Multicomponent Network Codes for a Channel with Random Linear Transformations and Packet Errors

Alexander Shishkin

Multicomponent Network Codes for a Channel with Random Linear Transformations and Packet Errors

Alexander Shishkin

Moscow Institute of Physics and Technology (State University)
sisoid@frtk.ru

Abstract. This work is related to new multicomponent network codes for a channel with random linear transformations and packet errors. New code construction is a generalization of Gabidulin-Pilipchuk codes, and similarly uses rank codes with restrictions as subcodes. Usage of a greedy algorithm, when selecting the code components, allows us to increase code cardinality compared with the previously known constructions. We provide the conditions under which the described multicomponent code can correct both erasures and packet errors in the channel. Finally, we present an efficient algorithm of decoding both types of errors and give some examples.

1 Introduction

Consider $\mathbb{K}_q$ - a finite field of q elements. Construct an n-dimensional vector space, $\mathbb{K}_q^n$, with elements from $\mathbb{K}_q$, and denote the set of all its subspaces as $\mathcal{A}(n)$. A subspace $\mathcal{V}$ of dimension $k \leq n$ from $\mathcal{A}(n)$ is the set of q^k vectors with length n and elements from $\mathbb{K}_q$.

The subspace $\mathcal{V}$ may be considered as a linear span over k vectors from $\mathbb{K}_q^n$, or, equivalently, over some $k \times n$ matrix with the elements from $\mathbb{K}_q$. Thus, every $k \times n$ matrix V, with the elements from $\mathbb{K}_q$, uniquely determines some k-dimensional subspace $\mathcal{V}$ from $\mathcal{A}(n)$. However, every subspace from $\mathcal{A}(n)$ may have several generating matrices: multiplying matrix V by a nonsingular $k \times k$ matrix, T, gives a $k \times n$ matrix $\tilde{V} = TV$ that generates the same subspace as the matrix V.

In connection with this, it is convenient to specify the subspaces with the matrices in reduced row echelon form, to which an arbitrary matrix of size $k \times n$ can be brought using the Gaussian elimination procedure. The reduced row echelon form of a matrix is characterized by the following properties:

- The first nonzero element in every row equals 1_q. It is called the leading element of the row.
- Every leading element is the only nonzero element in its column.
- The leading element of the next line is always located to the right of the leading element of the previous line.

– There are no restrictions on the remaining matrix elements. They are called free elements.

One can establish a bijection between k-dimensional subspaces from $\mathcal{A}(n)$ and $k \times n$ matrices over $\mathbb{K}_q$ in reduced row echelon form.

The location of the leading elements of a $k \times n$ matrix in reduced row echelon form can be described by a multi-index $\mathcal{I} = \{i_1, i_2, \ldots, i_k\}$, defined as the set of integers corresponding to the column numbers in which the leading elements are located. Another way to describe the location of leading elements is a vector-index v of length n, which contains ones in those components that correspond to the columns with leading elements, and zeros in the other components.

For example, the 5×12 matrix U from equation 1, which is in reduced row echelon form, can be described by the multi-index $U_{\mathcal{I}} = \{2, 3, 7, 11, 12\}$ or vector-index $U_v = (011000100011)$.

$$U = \begin{pmatrix} 0\,1\,0\,*\,*\,*\,0\,*\,*\,*\,0\,0 \\ 0\,0\,1\,*\,*\,*\,0\,*\,*\,*\,0\,0 \\ 0\,0\,0\,0\,0\,0\,1\,*\,*\,*\,0\,0 \\ 0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,1\,0 \\ 0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,1 \end{pmatrix}. \tag{1}$$

Note that symbols '*' correspond to the free elements of the matrix U.

