ՀԱՅԱՍՏԱՆԻ ԳԻՏՈՒԹՅՈՒՆՆԵՐԻ ԱԶԳԱՅԻՆ ԱԿԱԴԵՄԻԱ
НАЦИОНАЛЬНАЯ АКАДЕМИЯ НАУК АРМЕНИИ
NATIONAL ACADEMY OF SCIENCES OF ARMENIA
ДОКЛАДЫ ԶԵԿՈՒՅՑՆԵՐ REPORTS

Հատոր Том Volume 113 2013 № 2

INFORMATICS

УДК 025.4.03; 002.5:004

L. H. Aslanyan[1], H. E. Danoyan[2]

Complexity of Elias Algorithm for Hash Functions Based on Hamming and Extended Hamming Codes

Keywords: *best match, nearest neighbors, search algorithm, Elias algorithm, quasi-perfect codes, generalizations of perfect codes.*

1. Introduction. Let $E = \{0,1\}$. Consider Cartesian degree E^n, which is known as the set of vertexes of n-dimensional unit cube. For $x, y \in E^n$ denote by d(x,y) the Hamming distance between vectors x and y. For $x \in E^n$ denote by $S_r^n(x)$ the sphere of radius r, centred at x i.e. $S_r^n(x) = \{y / y \in E^n, d(x,y) \le r\}$ and by $O_r^n(x)$ denote the shell of radius r, of centre x i.e. $O_r^n(x) = \{y / y \in E^n, d(x,y) = r\}$. We will denote by $car(x)$ the carrier of vector $x = (x_1, \dots, x_n)$ then $car(x) = \{i / x_i = 1, i \in \{1, \dots, n\}\}$. Denote by $w(x)$ the weight of vector x i.e. $w(x) = \sum_{i=1}^{n} x_i$. Now let we have a file $F \subseteq E^n$ and a query element $x \in E^n$. Let us consider the problem of finding the set of all "nearest neighbors" of F to x. More precisely it is required to find the set $b(x,F) = \{y \in F / d(x,y) = min_{z \in F}\, d(x,z)\}$. To propose an algorithm for solving the problem of nearest neighbors in applied level, hash coding schemes are considered [1, 2]. A brief description of such schemes is brought below: Hash function is defined as a function $h: E^n \to V$ where $V = \{v_1, \dots, v_N\}$ is a finit set of N elements [1]. In some cases it is possible that $u \ne v, u, v \in E^n$ but $h(u) = h(v)$. Such situations are called collisions. The problem of collisions is solved by the technique called "chaining" [1, 2]. The method is to keep N distinct linked lists L_i(or buckets) one for each possible value of hash function. Denote by B_i the set $\{x \in E^n / h(x) = v_i\}$. In the i-th list L_i are stored those vectors belonging to F which have the same hash value, i.e. $L_i = \{x \in F / h(x) = v_i\}$ or $L_i = B_i \cap F$. Hash coding scheme is called balanced if $|B_i| = 2^n / N$.

2. Preliminaries. 2.1 Elias algorithm. The Elias algorithm [2] considers blocks B_i ordering them by their distances at vector x. Mention that we must

have an efficient method to find all blocks $B_{j_1}, B_{j_2}, \dots, B_{j_{s(j)}}$ located at distance j from x if such blocks exist. After the step of ordering the algorithm examines lists $L_{j_1}, L_{j_2}, \dots, L_{j_{s(j)}}$ one after the other by increase of j. Let the best match distance is denoted by δ. Due to $F \neq \emptyset$ initialisation of δ will happen on some stape. Now, if the current values obey $\delta < j$ the algorithm stops the work. All blocks with higher distances than δ at x do not need to be examined. In the reminder case $\delta \geq j$, examining nonempty list L_{j_k} algorithm can change the best match distance δ, also refreshing the current best match set, or the δ will remain unchanged and the current best match set will be updated.

