International Journal of Innovative Computing, Information and Control
Volume **10**, Number **1**, February **2014**

ICIC International ©2014 ISSN 1349-4198
pp. **57–66**

DESCRIPTION AND APPLICATIONS OF BINOMIAL NUMERAL SYSTEMS

Alexei A. Borisenko[1], Vyacheslav V. Kalashnikov[2,3,4]
Nataliya I. Kalashnykova[5] and Denis V. Gutenko[1]

[1]Department of Electronics and Computing
[4]Department of Applied Mathematics
Sumy State University
Rimsky-Korsakov St. 2, Sumy 40007, Ukraine
electron@sumdu.edu.ua; kalash@itesm.mx

[2]Systems and Industrial Engineering
Tecnológico de Monterrey (ITESM), Campus Monterrey
Ave. Eugenio Garza Sada 2501 Sur, Monterrey, Nuevo León 64849, Mexico

[3]Experimental Economics
Central Economics and Mathematics Institute (CEMI)
Russian Academy of Sciences
Nakhimovsky pr. 47, Moscow 117418, Russian Federation

[5]Applied Mathematics
Department of Physics and Mathematics (FCFM)
Universidad Autónoma de Nuevo León (UANL)
Ave. Universidad S/N, San Nicolás de los Garza, Nuevo León 66450, Mexico
nkalash2009@gmail.com

Received December 2012; revised June 2013

Abstract. *We develop a new class of positional numeral systems, namely the binomial ones, which form a subclass of generalized positional numeral systems (GPNS). The binomial systems have wide range of applications in the information transmission, processing, and storage due to their error-detection capabilities. In this paper, the binomial numeral systems are well-defined, their prefix and compactness properties are established. Algorithms of generating binomial coding words (non-uniform and uniform) are presented, as well as an enhanced procedure of construction of constant weight Boolean combinations based upon the non-uniform binomial coding words. The correctness of this procedure is established.*

Keywords: Generalized positional numeral systems, Binomial numeral systems, Constant weight codes

1. **Introduction.** Positional numeral systems are widely used in computing. More complicated numeral systems, in which the register's weight need not be equal to the power of the system's base (for example, in the binary or decimal system), have not been thoroughly studied yet. Such generalized positional numeral systems (GPNS) may have quite useful properties, like being noise-proof, easy in generating permutations (*see* [1-3]).

When combinatorial objects are generated and numerated, researchers use them to develop special methods for each individual problem, which can be characterized as a principal drawback of such an approach (*cf.* [4,5]). Therefore, a universal algorithm solving these problems at both the theoretical and practical levels would be very helpful. We propose a possible solution method based upon the GPNS. In particular, in this paper, a binomial numeral system is considered, which generates combinatorial objects making use of constant weight codes (*cf.* [6]). The total number of coding words in such codes is determined by binomial coefficients (*cf.* [7]).

It is worthwhile to develop digital devices using a GPNS and completing mainly logical and the simplest arithmetical operations with integer numbers, because these operations are realized by the GPNS in the most efficient way. Certain parts of such specialized devices, e.g., noise-proof counters, registers, are of interest for the universal computers as well [8,9].

One of the important problems arising while storing and transmitting information is its compression, for example by the optimal coding based on the Shannon-Fano and Huffman codes [10,11].

Next, problems of combinatorial optimization are of special importance. In the most general form, these problems may even not have an objective function but stated in some preference terms. Such problems are usually solved by an exhaustive search, or when it is impossible, by random search procedures [2]. In both cases, the GPNS can provide many efficient ways of generating the combinatorial objects in order to find a path to an optimal solution.

Therefore, the generalized positional numeral systems (GPNS) propose a unified approach allowing one to solve efficiently a series of practical problems of various natures. As an example of such a GPNS, our paper presents a binary binomial numeral system. The latter is characterized with the use of binomial coefficients as weights of the binary digits (*cf.* [12-14]).