2 Subspace codes

Between any two subspaces from $\mathcal{A}(n)$, the Grassmannian distance can be introduced as:

$$d_{sub}(\mathcal{U}, \mathcal{V}) = \dim(\mathcal{U} \cup \mathcal{V}) - \dim(\mathcal{U} \cap \mathcal{V}). \tag{2}$$

Let $\mathcal{U}$ be a subspace of dimension m with the generating matrix U, and $\mathcal{V}$ be a subspace of dimension k with the generating matrix V. Then the Grassmannian distance between $\mathcal{U}$ and $\mathcal{V}$ can be calculated as follows:

$$d_{sub}(\mathcal{U}, \mathcal{V}) = 2\mathrm{rank}\left(\begin{bmatrix} U \\ V \end{bmatrix}\right) - m - k. \tag{3}$$

When both subspaces have the same dimension ($m = k$) the distance d_{sub} will be even for any $\mathcal{U}$ and $\mathcal{V}$.

Using the Grassmannian distance, the concept of subspace codes can be introduced. Subspace $[n, M, d_{sub}, k]$-code is a set of k-dimensional subspaces from $\mathcal{A}(n)$, which contains M elements, and the distance between any two subspaces is not less than d_{sub}. There is the Johnson-type upper bound on the cardinality of the subspace code [1]:

$$M \leq \left\lfloor \frac{q^n - 1}{q^k - 1} \left\lfloor \frac{q^{n-1} - 1}{q^{k-1} - 1} \cdots \left\lfloor \frac{q^{n-k+\delta} - 1}{q^\delta - 1} \right\rfloor \cdots \right\rfloor \right\rfloor, \tag{4}$$

where $\delta = d_{sub}/2$.

Subspace codes are widely used in network coding theory. An important task in the construction of subspace codes is to maximize the cardinality M at a fixed correcting capability of a code, determined by the subspace distance d_{sub}.

Grassmannian distance is closely related to the rank distance used in the construction of rank-metric codes [2]. Let $\mathcal{U}$ and $\mathcal{V}$ be subspaces of dimension k with generating matrices U and V, which are in reduced row echelon form. To illustrate the connection of subspace codes with the rank codes we introduce the concept of a submatrix of the leading elements. For the matrix U, in reduced row echelon form, the set of its columns containing the leading elements is called the submatrix of the leading elements and is denoted by U_l. The set of all the remaining columns is called the submatrix of the free elements and is denoted by U_f. Note that U_f contains all the free elements of the matrix U, but not each of its elements is free. Further, there are two possibilities.

If multi-indices $U_{\mathcal{I}}$ and $V_{\mathcal{I}}$ are equal, then the leading elements matrices of U and V are equal too, and the Grassmannian distance between $\mathcal{U}$ and $\mathcal{V}$ can be written as follows:

$$d_{sub}(\mathcal{U},\mathcal{V}) = 2\text{rank}\left(\begin{bmatrix} U \\ V \end{bmatrix}\right) - 2k = 2\text{rank}\left(\begin{bmatrix} U_l|U_f \\ V_l|V_f \end{bmatrix}\right) - 2k = \\ = 2k + 2\text{rank}([U_f - V_f]) - 2k = 2\text{rank}([U_f - V_f]). \quad (5)$$

In the intermediate calculations we used the fact that for the subspaces of dimension k the rank of the leading elements matrix equals k. Note, that in this case, the final expression for the subspace distance between $\mathcal{U}$ and $\mathcal{V}$ corresponds exactly to the doubled rank-metric distance between their free elements matrices U_f and V_f.

Now consider a case when the multi-indices $U_{\mathcal{I}}$ and $V_{\mathcal{I}}$ are not equal. In this case the number of linearly independent columns in the matrix $\begin{bmatrix} U \\ V \end{bmatrix}$ is not less than the number of nonzero elements in the vector $U_v \vee V_v$. Therefore,

$$d_{sub}(\mathcal{U},\mathcal{V}) \geq 2(U_v \vee V_v) - 2k = 2(2k - (U_v \wedge V_v)) - 2k = \\ = 2k - 2(U_v \wedge V_v) = (U_v \oplus V_v) = d_{ham}(U_v; V_v), \quad (6)$$

where d_{ham} is the Hamming distance between the binary vector-indices U_v and V_v, corresponding to the subspaces $\mathcal{U}$ and $\mathcal{V}$ respectively.

