will be spent. In this case, the time spent will consist only of α_b cycles required to count the number of units until the moment when the number k of units in the combination exceeds the compression conditions (2).

On the average time plot $T[Y,f_{bv}]$ of the adaptive binomial-vector compression model at $n = 128$ (Fig. 3) it can be observed for region M_b, when $14 \leq k \leq 114$, that the time spent on processing sequences does not exceed 14 clock cycles. In comparison with the time plot $T[Y,f_{bg}]$ of the generalized binomial compression model (Fig. 2), the running time of the adaptive compression algorithm is reduced by more than 4.7 times for $k = 64$ and by 2.5 times for $k = 32$.

The advantage of our adaptive compression model of binary n-bit sequences is a significant reduction in time spent on the region when the number of units is within the average value $k = n/2$ (Fig. 2, 3). This is explained by the fact that if the compression condition (2) is not met, the counting of units is simplified, the costly operations of forming a binary binomial number are eliminated, and the original combination remains in the same form. This effect very favorably distinguishes the proposed adaptive compression from the generalized method considered in [17].

Thus, if there is an increase in the running time of coding algorithms based on binary binomial numbers in the absence of compression, one should switch to the adaptive binomial-vector method. Such a situation can occur quite often with a variety of data types circulating through communication channels in information systems and networks. In addition, the presence of a built-in mechanism for controlling the correctness of transformations by analyzing the allowed range of values of units in binomial numbers provides an additional impetus for the wider application of binomial compression in practice. The number of units in binary binomial numbers should always be less than or equal to k.

A certain limitation in the application of adaptive binomial-vector compression may be that our estimates of the transformation time relate to equally probable output sequences. In practice, in most cases, information arrays are encountered that consist of combinations with different probabilities of occurrence. But with increasing sequence length, this limitation is eliminated, as their probabilities become closer and closer to each other.

As a disadvantage, it can be noted that for unit values close to limiting α_b and β_b, slightly worse compression ratio results are observed compared to the generalized method. But ultimately, with a larger amount of processed binary data, the average compression ratio for the same time can be higher.

A promising avenue for developing the adaptive compression method based on binary binomial numbers is to study the degree of compression and time characteristics, relying on probabilistic models of binary information sources. In this case, binary sequences have different probabilities of occurrence, which is significantly closer to the practice of processing information arrays. This would allow for a more accurate prediction of the effectiveness of using adaptive binomial-vector compression in information systems.

7. Conclusion

1. As a result of our study, a model of binomial adaptive compression based on binary binomial numbers has been built. The main difference of the binomial adaptive compression based on binary binomial numbers is the procedure for selecting the encoding technique used depending on the number of units in the binary sequence being compressed. The algorithm of model's operation is characterized by fairly simple operations that are easy to implement in practice.

2. Despite the insignificant average decrease in the compression ratio in the transition area from binomial to vector coding, our model makes it possible to significantly, by several times, reduce the time spent on processing non-compressible sequences. In turn, this generally leads to an increase in the speed of binomial adaptive compression based on binary binomial numbers, which corresponds to the purpose of our study. Compared to the generalized binomial compression model, the constructed adaptive model shows a 2.5 and 4.7 times increase in the processing speed of 128-bit sequences, which have 32 and 64 units, respectively.

Conflicts of interest

The authors declare that they have no conflicts of interest in relation to the current study, including financial, personal, authorship, or any other, that could affect the study, as well as the results reported in this paper.

Funding

The study was conducted without financial support.

Data availability

The data will be provided upon reasonable request.

Use of artificial intelligence

The authors confirm that they did not use artificial intelligence technologies when creating the current work.

