sciendo
Control and Cybernetics
vol. **50** (2021) No. 2
pages: 335-346
DOI: 10.2478/candc-2021-0017

Coding theory based on balancing polynomials*

by

Bandhu Prasad

Department of Mathematics, Kandi Raj College
Kandi - 742137, India
bandhu_iit@rediffmail.com

Abstract: In this paper, we introduce a $Q_2^n(x)$ matrix, whose elements are balancing polynomials, and develop a new coding and decoding method following from the $Q_2^n(x)$ matrix. We establish the relations between the code matrix elements, error detection and correction for this coding theory.

Keywords: balancing numbers, k-balancing numbers, balancing polynomials

1. Introduction

Balancing numbers, n, and balancers, r, have been originally defined as the solution of the Diophantine equation $1 + 2 + 3 + \cdots + (n-1) = (n+1) + (n+2) + (n+3) + \cdots + (n+r)$. The first few balancing numbers are 1, 6, 35, 204, 1189, with respective balancers 0, 2, 14, 84, 492. Balancing numbers and their generalization have been studied in Belbachair and Szalay (2014), Berczes, Liptai and Pink (2010), Panda and Rout (2013), Prasad (2018), Behera and Panda (1999), Liptai et al. (2009), as well as Ray (2014). Multiplication of the balancing numbers was defined by Szakacs (2011). A positive integer, n, is called a multiplying balancing number if $1 \cdot 2 \cdot 3 \cdot \cdots \cdot (n-1) = (n+1) \cdot (n+2) \cdot (n+3) \cdot \cdots \cdot (n+r)$ for some positive integer, r, which is known as multiplying balancer, corresponding to the multiplying balancing number, n. These numbers have been studied broadly during the last twenty years. The latest results on balancing polynomials, powers of balancing polynomials and some consequences for Fibonacci sums, as well as identities for the generalized balancing numbers are provided in Frontczak (2019a, b, c).

The present article is organized as follows. In Section 2.1, k-balancing numbers and balancing numbers are defined. In Section 2.2, balancing polynomials,

*Submitted: January 2020; Accepted: January 2021.

Binet formula for balancing polynomials and relations between polynomials are considered. In Section 3.1, a new balancing polynomial matrix $Q_2(x)$ and its properties are described. In Section 3.2, balancing polynomial coding and decoding method is presented. In Sections 3.3 and 3.4, the determinant of the code matrix and the relation between code matrix elements are established, respectively. Then, in Section 3.5, error detection and correction are presented. In Sections 3.6 and 3.7, comparison of the balancing coding method with the classical coding method and role of polynomials in cryptographic protection are described, respectively. Finally, in Sections 4 and 5, conclusions and open problem are provided, respectively.

2. Preliminaries

2.1. k-balancing numbers

For any positive number k, k-balancing numbers, $\{B_{k,n}\}_{n=0}^{\infty}$, are defined by the recurrence relation

$$B_{k,n+1} = 6kB_{k,n} - B_{k,n-1}, \quad \text{for} \quad n \geq 1 \tag{1}$$

with initial conditions

$$B_{k,0} = 0, B_{k,1} = 1.$$

For $k = 1$, (1) gives the sequence of balancing numbers and the characteristic equation for (1) is given by

$$\alpha^2 - 6k\alpha + 1 = 0, \tag{2}$$

where the roots are $\alpha_1 = 3k + \sqrt{9k^2 - 1}$ and $\alpha_2 = 3k - \sqrt{9k^2 - 1}$.

For $k = 1$, $\alpha_1 = 3+2\sqrt{2}$, $\alpha_2 = 3-2\sqrt{2}$, α_1, α_2 are conjugate with respect to each other and $\lambda = 3 + 2\sqrt{2}$ is known as balancing constant or balancing mean, see Ray (2014).