The binomial numeral systems and therewith generated uniform binomial numbers are an absolutely new result in the area of generalized positional numeral systems. The first seminal manuscript giving birth to the binomial systems is the paper published by Borisenko [15] more than 30 years ago. The latter was the first landmark in this area, although many unsuccessful attempts had been made before this one (e.g., in the book by Oberman [9]). Binomial numeral systems were also mentioned (without their detailed description or analysis) in the famous monograph by Knuth [16]. The nearest concept to the non-uniform binomial numbers were the codes developed in [17]; however, none of the above-mentioned attempts had led to either creation of the theory of binomial numeral systems or development of algorithms to synthesize the binomial numbers. Although, in principle, it is possible to construct non-uniform binomial numbers by partitioning the binomial coefficients into smaller equivalence classes making use of the main property of combinations. The latter approach was exploited in the above-mentioned monograph [17], too, but its practical value is very much restricted without the theory of binomial numbers.

However, the practical importance of the Theory of Binomial Numbers is difficult to overestimate for the development of various counting devices. Many such devices (about 20 in total) have been registered as inventions (*see*, e.g., [8]).

The binomial numbers are very widely implemented in practice. For example, Table 2 in Section 4 provides the increasing uniform binomial numbers generated by the algorithm described below. This algorithm served as the base of developing, modeling and producing a counter performed as a microchip on a crystal matrix and registered as the invention (*cf.*, [18]).

The same counter is part of the invention [19] exploited in practice to count *constant weight codes*. A code has a constant weight if each combination in this code has the same number of units. As an example of a constant weight code, consider the code containing six combinations with two units each: 0011, 0101, 0110, 1001, 1010, and 1100 (*see*, also, the constant weight code in Table 3 enjoying 4 units in each of the 15 combinations).

The principal result of the paper is the additional analysis of the binomial numeral systems, and the improved algorithm of creating the binomial numbers. Based upon this algorithm, we develop a synthesis of binomial numbers containing no more than one unit (alternatively, no more than one zero).

The main novelty of the presented paper as compared with the authors' previous works are the following ones:

(1) The previous works (*cf.*, the List of References) did not describe *all* the algorithms of generation of binomial numbers.

(2) The algorithms proposed and studied in the authors' previous papers needed to be improved, since they were not intended for the computer implementation and generation of constant weight codes.

(3) The special kinds of binomial codes, the combinations of which contain no more than 1 unit or, vice versa, 1 zero, were not paid attention in the authors' previous publications. However, the latter is extremely important as these binomial numbers are very similar to those of the unitary code, which, due to its combinations being unprecedently noise-proof, becomes more and more popular in practical applications. Now the binomial counters transform into the shift registers with only one unit or, alternatively, only one zero. Such combinations boast with the remarkably noise stability, because the appearance of more than one unit (or, alternatively, zero) in a binomial number immediately leads to the detection of an error and, in special cases, its correction.

(4) As the binomial combinations are numbers, a possibility of creation of fast-running noise-proof adders (or, summers) clearly arises. Hence, new circuits for computers are also possible to develop. Such devices have already been tested in practice and confirmed their working abilities. The speed of such devices is almost maximum possible, since the transitions are completely excluded.

(5) Finally, in contrast to the authors' previous publications, the theoretical part has been also improved and brought to a mathematically rigorous form: here, all the necessary results have been established as the corresponding lemmas and theorems.

The rest of the paper is arranged as follows. In Section 2, we define the principal structure of the binomial calculus system. Section 3 presents the main results establishing the key properties of the binomial systems, namely, the prefix and the compactness properties. Next, Section 4 deals with the algorithms to generate and numerate binomial combinations of various lengths (non-uniform codes), the constant length (uniform codes), and the constant weight combinations. Section 5 describes two interesting particular cases of codes with only one unit ($k = 1$) and only one zero ($k = n - 1$) and the properties of such binomial numbers. Conclusion, acknowledgement and the reference list complete the paper.