Thus, for the subspaces within the same multi-index, the problem of building the code with the minimal subspace distance d_{sub} reduces to the construction of a rank-metric code with the minimal distance $d_{rank} = d_{sub}/2$. For subspaces with different multi-indices, the Grassmannian distance is not less than the Hamming distance between the corresponding vector-indices.

Now the task of building subspace codes can be decomposed into two subtasks. First, we need to build a binary code of vector-indices with a minimal Hamming distance $d_{ham} = d_{sub}$. Then, for subspaces within the same vector-index, we need to build a rank-metric code with the minimal distance $d_{rank} = d_{sub}/2$, using the free elements of the subspace generating matrix.

Codewords within the same multi-index are called the code component. A subspace code, that contains codewords from more than one code components, is called the multicomponent subspace code.

3 Constructions of the subspace codes

Historically, the first construction of the subspace codes for random network coding is the Silva-Koetter-Kshishang code (SKK-code) [3]. Codewords of this code are subspaces, for which the generating matrices in reduced row echelon form have the following representation:

$$U = \left[I_k \; U_f \right], \tag{7}$$

where I_k is the square identity matrix of order k and the submatrix of the free elements U_f is the $k \times (n-k)$ rank code matrix over the field $GF(q)$.

Equation 8 illustrates the form of SKK-code generating matrices for $n = 9$ and $k = 3$. Let d_{rank} be the minimal distance of the rank subcode. Then the minimal subspace distance of the SKK-code is equal to $d_{sub} = 2d_{rank}$. This code construction is often referred to as a lifting construction of the rank-metric code. Note that the SKK-code has only one code component, so it is not the multicomponent code.

$$U = \begin{pmatrix} 1\,0\,0 * * * * * * \\ 0\,1\,0 * * * * * * \\ 0\,0\,1 * * * * * * \end{pmatrix}. \tag{8}$$

Gabidulin-Bossert codes with zero prefix [4] have a multicomponent structure. The first component is an SKK-code with the $k \times (n-k)$ rank code submatrix. The second component has an all-zero matrix as a prefix of the whole code matrix. The number of rows in this all-zero matrix is equal to k, while the number of columns is equal to $\delta = d_{sub}/2$. The identity matrix of order k follows the zero prefix, and the leftmost positions in the network code matrix are occupied by $k \times (n-k-\delta)$ rank code matrix. The third code component contains two all-zero matrices as a prefix, and so on for the remaining components.

The number of components in the Gabidulin-Bossert code depends on the code length. The last component contains only the zero prefix and the identity matrix when $(n-k)$ is divisible by δ. Equation 9 represents 3 components of the Gabidulin-Bossert code for $n = 9$, $k = 3$ and $d_{sub} = 6$.

$$U_1 = \begin{pmatrix} 1\,0\,0 * * * * * * \\ 0\,1\,0 * * * * * * \\ 0\,0\,1 * * * * * * \end{pmatrix}, \quad U_2 = \begin{pmatrix} 0\,0\,0\,1\,0\,0 * * * \\ 0\,0\,0\,0\,1\,0 * * * \\ 0\,0\,0\,0\,0\,1 * * * \end{pmatrix},$$

$$U_3 = \begin{pmatrix} 0\,0\,0\,0\,0\,0\,1\,0\,0 \\ 0\,0\,0\,0\,0\,0\,0\,1\,0 \\ 0\,0\,0\,0\,0\,0\,0\,0\,1 \end{pmatrix}. \tag{9}$$

In the works [5, 6] Etzion and Silberstein introduced a greedy lexicographic search of the network code vector-indices among the set of all binary vectors

with the length n and Hamming weight k. Their approach gives considerable freedom to choose parameters of the network code, and avoids the use of complex combinatorial circuits. However, this approach does not guarantee that all subcodes in the multicomponent construction are linear rank codes and therefore the decoding algorithm may be nontrivial.