2.2. Balancing polynomials

The balancing polynomials, $B_n(x)$, are the extension of the k-balancing numbers, $B_{k,n}$, and are defined by

$$B_{n+1}(x) = \begin{cases} 1, & \text{for } n = 0, \\ 6x, & \text{for } n = 1, \\ 6xB_n(x) - B_{n-1}(x) & \text{for } n > 1. \end{cases}$$

The first few balancing polynomials are $B_0(x) = 0$, $B_1(x) = 1$, $B_2(x) = 6x$, $B_3(x) = 36x^2 - 1$, $B_4(x) = 216x^2 - 12x$, $B_5(x) = 1296x^4 - 108x^2 + 1$.

THEOREM 1 *If $B_n(x)$ denotes the n^{th} balancing polynomial, then*

$$\lim_{n\longrightarrow\infty} \frac{B_{n+1}(x)}{B_n(x)} = 3x + \sqrt{9x^2 - 1}.$$

PROOF: Let $\lim_{n\longrightarrow\infty} \frac{B_{n+1}(x)}{B_n(x)} = \lambda(x)$.

Then $\lambda(x) = \lim_{n\longrightarrow\infty}[\frac{6xB_n(x)-B_{n-1}(x)}{B_n(x)}] = 6x - \frac{1}{\lambda(x)}$.

Therefore, $\lambda^2(x) - 6x\lambda(x) + 1 = 0$. The roots thereof are $\lambda(x) = 3x + \sqrt{9x^2 - 1}$ and $\lambda^{-1}(x) = 3x - \sqrt{9x^2 - 1}$. □

The Binet form for the balancing polynomials is

$$B_n(x) = \frac{\lambda^n(x) - \lambda^{-n}(x)}{\lambda(x) - \lambda^{-1}(x)},$$

where $\lambda(x) = 3x + \sqrt{9x^2 - 1}$ and $\lambda^{-1}(x) = 3x - \sqrt{9x^2 - 1}$.

The relation $B_n(-x) = (-1)^{n+1}B_n(x)$ follows from $\lambda(-x) = -\lambda^{-1}(x)$ and $\lambda^{-1}(-x) = -\lambda(x)$ and we also find the relation $B_{n-r}(x)B_{n+r}(x) - B_n^2(x) = -B_r^2(x)$ for integers n, r.

In this paper, we define a new balancing polynomial matrix, $Q_2^n(x)$, whose elements are balancing polynomials, and we establish the relations between code matrix elements, error detection and corrections. In 2018, Prasad (2018) published a paper, based on balancing numbers. The present paper is based on balancing polynomials and the suitable initial conditions, so that the here introduced $Q_2^n(x)$ matrix is applicable for the proposed coding and decoding method.

3. Main results

3.1. Balancing polynomial matrix, $Q_2(x)$, and its properties

In this section, we define the new balancing polynomial matrices, $Q_2(x)$ and $Q_2^n(x)$:

$$Q_2(x) = \begin{pmatrix} 6x & -1 \\ 1 & 0 \end{pmatrix} = \begin{pmatrix} B_2(x) & -B_1(x) \\ B_1(x) & B_0(x) \end{pmatrix}, \tag{3}$$

$$Q_2^n(x) = \begin{pmatrix} B_{n+1}(x) & -B_n(x) \\ B_n(x) & -B_{n-1}(x) \end{pmatrix}. \tag{4}$$

Proof: We will prove the above by mathematical induction.
For $n = 1$,

$$Q_2(x) = \begin{pmatrix} B_2(x) & -B_1(x) \\ B_1(x) & B_0(x) \end{pmatrix} = \begin{pmatrix} 6x & -1 \\ 1 & 0 \end{pmatrix}.$$

which is true for $n = 1$.