2. **Binomial Systems.**

Now we describe one of the GPNS, namely the binomial system with the binomial weights and the binary alphabet $\{0, 1\}$ (*cf.* [12-14]).

In a k-binomial system with n registers ($k < n$), the quantitative equivalent QA_i of a code combination $A_i = (a_{j-1}, a_{j-2}, \ldots, a_0)$, $i = 0, 1, \ldots, P - 1$, with $P = C_n^k$, where $j = j(i)$ is the combination's length, is defined as follows:

$$QA_i = a_{j-1}C_{n-1}^{k-q_j} + \cdots + a_\ell C_{n-j+\ell}^{k-q_{\ell+1}} + \cdots + a_0 C_{n-j}^{k-q_1}, \tag{1}$$

where the following conditions must hold: either

$$\begin{cases} q_0 = k, \\ j < n; \\ a_0 = 1, \end{cases} \tag{2}$$

or

$$\begin{cases} n-k=j-q_0, \\ q_0<k; \\ a_0=0. \end{cases} \tag{3}$$

Here q_0 is the quantity of units (ones) in the binomial number, P is the range of the system, j is the quantity of registers (positions) in the binomial number (or, its length), $\ell = 0, 1, \ldots, j-1$ is the register's ordinal number, and q_ℓ is the sum of the digits occupying the registers $(j-1)$ through ℓ, inclusively, i.e.,

$$q_\ell = \sum_{i=\ell}^{j-1} a_i, \tag{4}$$

with $q_j = 0$.

A positional numeral system must be finite, effective, and well-defined. However, it is not enough for a generalized positional system. In addition, it has to be a prefix code system, i.e., with the "prefix property": there is no valid code word in the system that is a prefix (start) of any other valid code word in the set. With a prefix code, a receiver can identify each word without requiring a special marker between words. The generalized positional numeral system should be also continuous, which means that for any number s from the system's range (except for the maximal number), there exists a combination, whose quantitative equivalent is equal to $(s+1)$. All these properties of the binomial numeral systems will be established in the next section.

3. **The Binomial System is Finite, Effective, Prefix, and Well-Defined.** Formula (1) shows that the binomial numeral system is *finite* and *effective*, because there exists a numeration algorithm, which, after a finite of number of steps, converts the coding combination A_i into its quantitative equivalent QA_i. Now the following theorem establishes the *prefix property* of the binomial numeral system. Its proof can be found in [12].

Theorem 3.1. [12] *The k-binomial numeral system with n registers (where $k < n$) is a prefix code system.*

To show that the binomial system is well-defined, that is, two distinct coding combinations cannot be equivalent to the same numerical value, we established the following result (again, *see*, [12]).

Theorem 3.2. [12] *The k-binomial system with n registers (where $k < n$) is well-defined.*

Theorems 3.1 and 3.2 imply the following important corollary, which proves the compactness of the binomial numeral systems.

Corollary 3.1. *The k-binomial system with n registers $(k < n)$ is compact, that is, its range is complete and covers all the integers between 0 and $(C_n^k - 1)$.*

Remark 3.1. *It is straightforward that for the k-binomial calculus system with n registers, the range parameter P is equal to C_n^k.*

4. **Algorithms Generating Binomial Combinations.** Table 1 contains the binomial combinations and their quantitative equivalents for the k-binomial system with n registers, where $n = 6$ and $k = 4$. They are generated by the following algorithm:

Step 1. An initial combination A_0 consisting of $(n-k)$ zeros is composed and referred to as a *keyword*.

Step 2. The digit 1 is put into the right end register, and zero is added to the right side of it.