References

1. Jerzy, S. A. (2013). Information Systems and Data Compression. Springer, 494. https://doi.org/10.1007/978-0-585-27999-2_6
2. Routray, S. K., Javali, A., Sharmila, K. P., Semunigus, W., Pappa, M., Ghosh, A. D. (2020). Lossless Compression Techniques for Low Bandwidth Networks. 2020 3rd International Conference on Intelligent Sustainable Systems (ICISS), 823–828. https://doi.org/10.1109/iciss49785.2020.9315936
3. Salomon, D. (2010). Handbook of Data Compression. Springer, 1383. https://doi.org/10.1007/978-1-84882-903-9
4. Li, Z.-N., Drew, M. S., Liu, J. (2021). Fundamentals of Multimedia. Springer International Publishing Switzerland, 824. https://doi.org/10.1007/978-3-030-62124-7
5. Duda, J., Tahboub, K., Gadgil, N. J., Delp, E. J. (2015). The use of asymmetric numeral systems as an accurate replacement for Huffman coding. 2015 Picture Coding Symposium (PCS), 65–69. https://doi.org/10.1109/pcs.2015.7170048
6. Mrudula, S. T., Srinivasa Murthy, K. E., Prasad, M. N. G. (2022). Optimized Context-Adaptive Binary Arithmetic Coder in Video Compression Standard Without Probability Estimation. Mathematical Modelling of Engineering Problems, 9 (2), 458–462. https://doi.org/10.18280/mmep.090222
7. Stakhov, A. (2016). Fibonacci p-codes and Codes of the "Golden" p-proportions: New Informational and Arithmetical Foundations of Computer Science and Digital Metrology for Mission-Critical Applications. British Journal of Mathematics & Computer Science, 17 (1), 1–49. https://doi.org/10.9734/bjmcs/2016/25969
8. Borysenko, O. A., Horiachev, O. Ye., Serdiuk, V. V., Yermakov, M. S. (2018). Information protection problems of factorial numbers. Ukrainian Scientific Journal of Information Security, 24 (3), 169–174. https://doi.org/10.18372/2225-5036.24.13069
9. Borysenko, O., Matsenko, S., Bobrovs, V. (2021). Binomial Number System. Applied Sciences, 11 (23), 11110. https://doi.org/10.3390/app112311110
10. Luzhetskyi, V. A., Savytska, L. A. (2015). Development and research of adaptive data compression methods based on linear fibonacci form. Eastern-European Journal of Enterprise Technologies, 1 (9 (73)), 16–22. https://doi.org/10.15587/1729-4061.2015.37026
11. Borysenko, A. A. (2004). Bynomyalnii schet. Teoryia y praktyka. Sumi: YTD «Unyversytetskaia knyha», 170. Available at: https://essuir.sumdu.edu.ua/handle/123456789/55161
12. Borisenko, A. A., Kulik, I. A. (2010). Binomialnoe kodirovanie. Sumy: Izd-vo SumGU, 206.
13. Schalkwijk, J. (1972). An algorithm for source coding. IEEE Transactions on Information Theory, 18 (3), 395–399. https://doi.org/10.1109/tit.1972.1054832
14. Cover, T. (1973). Enumerative source encoding. IEEE Transactions on Information Theory, 19 (1), 73–77. https://doi.org/10.1109/tit.1973.1054929

15. Butler, J. T., Sasao, T. (2011). Fast constant weight codeword to index converter. 2011 IEEE 54th International Midwest Symposium on Circuits and Systems (MWSCAS), 1–4. https://doi.org/10.1109/mwscas.2011.6026312
16. Borysenko, O., Matsenko, S., Salgals, T., Spolitis, S., Bobrovs, V. (2022). The Lossless Adaptive Binomial Data Compression Method. Applied Sciences, 12 (19), 9676. https://doi.org/10.3390/app12199676
17. Kulyk, I., Berezhna, O., Shevchenko, M. (2018). Development of data compressing coding methods on basis of binary binomial numbers. Technology Audit and Production Reserves, 2 (2 (46)), 12–18. https://doi.org/10.15587/2312-8372.2019.169897
18. Kulyk, I. A., Novhorodtsev, A. Y., Shevchenko, M. S. (2019). Method for borders estimation of compression on basis of binary binomial numbers. Systemy obrobky informatsii, 2 (157), 57–62. https://doi.org/10.30748/soi.2019.157.07
19. Reingold, M. E., Nievegelt, J., Deo, N. (1977). Combinatorial Algorithms: Theory and Practice. Pearson College Div, 930.
20. Kulyk, I. A. (2004). O srednei dlyne dvoychnikh bynomyalnikh chysel. Visnyk Sumskoho derzhavnoho universytetu. Seriia «Tekhnichni nauky», 12 (71), 106–112.