Suppose (4) is true for integer $n = k$, then

$$Q_2^k(x) = \begin{pmatrix} B_{k+1}(x) & -B_k(x) \\ B_k(x) & -B_{k-1}(x) \end{pmatrix}.$$

Now, we can write

$$Q_2^{k+1}(x) = (Q_2^k(x))(Q_2(x)) = \begin{pmatrix} B_{k+1}(x) & -B_k(x) \\ B_k(x) & -B_{k-1}(x) \end{pmatrix} \begin{pmatrix} 6x & -1 \\ 1 & 0 \end{pmatrix}$$
$$= \begin{pmatrix} B_{k+2}(x) & -B_{k+1}(x) \\ B_{k+1}(x) & -B_k(x) \end{pmatrix}.$$

Hence, by induction, we can write

$$Q_2^n(x) = \begin{pmatrix} B_{n+1}(x) & -B_n(x) \\ B_n(x) & -B_{n-1}(x) \end{pmatrix},$$

for any value of x, the determinant of $Q_2(x)$, Det $Q_2(x) = 1$ and determinant of $Q_2^n(x)$, Det $Q_2^n(x) = 1 \Rrightarrow B_n^2(x) - B_{n+1}(x)B_{n-1}(x) = 1$, which is known as Cassini formula for the balancing polynomials, see Frontczak (2019b). □

3.2. Balancing polynomial coding and decoding method

In this paper, we introduce a new coding theory, to which we refer as balancing polynomial coding and decoding method. In this method, we represent the message in the form of nonsingular square matrix, M, of order 2, and we represent the balancing polynomial matrix, $Q_2^n(x)$, of order 2 as coding matrix and its inverse matrix, $(Q_2^n(x))^{-1}$, as a decoding matrix. We represent a transformation $M \times Q_2^n(x) = E$ as balancing polynomial coding and a transformation $E \times (Q_2^n(x))^{-1} = M$ as balancing polynomial decoding. We represent the matrix E as code matrix.

3.2.1. Example of the balancing polynomial coding and decoding method

Let us represent the initial message in the form of the nonsingular square matrix, M, of order 2:

$$M = \begin{pmatrix} m_1 & m_2 \\ m_3 & m_4 \end{pmatrix}. \tag{5}$$

Assume that all elements of the matrix are positive integers, i.e., m_1, m_2, m_3, $m_4 > 0$. Let us now do the selection for any value of n for a $Q_2^n(x)$ matrix in (4). We simply write for $n = 2$

$$Q_2^2(x) = \begin{pmatrix} B_3(x) & -B_2(x) \\ B_2(x) & -B_1(x) \end{pmatrix} = \begin{pmatrix} 36x^2 - 1 & -6x \\ 6x & -1 \end{pmatrix}. \tag{6}$$

Then, the inverse of $Q_2^2(x)$ is given by

$$(Q_2^2(x))^{-1} = \begin{pmatrix} -B_1(x) & B_2(x) \\ -B_2(x) & B_3(x) \end{pmatrix} = \begin{pmatrix} -1 & 6x \\ -6x & 36x^2 - 1 \end{pmatrix}. \tag{7}$$

Given the above, the coding of the message (5) consists in the multiplication of the initial matrix (6), that is

$$M \times Q_2^2(x) = \begin{pmatrix} m_1 & m_2 \\ m_3 & m_4 \end{pmatrix} \begin{pmatrix} 36x^2 - 1 & -6x \\ 6x & -1 \end{pmatrix} = \begin{pmatrix} e_1 & e_2 \\ e_3 & e_4 \end{pmatrix} = E, \tag{8}$$

where $e_1 = 36x^2m_1 + 6m_2x - m_1,\ e_2 = -6m_1x - m_2,\ e_3 = 36x^2m_3 + 6m_4x - m_3,\ e_4 = -6m_3x - m_4$.