There is another method to construct multicomponent codes. It uses incomplete balanced block designs [7]. These block designs define multi-indices which in turn specify the location of the columns with the leading elements in the network code matrix. Each multi-index corresponds to one code component. The first code component is an SKK-code and has the greatest cardinality among all the other components. Free elements of the code matrix are used for building rank codes with restrictions [8] as subcodes.

This paper is concerned with a combined approach which takes advantage of the two previously described methods. It uses the greedy search of the network code components and linear rank codes with restrictions as subcodes. The remaining sections are devoted to the algorithm of the new multicomponent code construction and its decoding for a network channel with random linear transformations and packet errors.

4 Rank-metric subcodes

Before describing the new subspace code construction algorithm, we want to comment on the construction of the rank-metric subcodes we will need. For SKK-codes and Gabidulin-Bossert codes, each matrix of the free elements includes a rectangular submatrix that consists only of the free elements (see equation 8 and the first two matrices from equation 9). In this case, one can use these free elements to construct a conventional rank-metric code which is described in detail in [2].

However, in the case where the columns containing leading elements in the subspace generating matrix are not consecutive, there are some restrictions on the free elements matrix. For example, the generating matrix from equation 1 has the following free elements matrix:

$$U_f = \begin{pmatrix} 0 * * * * * * \\ 0 * * * * * * \\ 0\,0\,0\,0 * * * \\ 0\,0\,0\,0\,0\,0\,0 \\ 0\,0\,0\,0\,0\,0\,0 \end{pmatrix}. \tag{10}$$

Zero elements in the free elements matrix are a result of the Gaussian elimination procedure, and they can not be used as a rank-metric code symbols. Therefore, we need to construct a rank-metric code with restrictions, which means that some of its symbols will be zeros for any input consequence. To solve this problem, we suggest using a systematic form of the rank-metric code, which implies that the input consequence is converted to some of the code symbols

without changes. These symbols are called informational and all the restrictions will be imposed on them.

Our task is to construct a rank-metric code with the minimal distance $d_{rank} = 3$ using the free elements matrix from equation 10. Gabidulin construction of the rank-metric code attains the Singleton bound, so $d_{rank} = n_{rank} - k_{rank} + 1$, where n_{rank} is a total number of the rank-metric code symbols, and k_{rank} is the number of its informational symbols. Note that rank-metric code symbols are vectors over the base field $\mathbb{K}_q$, and we can use both horizontal and vertical representation of its expansion in the basis.

Let the vertical representation of the rank-metric code symbols be used. Since we can place restrictions only on the informational symbols of the rank-metric code, $n_{rank} - k_{rank}$ code symbols should not have any restrictions. In our case, it means that we have to find $n_{rank} - k_{rank} = d_{rank} - 1 = 2$ columns with at least N free elements each, where N determines the size of the rank-metric code. Our goal is to build the code with the biggest cardinality, so the value of N should be the maximum possible. The number of the informational symbols in the rank-metric code will be equal to $N - (n_{rank} - k_{rank})$.

In our case for the free elements matrix from equation 10, we can find $d_{rank} - 1 = 2$ columns with exactly three free elements. It means than $N = 3$ and the number of information symbols is equal to $N - 2 = 1$. Again, in order to build the rank-metric code with the biggest cardinality, we want informational symbols to be the vectors with maximum possible length. This leads us to construct the following code, using the free elements matrix from equation 10:

$$U_f = \begin{pmatrix} 0 & * & * & * & i_{11} & c_{11} & c_{21} \\ 0 & * & * & * & i_{12} & c_{12} & c_{22} \\ 0 & 0 & 0 & 0 & i_{13} & c_{13} & c_{23} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}. \tag{11}$$