TABLE 1. Binomial coding combinations of non-constant length (non-uniform code)

Binomial word	Its quantitative equivalent
00	$0C_5^4 + 0C_4^4 = 0$
010	$0C_5^4 + 1C_4^4 + 0C_3^3 = 1$
0110	$0C_5^4 + 1C_4^4 + 1C_3^3 + 0C_2^2 = 2$
01110	$0C_5^4 + 1C_4^4 + 1C_3^3 + 1C_2^2 + 0C_1^1 = 3$
01111	$0C_5^4 + 1C_4^4 + 1C_3^3 + 1C_2^2 + 1C_1^1 = 4$
100	$1C_5^4 + 0C_4^3 + 0C_3^3 = 5$
1010	$1C_5^4 + 0C_4^3 + 1C_3^3 + 0C_2^2 = 6$
10110	$1C_5^4 + 0C_4^3 + 1C_3^3 + 1C_2^2 + 0C_1^1 = 7$
10111	$1C_5^4 + 0C_4^3 + 1C_3^3 + 1C_2^2 + 1C_1^1 = 8$
1100	$1C_5^4 + 1C_4^3 + 0C_3^2 + 0C_2^2 = 9$
11010	$1C_5^4 + 1C_4^3 + 0C_3^2 + 1C_2^2 + 0C_1^1 = 10$
11011	$1C_5^4 + 1C_4^3 + 0C_3^2 + 1C_2^2 + 1C_1^1 = 11$
11100	$1C_5^4 + 1C_4^3 + 1C_3^2 + 0C_2^1 + 0C_1^1 = 12$
11101	$1C_5^4 + 1C_4^3 + 1C_3^2 + 0C_2^1 + 1C_1^1 = 13$
1111	$1C_5^4 + 1C_4^3 + 1C_3^2 + 1C_2^1 = 14$

Step 3. Step 2 is repeated while the number of 1's in the coding word is less than $k-1$. If the number of 1's is equal to $k-1$, then go to Step 4.

Step 4. If the right end position contains zero, we replace it with 1. Go to Step 5.

Step 5. Check the number of 1's in the coding combination: if it equals k but the 1's do not occupy the first k registers counted from left to right, go to Step 6. Otherwise, i.e., if the 1's occupy the first k registers counted from left to right, then **STOP**: all the combinations have been generated.

Step 6. Update the keyword A_0 by putting 1 as a prefix before the beginning of the keyword (i.e., its left end). If the total number of 1's in the keyword is less than k, go to Step 2.

The binomial systems find various important applications, in which the following useful features are exploited: (i) the binomial systems are noise-proof in the information transmission, processing, and storage; (ii) they are able to search, generate and numerate coding combinations with a constant weight; (iii) they can be used to construct noise-proof digital devices.

To detect errors with the aid of binomial coding combinations, they should be completed with zeros to obtain uniform $(n-1)$-digital binomial coding words given in Table 2.

The main tokens of errors in a binomial coding combination are either the number of 1's being greater than k, or the number of zeros exceeding $(n-k)$. The principal feature of the binomial noise-proof code is its ability to detect errors while processing information. This feature allows one to arrange the throughout control in the information processing channels involving the digital devices.

Generation of binomial coding combinations with a constant weight. Next, Table 3 shows a transformation of binomial coding combinations to coding words with a constant weight: this is done by adding (to the right end) either 1's if the binomial combination contains $(n-k)$ zeros, or adding zeros if the combination comprises k digits 1, until the combination's length reaches n.

Each binomial combination (column 2 of Table 3) has the corresponding combination with the constant weight (column 3 of Table 3), hence the former is a compressed image

TABLE 2. Binomial coding combinations of a constant length (uniform code)

NN	Binomial word	Binomial uniform word
0	00	00000
1	010	01000
2	0110	01100
3	01110	01110
4	01111	01111
5	100	10000
6	1010	10100
7	10110	10110
8	10111	10111
9	1100	11000
10	11010	11010
11	11011	11011
12	11100	11100
13	11101	11101
14	1111	11110

TABLE 3. Binomial coding combinations of a constant weight

NN	Binomial word	Binomial constant weight word
0	00	001111
1	010	010111
2	0110	011011
3	01110	011101
4	01111	011110
5	100	100111
6	1010	101011
7	10110	101101
8	10111	101110
9	1100	110011
10	11010	110101
11	11011	110110
12	11100	111001
13	11101	111010
14	1111	111100

of the latter. If one needs to label a combination with the constant weight by some traditional numeral system numbers (e.g., decimals of column 1 in Table 3), Formula (1) has to be used. In the latter case, a compression of binomial numbers is completed.