Then the code message, $E = e_1,\ e_2,\ e_3,\ e_4$, is sent to a channel. The decoding of the code message, E, given with (7), is performed in the following way,

$$\begin{pmatrix} e_1 & e_2 \\ e_3 & e_4 \end{pmatrix} \begin{pmatrix} -1 & 6x \\ -6x & 36x^2 - 1 \end{pmatrix} = \begin{pmatrix} -e_1 - 6e_2x & 6e_1x + 36e_2x^2 - e_2 \\ -e_3 - 6e_4x & 6e_3x + 36e_4x^2 - e_4 \end{pmatrix} = \begin{pmatrix} m_1 & m_2 \\ m_3 & m_4 \end{pmatrix} = M.$$

3.3. Determinant of the code matrix E

The code matrix, E, is defined by the formula $E = M \times Q_2^n(x)$. According to the matrix theory, see Stakhov (1977, 2006), we have

$$DetE = Det(M \times Q_2^n(x)) = DetM \times DetQ_2^n(x) = DetM \times (1)^n = DetM. \tag{9}$$

3.4. Relations between the code matrix elements

We can write the code matrix, E, and the initial message, M, as the following expressions

$$E = M \times Q_2^n(x) = \begin{pmatrix} m_1 & m_2 \\ m_3 & m_4 \end{pmatrix} \begin{pmatrix} B_{n+1}(x) & -B_n(x) \\ B_n(x) & -B_{n-1}(x) \end{pmatrix} = \begin{pmatrix} e_1 & e_2 \\ e_3 & e_4 \end{pmatrix}$$

and

$$M = E \times (Q_2^n(x))^{-1} = \begin{pmatrix} e_1 & e_2 \\ e_3 & e_4 \end{pmatrix} \begin{pmatrix} -B_{n-1}(x) & B_n(x) \\ -B_n(x) & B_{n+1}(x) \end{pmatrix} = \begin{pmatrix} -e_1 B_{n-1}(x) - e_2 B_n(x) & e_1 B_n(x) + e_2 B_{n+1}(x) \\ -e_3 B_{n-1}(x) - e_4 B_n(x) & e_3 B_n(x) + e_4 B_{n+1}(x) \end{pmatrix}. \tag{10}$$

Since $m_1,\ m_2,\ m_3,\ m_4$ are positive integers, we have

$$m_1 = -e_1 B_{n-1}(x) - e_2 B_n(x) > 0, \tag{11}$$

$$m_2 = e_1 B_n(x) + e_2 B_{n+1}(x) > 0, \tag{12}$$

$$m_3 = -e_3 B_{n-1}(x) - e_4 B_n(x) > 0, \tag{13}$$

$$m_4 = e_3 B_n(x) + e_4 B_{n+1}(x) > 0. \tag{14}$$

From (11) and (12) we get

$$\frac{B_n(x)}{B_{n-1}(x)} < -\frac{e_1}{e_2} < \frac{B_{n+1}(x)}{B_n(x)}.$$

Now, from (13) and (14) we get

$$\frac{B_n(x)}{B_{n-1}(x)} < -\frac{e_3}{e_4} < \frac{B_{n+1}(x)}{B_n(x)}.$$

Therefore, for large n we obtain

$$-\frac{e_1}{e_2} \approx \lambda(x), \quad -\frac{e_3}{e_4} \approx \lambda(x) \text{ where } \lambda(x) = 3x + \sqrt{9x^2 - 1}. \tag{15}$$

3.5. Error detection and correction

3.5.1. Error detection

One of the main aims of the coding theory is the detection and correction of errors arising in the code message, E, under the influence of noise in the communication channel. The most important idea is to use the property of determinant of the matrix as the verification criterion of the transmitted message, E. Let the initial message, M, be given by

$$M = \begin{pmatrix} m_1 & m_2 \\ m_3 & m_4 \end{pmatrix}, \tag{16}$$

where all elements m_1, m_2, m_3, m_4 of the matrix M are positive integers.