This code contains one informational symbol, $i_1 = i_{11}\alpha^0 + i_{12}\alpha^1 + i_{13}\alpha^2$, and two code symbols, $c_1 = c_{11}\alpha^0 + c_{12}\alpha^1 + c_{13}\alpha^2$, and, $c_2 = c_{21}\alpha^0 + c_{22}\alpha^1 + c_{23}\alpha^2$, where α is the primitive element of the field $\mathbb{K}_q^N$. The cardinality of this code is equal to the number of components of the informational symbols vectors. In our case, it means that the rank-metric subcode cardinality is equal to $C_v = q^3$. Note that, after the rank-metric subcode construction, we still have six free elements in the matrix U_f. In our code construction, these free elements can not increase the rank-metric code cardinality, so their values may be chosen arbitrarily.

Now, let the horizontal representation of the rank-metric code symbols be used for the same free elements matrix from equation 10. In this case, we can find $d_{rank} - 1 = 2$ rows with exactly six free elements. It means than $N = 6$ and the number of information symbols is equal to $N - 2 = 4$. But, there is only one row with the free elements remaining. It means that we have to use three all-zero informational symbols, and only one informational symbol which depends on the

input consequence. So the code construction will be the following:

$$U_f = \begin{pmatrix} 0 & c_{21} & c_{22} & c_{23} & c_{24} & c_{25} & c_{26} \\ 0 & c_{11} & c_{12} & c_{13} & c_{14} & c_{15} & c_{16} \\ 0 & 0 & 0 & 0 & i_{44} & i_{45} & i_{46} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}. \tag{12}$$

This code contains four informational symbols (three of which are all-zero), $i_1 = 0$, $i_2 = 0$, $i_3 = 0$, $i_4 = i_{44}\alpha^3 + i_{45}\alpha^4 + i_{46}\alpha^5$, and two code symbols, $c_1 = \sum_{i=1}^{N} c_{1i}\alpha^{i-1}$ and $c_2 = \sum_{i=1}^{N} c_{2i}\alpha^{i-1}$. The cardinality of this code, as in the vertical representation case, is equal to $C_h = q^3$, because we again have exactly three free elements in the informational vector positions. However, in some cases it may happen that the cardinalities of the vertical and horizontal representations of the subspace code are not equal. Then, one should chose the representation with the biggest cardinality and the resulting cardinality of the network code component will be equal to $C = \max\{C_v; C_h\}$.

If, in the process of the rank-metric code construction, we find that there are no free elements in the informational vector positions, then the network code component cardinality will be equal to $C = q^0 = 1$. This means that this code component will have no subcode. An example of such a code component is the component with the generating matrix U_3 from equation 9.

5 New multicomponent code construction

Now, knowing how to calculate the cardinality of the rank-metric subcode for any network code component, we are ready to introduce the following greedy algorithm of the code construction.

For the set of all binary vectors-indices of the length n and Hamming weight k, the cardinality of the corresponding rank subcodes is defined. Lexicographically the first vector-index corresponds to the SKK-code, and it will be used as the first network code component, since the SKK-code will always have the biggest cardinality among all the components [8].

Then, the greedy search starts for the code component with the biggest cardinality among the remaining components. If its subspace distance to all already added to the code subspaces is not less than d_{sub}, then it is included into the code, and so on. The greedy search ends when there are no subspaces left with the desired subspace distance.

In the case where the intermediate stage of the greedy search algorithm yields more than one code component with the same cardinality, we choose the component with the lexicographically first vector-index. This heuristic is based on an assumption that, with this choice, the cardinality of the remaining components will be greater.