Algorithms of search and generation of binomial combinations and those with constant weights can be also found in [14]. Now we describe one of modifications of such algorithms and prove its efficiency as follows. This method is based upon the fact that the range of binomial numbers of length n and with parameter k ($k < n$) coincides with the range of the constant weight coding combinations with k units among n registers. Therefore, the formal description of the algorithm is as follows:

Step 1. Select an arbitrary non-uniform binomial coding combination.

Step 2. If the coding combination ends with the digit 1, then put zeros into all registers up to the right end (register n), which is considered as auxiliary. The thus obtained combination ending with 0 will be the combination with the constant weight.

Step 3. If the coding combination ends with the digit 0, then set units (ones) into all registers up to the right end (register n, or the auxiliary register). The thus created combination ending with 1 will be the combination with the constant weight.

Step 4. Verify that the thus obtained combination is indeed with the constant weight by counting the total number of ones (units). If this number is k then the combination is indeed a desired one. Select another non-uniform binomial coding combination and go to Step 2. If all the non-uniform binomial coding combinations have been already selected, then **STOP**: all the constant weight combinations of this range have been generated.

The above algorithm generates the complete range of the corresponding combinations of the constant weight, which is confirmed by the following theorem.

Theorem 4.1. *With the aid of the above algorithm, for every non-uniform binomial combination of length n with parameter k ($k < n$), one obtains the unique corresponding coding combination with (the constant) weight k and length n.*

5. Binomial Numbers for $k = 1$ and $k = n - 1$.

As an important particular case, let us consider two special values of parameter k, namely, $k = 1$ and $k = n - 1$.

5.1. Binomial numbers for $k = 1$.

In the first of these two cases, when $k = 1$, formulas (1)-(3) reduce to the following:

$$QA_i = a_{j-1}C_{n-1}^{1-q_j} + \cdots + a_\ell C_{n-j+\ell}^{1-q_{\ell+1}} + \cdots + a_0 C_{n-j}^{1-q_1}, \tag{5}$$

where the following conditions must hold: either

$$\begin{cases} q_0 = 1, \\ j < n; \\ a_0 = 1, \end{cases} \tag{6}$$

or

$$\begin{cases} n - 1 = j - q_0, \\ q_0 < 1; \\ a_0 = 0. \end{cases} \tag{7}$$

Equation $q_0 = 1$ from (6) immediately implies that the total number of units in the binomial number is equal to 1. Moreover, since $a_0 = 1$ by (6), then 0 is the unique position having the digit 1, i.e., $a_0 = 1$, while all the other positions $a_{j-1}, \ldots, a_1$ will be filled in with zero. In other words, then formula (5) has the form:

$$QA_i = 0 \cdot C_{n-1}^{1-q_j} + \cdots + 0 \cdot C_{n-j+\ell}^{1-q_{\ell+1}} + \cdots + 1 \cdot C_{n-j}^{1-q_1}. \tag{8}$$

Now because $q_1 = 0$ (being the sum of all digits in the positions peceding position 0) is evidently equal to zero, then finally the quantity function yields (in case $k = 1$) the following rule:

$$QA_i = C_{n-j}^{1} = n - j. \tag{9}$$

The latter can be interpreted as follows: in the binomial system with $k = 1$, the quantitative equivalent QA_i of a code combination A_i coincides with $n - j$, where $j = j(i)$ is the length of this code combination.