Now, the determinant of M is

$$Det\ M = m_1 m_4 - m_2 m_3 \tag{17}$$

and the code message, E,

$$E = (M \times (Q_2^n(x)). \tag{18}$$

So,

$$\begin{aligned} &Det\ E = Det\ (M \times Q_2^n(x)) = \\ &Det\ M \times Det\ Q_2^n(x) = Det\ M \times (1)^n = Det\ M. \end{aligned} \tag{19}$$

This shows that the determinant of the initial message, M, is connected with the determinant of the code message, E, by a definite relation. The identity (19) specifies the new method of error detection, based on the application of the $Q_2^n(x)$ matrix. The gist of the method consists in that the sender calculates the determinant of the initial message, M, represents it in the matrix form (16) and sends it to the channel after the code message, E, (18). The receiver calculates the determinant of the code message, E, (18), and compares it with the determinant of the initial message, M, (16), received from the channel. If this comparison corresponds to (19), this means that the code message, E, (18), is correct and the receiver can decode the code message, E, (18), otherwise the code message, E, (18), is not correct. Error detection is the first step in communication of messages.

3.5.2. Error correction

The possibility of restoration of the code message, E, can be realized by using the property of the $Q_2^n(x)$ matrix. For the selection of $n = 2$, the $Q_2^n(x)$ matrix will be

$$Q_2^2(x) = \begin{pmatrix} B_3(x) & -B_2(x) \\ B_2(x) & -B_1(x) \end{pmatrix} = \begin{pmatrix} 36x^2 - 1 & -6x \\ 6x & -1 \end{pmatrix}. \tag{20}$$

Then, the coding of the message (16) consists in the multiplication of the initial matrix (20), that is

$$M \times Q_2^2(x) = \begin{pmatrix} m_1 & m_2 \\ m_3 & m_4 \end{pmatrix} \begin{pmatrix} 36x^2 - 1 & -6x \\ 6x & -1 \end{pmatrix} =$$
$$\begin{pmatrix} 36x^2 m_1 + 6m_2 x - m_1 & -6m_1 x - m_2 \\ 36x^2 m_3 + 6m_4 x - m_3 & -6m_3 x - m_4 \end{pmatrix} = \begin{pmatrix} e_1 & e_2 \\ e_3 & e_4 \end{pmatrix} = E, \quad (21)$$

where $e_1 = 36x^2 m_1 + 6m_2 x - m_1$, $e_2 = -6m_1 x - m_2$, $e_3 = 36x^2 m_3 + 6m_4 x - m_3$, $e_4 = -6m_3 x - m_4$.

After constructing the code matrix, E, we calculate the determinant of the initial matrix, M, (16). The determinant is sent to the communication channel after the code message, $E = e_1, e_2, e_3, e_4$. Assume that the communication channel has the special means for detecting the error in each of the elements e_1, e_2, e_3, e_4 of the code message, E. Assume that the first element e_1 of E is received with an error. Then, we can represent the code message in the matrix form as

$$E' = \begin{pmatrix} u & e_2 \\ e_3 & e_4 \end{pmatrix}, \quad (22)$$

where u is the corrupted element of the code message, E, but the rest of the matrix entries must be correct and equal to the following:

$$e_2 = -6m_1 x - m_2; e_3 = 36x^2 m_3 + 6m_4 x - m_3; e_4 = -6m_3 x - m_4. \quad (23)$$

Then, according to the properties of the coding method, we can write the following equation for calculation of u:

$$\begin{aligned} & ue_4 - e_2 e_3 = \\ & u(-6m_3 x - m_4) - (-6m_1 x - m_2)(36x^2 m_3 + 6m_4 x - m_3) = \\ & (m_1 m_4 - m_2 m_3). \end{aligned} \quad (24)$$