Consider an example of the new construction for the code with the following parameters: $n = 7$, $k = 3$ and $d_{sub} = 4$. All binary vector-indices of the length

$n = 7$ and Hamming weight $k = 3$ are listed in equation 13 in lexicographic order:

$$\begin{aligned}\{&1110000, 1101000, 1100100, 1100010, 1100001, 1011000, 1010100,\\ &1010010, 1010001, 1001100, 1001010, 1001001, 1000110, 1000101,\\ &1000011, 0111000, 0110100, 0110010, 0110001, 0101100, 0101010,\\ &0101001, 0100110, 0100101, 0100011, 0011100, 0011010, 0011001,\\ &0010110, 0010101, 0010011, 0001110, 0001101, 0001011, 0000111\}.\end{aligned} \tag{13}$$

The cardinalities of the corresponding rank-metric subcodes for each vector-index are listed in equation 14:

$$\begin{aligned}\{&q^8, q^7, q^6, q^5, q^4, q^6, q^5,\\ &q^4, q^3, q^4, q^3, q^2, q^2, q^1,\\ &q^0, q^6, q^5, q^4, q^3, q^4, q^3,\\ &q^2, q^2, q^1, q^0, q^3, q^2, q^2,\\ &q^1, q^1, q^0, q^0, q^0, q^0, q^0\}.\end{aligned} \tag{14}$$

We chose the SKK-code with the vector-index $v_1 = (1110000)$ and cardinality $C_1 = q^8$ as the first code component. The next (in descending order of cardinality) component has the vector-index $v_2 = (1101000)$. But its subspace distance to the first code component is equal to 2, which is less than the desired $d_{ham} = d_{sub} = 4$, so we can not add it to the code. The first component in the descending cardinality list with the appropriate subspace distance has the vector-index $v_{10} = (1001100)$ and cardinality $C_{10} = q^4$. It is added to the code and the greedy search continues.

Upon completion of the greedy search we obtain the network code of 7 components with the following muli-indices: $\mathcal{I}_1 = \{1, 2, 3\}$, $\mathcal{I}_2 = \{1, 4, 5\}$, $\mathcal{I}_3 = \{2, 4, 6\}$, $\mathcal{I}_4 = \{3, 4, 7\}$, $\mathcal{I}_5 = \{2, 5, 7\}$, $\mathcal{I}_6 = \{3, 5, 6\}$, $\mathcal{I}_7 = \{1, 6, 7\}$. Its cardinality is equal to $M = q^8 + q^4 + q^3 + q^2 + q + q + 1$.

This code is equivalent to the code built in [8] for the same parameters using block designs. But, in some cases, the new code construction allows to obtain a larger cardinality. To illustrate this, we use the concept of the code efficiency - the ratio between its actual value and the theoretical bound from equation 4. The following table shows the values of efficiency for the SKK-code – η_{SKK}, the Gabidulin-Bossert code – η_{GB}, and the new multicomponent code construction – η_{new}, for a fixed set of the code parameters.

Table 1. Efficiency of the network codes for $n = 16$ and $d_{sub} = 3$.

k	3	4	5	6	7	8
η_{SKK}	0,875	0,82	0,794	0,782	0,777	0,774
η_{GB}	1	0,823	0,796	0,782	0,777	0,774
η_{new}	1	0,835	0,798	0,785	0,778	0,775

6 Decoding for channel with errors and erasures

In random network coding the subspaces are transmitted through the channel by means of their generating matrices. For this purpose, the source of the message sequentially transmits all k rows of the $k \times n$ generating matrix, X, that corresponds to some subspace.

Each of the intermediate nodes in the network accumulates all the received rows of the matrix X. Then, the random linear combinations of the rows are transmitted at the first opportunity. In the case when some nodes are under the control of the attacker, these nodes can also distort received messages or add new messages that are neither rows of the matrix X nor their linear combinations.

This channel model can be described by the following equation:

$$Y = AX + Z, \tag{15}$$

where the $m \times k$ matrix A specifies the random linear transformations of the matrix X, the $m \times n$ matrix Z corresponds to the attacker intervention and the $m \times n$ matrix Y is the message received by the destination node.