In addition, system (7) directly imples that $q_0 = 0$, that is, the number of units in such a combination is zero. In other words, constraints (7) generate only one code combination with $n-1$ positions, each of which is occupied by zero. Therefore, the quantitative equivalent of such a combination is zero, too:

$$QA_i = 0 \cdot C_{n-1}^{1-q_j} + \dots + 0 \cdot C_{n-j+\ell}^{1-q_{\ell+1}} + \dots + 0 \cdot C_{n-j}^{1-q_1} = 0. \tag{10}$$

To resume, the function of quantitative equivalent for the binomial system with $k=1$ has the following simplified form:

$$QA_i = \begin{cases} 0, & \text{if } q_0 = 0; \\ n-j, & \text{if } q_0 = 1. \end{cases} \tag{11}$$

Example 5.1. *For $n=5$ and $k=1$, one has the following binomial system.*

TABLE 4. Quantitative equivalents for binomial coding combinations for $k=1$

QA_i	Binomial word	Binomial uniform word	Binomial constant weight word
0	0000	0000	00001
1	0001	0001	00010
2	001	0010	00100
4	01	0100	01000
5	1	1000	10000

5.2. **Binomial numbers for $\boldsymbol{k} = n-1$.** Now consider another extremum, that is, the binomial system with $k=n-1$. For this particular case, system (2) transforms into the following system of equations:

$$\begin{cases} q_0 = n-1, \\ j = n-1; \\ a_0 = 1, \end{cases} \tag{12}$$

which evidently allows only one code combination of length $n-1$ containing ones in all positions. Therefore, the quantitative equivalent to this combination is calculated by the formula

$$QA_i = 1 \cdot C_{n-1}^{1-q_j} + \dots + 1 \cdot C_{n-j+\ell}^{1-q_{\ell+1}} + \dots + 1 \cdot C_{n-j}^{1-q_1} = n-1. \tag{13}$$

In its turn, constraints (3) are reduced to the form

$$\begin{cases} q_0 = j-1, \\ 1 \le q_0 \le n-1; \\ a_0 = 0. \end{cases} \tag{14}$$

The latter means that all code combinations satisfying (14) will always comprise $j-1$ positions filled in by ones; hence, only one position will accept the zero value. According to (14), this is the rightest (number zero) position. Therefore, the quantitative equivalent values function will be of the kind:

$$QA_i = 1 \cdot C_{n-1}^{1-q_j} + \dots + 1 \cdot C_{n-j+\ell}^{1-q_{\ell+1}} + \dots + 0 \cdot C_{n-j}^{1-q_1} = 1 \cdot (j-1) = j-1. \tag{15}$$

Example 5.2. *For $n=5$ and $k=n-1$, one obtains the following binomial system.*

Table 5. Quantitative equivalents for binomial coding combinations for $k = n - 1$

QA_i	Binomial word	Binomial uniform word	Binomial constant weight word
0	0	0000	01111
1	10	1000	10111
2	110	1100	11011
4	1110	1110	11101
5	1111	1111	11110

6. **Conclusion.** The binomial numbers are a useful tool for development of new bases of fast and noise-proof digital devices, as well as in telecommunication systems with the aim of construction of the coding and decoding algorithms. The systems using binomial numbers are characterized by the enhanced transmission speed, noise-and-secrecy stability and reliability.

The principal result of the paper is the additional analysis of the binomial numeral systems, and the improved algorithm of creating the binomial numbers. Based upon this algorithm, we develop a synthesis of binomial numbers containing no more than one unit (alternatively, no more than one zero).

In more detail, in this paper, we have described the error-detecting binomial numeral systems capable of transmitting, processing and storing information. The systems can also generate and numerate combinatorial configurations, for example, coding words with a constant weight, as well as compositions, combinations with repetitions, etc. The binomial systems and codes studied in Section 5 with $k = 1$ and $k = n - 1$ are especially redundant and hence extremely noise-proof. Moreover, they boast with very easy tools for the detection and (sometimes even) correction of possible transmission errors. In addition, counters, registors, summators and other digital devices based upon the binomial systems and codes have enhanced speed of performance, which is often even more important than the property of being noise-proof.