From (24), we get

$$u = 36x^2 m_1 + 6m_2 x - m_1. \quad (25)$$

Comparing the calculated value (25) with the entry e_1 of the code matrix, E, given with (21), we conclude that $u = e_1$. Thus, we have restored the coded message, E, using the property of determinant of the $Q_2^n(x)$ matrix. But in the real situation, we usually do not know what element of the code message is corrupted. In this case, we suppose different hypotheses about the possible corrupted elements and then we test these hypotheses. However, we have one more condition for the elements of the code matrix, E, that all its elements are integers. Our first hypothesis is that we have the case of a single error in the code matrix, E, received from the communication channel. It is clear that there are four variants of the single errors in the code matrix, E:

$$(a)\begin{pmatrix} u & e_2 \\ e_3 & e_4 \end{pmatrix} \quad (b)\begin{pmatrix} e_1 & v \\ e_3 & e_4 \end{pmatrix} \quad (c)\begin{pmatrix} e_1 & e_2 \\ w & e_4 \end{pmatrix} \quad (d)\begin{pmatrix} e_1 & e_2 \\ e_3 & z \end{pmatrix}, \tag{26}$$

where u, v, w, z are the corrupted elements. In this case we can check different hypotheses (26). For checking the hypotheses (a), (b), (c), (d) we can write the following algebraic equations based on the checking relation (19):

$$ue_4 - e_2e_3 = Det\ M(\text{a possible single error is in the element } e_1), \tag{27}$$

$$e_1e_4 - ve_3 = Det\ M(\text{a possible single error is in the element } e_2), \tag{28}$$

$$e_1e_4 - e_2w = Det\ M(\text{a possible single error is in the element } e_3), \tag{29}$$

$$e_1z - e_2e_3 = Det\ M(\text{a possible single error is in the element } e_4). \tag{30}$$

It follows from (27)-(30) that there are four variants for the calculation of the possible single errors.

$$u = \frac{Det\ M + e_2e_3}{e_4}, \tag{31}$$

$$v = \frac{-Det\ M + e_1e_4}{e_3}, \tag{32}$$

$$w = \frac{-Det\ M + e_1e_4}{e_2}, \tag{33}$$

$$z = \frac{Det\ M + e_2e_3}{e_1}. \tag{34}$$

The formulae (31)-(34) give four possible variants of a single error, but we have to choose the correct variant only among the cases of the integer solutions u, v, w, z; besides, we have to choose such solutions, which satisfy the additional checking relations (15). If calculations by the formulae (31)-(34) do not give an integer result, we have to conclude that our hypothesis about the single error is incorrect or we have an error in the checking element Det M. For the latter case we can use the approximate equalities (15) for checking the correctness of the code matrix, E. By analogy, we can check all the hypotheses of a double error in the code matrix. Let us consider the following case of a double error in the code matrix, E

$$\begin{pmatrix} u & v \\ e_3 & e_4 \end{pmatrix}, \tag{35}$$

where u, v are the corrupted elements of the code message. Using the first checking relation (19) we can write the following algebraic equation for the matrix (35):

$$ue_4 - ve_3 = Det\ M. \tag{36}$$

However, according to the second checking relation (15), there is the following relation between u and v

$$u \approx -\lambda(x)v. \tag{37}$$

It is important to emphasize that (36) is a Diophantine equation. As a Diophantine equation, (36) has many solutions, and we have to choose such solutions u, v, which satisfy the checking relation (37). By analogy, one may prove that by using the checking relations (15), (19) by means of solution of the Diophantine equation similar to (36) we can correct all possible double errors in the code matrix. However, we can show that by using such approach there is also a possibility of correcting all the possible triple errors in the code matrix E, for example $\begin{pmatrix} u & v \\ w & e_4 \end{pmatrix}$ etc., where u, v, w are the corrupted elements.

Thus, our method of error correction is based on the verification of different hypotheses about errors in the code matrix by using the checking relations (15), (19), and by using the fact that the elements of the code matrix are integers. If all our solutions do not lead to integer outcomes, conform to (15) and (19), it means that the checking element Det M is erroneous or we have the case of fourfold error in the code matrix, E and we have to reject the code matrix, E, as defective and not corrigible. Our method allows for correction 14 cases among $({}^4C_1+{}^4C_2+{}^4C_3+{}^4C_4)=2^4-1=15$ cases. It means that correction ability of the method is $\frac{14}{15}=0.9333=93.33\%$.