The decoding of the received message is possible when $d(X;Y) = d(X;AX+Z) < d_{sub}/2$, where d_{sub} is the subspace code distance. It was shown in [3] that this condition is satisfied when the following restrictions on the matrices A and Z hold:

$$k - \mathsf{rank}(A) + 2\mathsf{rank}(Z) < d_{sub}/2. \tag{16}$$

As well as the code construction algorithm, the decoding algorithm consists of two steps. First, we have to recover the code component of the transmitted subspace. For this purpose we apply the Gaussian elimination procedure to the received matrix Y through multiplication by a nonsingular matrix S: $Y' = SY = SAX + SZ$. Note that the matrix Y' corresponds to the same subspace as the received matrix Y. Due to the fact that the matrix S is nonsingular, the decoding condition remains in force: $k - \mathsf{rank}(SA) + 2\mathsf{rank}(SZ) = k - \mathsf{rank}(A) + 2\mathsf{rank}(Z) < d_{sub}/2$.

Since the matrix Y' is in the reduced row echelon form, we can determine its vector-index $v_{Y'}$. From equation 6 we know that $d_{ham}(v_{Y'}; v_X) \leq d(X;Y') < d_{sub}/2$. Together with the subspace code construction method, this condition provides that there is only one code component that has the smallest Hamming distance to the $v_{Y'}$. So, we can find this code component, which corresponds to the transmitted matrix X. The following table represents the simulation results which confirm that for $k - \mathsf{rank}(A) + 2\mathsf{rank}(Z) < d_{sub}/2$ the transmitted code component can be recovered correctly in this manner:

Table 2. Error probability in the code component recovery for $n = 15, k = 5, d_{sub} = 8$.

$k - rank(A) + 2rank(Z)$	0	1	2	3	4	5
P_{err}, %	0	0	0	0	8.05	72.85

After the recovery of the code component we have to decode its rank-metric subcode. To achieve this purpose, we suggest making a rearrangement of the leading elements columns in the matrix Y', by moving them in the first k columns, as in the SKK-code. Since the code component has already been recovered, this rearrangement does not affect the decoding conditions. Then, we apply to the resulting matrix exactly the same decoding technique as described in [8] for the SKK-code.

7 Conclusion

We have proposed a new method of multicomponent network code construction, which has no limits on code parameters and allows decoding of channel errors and erasures by means of standard algorithms. In some cases, it allows us to achieve a higher code cardinality than analogous methods.

For the code construction, we use the greedy search algorithm which may seem computationally expensive. But the exhaustive search is required only during the code construction stage, and does not demand additional resources in the process of messages transmission.

An open problem is to effectively find the vector-index which has the smallest Hamming distance to the vector-index of the received matrix. However, we believe that the number of code components in practical use will be quite small, and, thus, the brute force search is not the worst option.

References

1. *S. T. Xia and F. W. Fu* Johnson type bounds on constant dimension codes // Designs, Codes, Cryptogr., vol. 50, pp. 163-172, Feb. 2009.
2. *Gabidulin E. M.* Theory of Codes with Maximum Rank Distance // Probl. Inform. Transm., vol. 21, No. 1, pp. 1-12, 1985.
3. *D. Silva, F. R. Kschischang, R. Koetter* A Rank-Metric Approach to Error Control in Random Network Coding // IEEE Transactions on Information Theory, vol. 54, pp. 3951-3967, No. 9, Sept. 2008.
4. *Gabidulin E.M., Bossert M.* Codes for Network Coding // IEEE Intern. Symposium on Inf. Theory. Proc. ISIT-08. – 2008. – P. 867-870.
5. *Etzion T., Silberstein N.* Error-Correcting Codes in Projective Spaces via Rank-Metric Codes and Ferrers Diagrams // IEEE Transactions on Information Theory. — 2011. — V. 55, N. 7. — P. 2909–2919.
6. *Etzion T., Silberstein N.* Large Constant Dimension Codes and Lexicodes // Advances in Mathematics of Communications. — 2011. — V. 5, N. 2. — P. 177–189.
7. *Hall M.* // Combinatorial Theory, 1967.
8. *Gabidulin E.M., Pilipchuk N.I.* Rank subcodes in multicomponent network coding // Probl. Inform. Transm. vol. 49, No. 1, pp. 46–60, 2013.