Apart from that, the binomial systems can be applied to produce efficient information compression and defense. The latter is the goal of our further research, as well as the new and promising concept of matrix binomial numbers.

Acknowledgement. The research activity of the second author was financially supported by the R&D Department (Cátedra de Investigación) CAT-174 of the Instituto Tecnológico y de Estudios Superiores de Monterrey (ITESM), Campus Monterrey, and by the SEP-CONACYT project CB-2008-01-106664, Mexico. Also, the work of the third author was supported by the National Counsel of Science and Technology (CONACyT) of Mexico as part of the project CB-2011-01-169765; PROMEP 103.5/11/4330, and PAICYT 464-10.

REFERENCES

[1] A. P. Stakhov, *Golden Section Codes*, Radio i Sviaz', Moscow, 1984 (in Russian).
[2] Y. G. Stoyan and V. Z. Sokolovsky, *Solution of Some Extremum Problems by the Method of Contracting Neighborhoods*, Naukova Dumka, Kiev, 1980 (in Russian).
[3] A. A. Borisenko, Numerical coding based upon combinatorial calculus systems, *Proc. of International Conference on Theoretical and Applied Aspects of Program Systems Development*, Kiev, Ukraine, pp.98-104, 2010 (in Russian).
[4] V. A. Amel'kin, *Numerical Coding Methods*, Nauka, Novosibirsk, 1986 (in Russian).
[5] E. M. Reingold, J. Nievergelt and N. Deo, *Combinatorial Algorithms: Theory and Practice*, Prentice Hall, Englewood Cliffs, New Jersey, 1977.

[6] A. A. Borisenko and I. A. Kulik, *Binomial Coding*, Sumy State University, Sumy, Ukraine, 2010 (in Russian).
[7] J. A. Anderson, *Discrete Mathematics with Combinatorics*, Prentice Hall, Upper Saddle River, New Jersey, 2001.
[8] A. A. Borisenko, *Binomial Calculus and Counters*, Sumy State University, Sumy, Ukraine, 2008 (in Russian).
[9] R. M. M. Oberman, *Counting and Counters*, Wiley & Sons, Hoboken, New Jersey, 1981.
[10] V. P. Tsymbal, *Information and Coding Theory*, Vyshaya Shkola, Kiev, 1977 (in Russian).
[11] I. V. Kuz'min and V. A. Kedrus, *A Basic Theory of Information and Coding*, Vyshaya Shkola, Kiev, 1977 (in Russian).
[12] A. A. Borisenko, V. V. Kalashnikov and N. I. Kalashnykova, Binomial calculus: Advantages and prospects, *ICIC Express Letters*, vol.2, no.1, pp.123-130, 2008.
[13] A. A. Borisenko, *Introduction to the Theory of Binomial Calculus*, Universitetskaya Kniga, Sumy, Ukraine, 2004 (in Russian).
[14] A. A. Borisenko, *Binomial Calculus: Theory and Applications*, Universitetskaya Kniga, Sumy, Ukraine, 2004 (in Russian).
[15] A. A. Borisenko, A numeral system wth the binomial base and binary alphabet, *Deposited at VINITI*, Moscow, no.90982, 1982 (in Russian).
[16] D. E. Knuth, *Art of Computer Programming, Vol. 2: Seminumerical Algorithms*, 3rd Edition, Addison-Wesley Professional, Boston, 1997.
[17] P. P. Kuvyrkov and F. E. Temnikov, *Combinatorial Systems*, Energy Editorial (Energia), Moscow, 1975 (in Russian).
[18] A. A. Borisenko, I. L. Puz'ko and L. A. Stetsenko, *Counter of Impulses*, USSR Author's Certificate of Invention, No. 1077054.
[19] A. A. Borisenko and G. V. Kuno, *Counter of Impulses*, USSR Author's Certificate of Invention, No. 1150761.