3.6. Comparison of the balancing coding method with the classical coding method

The balancing coding method is based on the matrix approach, which possesses many peculiarities and advantages in comparison to the classical (algebraic) coding method. The use of matrix theory for designing new error-correction codes is the first peculiarity of this coding method. The large information units, in particular matrix elements, are objects of detection and correction of errors in this coding method. There is no theoretical restriction for the value of the numbers that can be the matrix elements, whereas in algebraic coding theory there are very small information elements, bits and their combinations are the objects of detection and correction. This coding method has very high correction ability in comparison to classical coding.

3.7. The role of polynomials in cryptographic protection

Polynomials have, definitely, a prominent position in mathematics. Gradually, polynomials have become unavoidable also in cryptography. They are very important in encryption and decryption procedures for security purposes. The security and complexity are increasing as the degrees of polynomials are increasing.

4. Conclusion

The balancing coding method is the main application of the $Q_2^n(x)$ matrix, whose elements are balancing polynomials. This coding method reduces to matrix multiplication, a well-known algebraic operation, which is realized very well in modern computers. The main practical peculiarity of this method is that the large information units, in particular, matrix elements, are objects of detection and correction of errors. The elements of the initial matrix, M, and therefore the elements of the code matrix, E, can be the numbers of unlimited value. It means that, theoretically, this coding method allows to correct the numbers of unlimited value. The correction ability of this method is 93.33%. The correction ability and detection ability of this coding method are very high in comparison with the algebraic coding methods.

5. Open problem

There is a problem of defining a $Q_k^n(x)$ matrix of order $k \geq 3$, whose elements are balancing polynomials. This matrix will be useful for establishing a code matrix, error detection and correction for more general situations.

References

Berczes, A., Liptai, K. and Pink, I. (2010) On generalized balancing sequences. *Fibonacci Quart.* **48**(2), 121–128.

Behera, A. and Panda, G. K. (1999) On the square roots of triangular numbers. *Fibonacci Quart.* **37**(2), 98–105.

Belbachair, H. and Szalay, L. (2014) Balancing in direction (1,-1) in Pascals triangle. *Armenian J. Math.* **6**(1), 32–40.

Frontczak, R. (2019a) Identities for generalized balancing numbers. *Notes on Number Theory and Discrete Mathematics*, **25**, 169–180.

Frontczak, R. (2019b) On balancing polynomials. *Appl. Math. Sci.* **13**, 57–66.

Frontczak, R. (2019c) Powers of balancing polynomials and some consequences for Fibonaccci sums. *Int. J. Math. Anal.* **13**, 109–115.

Liptai, K., Luca, F., Pinter, A. and Szalay, L. (2009) *Generalized balancing numbers. Indagationes Math.* **20**, 87–100.

Panda, G. K. and Rout, S. S. (2013) Gap balancing numbers. *Fibonacci Quart.* **51**(3), 239–248.

Prasad, B. (2018) Coding theory on balancing numbers. *Int. J. Open Problems Compt. Math.* **11**(4), 73–85.

Ray, P. K. (2014) Balancing sequences of matrices with application to algebra of balancing numbers. *Notes on Number Theory and Discrete Mathematics* **20**(1), 49–58.

Stakhov, A. P. (1977) *Introduction into algorithm measurement theory.* Soviet Radio, Moscow (In Russian).

Stakhov, A. P. (2006) Fibonacci matrices, a generalization of the cassini formula and a new coding theory. *Chaos, Solitons and Fractals* **30**, 56–66.

Szakacs, T. (2011) Multiplying balancing numbers. *Acta Univ. Sapientiae Math.* **3**(1), 90–96.