Elias Algorithm: comment: n is the word length, N is the number of blocks
Input x, F, comment: F $\neq \emptyset$
Integer $\delta = \infty$, comment: the current best match distance
Set S $= \emptyset$, comment: S-is the current set of vectors of F located at distance δ from x
integer j=-1,
while($j < \delta$)
{
j++,
if($s(j) \neq 0$)
for($integer\ i = 0;\ i < s(j);\ i++$)
{
if($L_{j_i} \neq \emptyset$) comment: start examine the list L_i
if ($\delta \leq d(x, L_{j_i})$
$S = S \cup \left(O_\delta^n(x) \cap L_{j_i}\right)$ comment: δ is unchanged
else
{
$S = O_\delta^n(x) \cap L_{j_i}$, comment: δ is changed
$\delta = d(x, L_{j_i})$,
}
}
}
return S, comment: $S = b(x, F), \delta = d(x, F)$

By the complexity of algorithm we mean the average number of examined lists over all files and queries, supposing that each vector $z \in E^n$ can independently appear in F with the same probability p.

2.2 Error-correcting codes. We call a code a nonempty subset C of E^n [3]. Usually for codes some other prescribed properties obeyed (linearity, cyclicity, etc). The code C will be called linear if C is a linear subspase of E^n. Due to the binary nature of spaces considered C is linear when: $\forall c_1, c_2 \in C \Longrightarrow c_1 + c_2 \in C$, mod2 summation is applied. Denote by d_C the minimum distance of the code C i.e. $d_C = min_{\substack{c_1, c_2 \in C \\ c_1 \neq c_2}} d(c_1, c_2)$. The packing radius [3, 4] of C is called the following nonnegative integer: $r_C = [(d_C - 1)/2]$. Denote by R_C the covering

radius [3] of the code C, i.e. $R_C = max_{x\in E^n}\, min_{c\in C}\, d(x,c)$. In the sequel, when it doesn't make a confusion we use notations d, r and R instead of d_C, r_C and R_C respectively. We say that we have an [n, k, d]R code C if the code C is linear, has dimension k, length n, minimum distance d and covering radius R. When the code is nonlinear (or it is not known it being nonlinear) we use the notation $(n, M, d)R$ instead, where $M = |C|$. Denote by $\langle x, y\rangle$ the scalar product of vectors $x = (x_1, \dots, x_n)$ and $y = (y_1, \dots, y_n)$, i.e. $\langle x, y\rangle = x_1y_1 + \cdots + x_ny_n$, where addition is taken by modulo 2. For $x \in E^n$ the coset of linear code C is called the set $x + C = \{x + c/c \in C\}$. As it is known [3] two different cosets do not intersect, and their union covers the space $\mathrm{E^n}$. We denote by G_C the generator matrix of the linear code $C[n, k]$, whith rows forming a basis of code C. Let us denote by H_C the parity check matrix of linear code C. Recall that $\mathrm{H_C}$ is $(\mathrm{n} - \mathrm{k}) \times \mathrm{k}$ matrix and for H_C holds relation $c \in C \Leftrightarrow H_C c^T = 0$. The nonnegative integers $\mathrm{A_0^C, A_1^C, \dots, A_n^C}$, where $\mathrm{A_i^C} = |\{\mathrm{c} \in \mathrm{C/w(c)} = \mathrm{i}\}|$are called weight spectra of code C. Denote by $\mathrm{K_j^n(i)}$ the Kravchouk polynomial of degree j [3, 4] i.e.

$$K_j^n(x) = \sum_{l=0}^{j} (-1)^j \binom{n-x}{j-l}\binom{x}{l}, \text{ where } \binom{x}{l} = \frac{x(x-1)\dots(x-l+1)}{l!}.$$

3. Perfect Codes and some Generalizations. For balanced hash coding schemes it is proposed that the Elias algorithm may be optimal when the blocks $\mathrm{B_i}$ are isoperimetric sets [2, 5] (in simple case spheres). In connectin to it we consider coverings of unit cube by non intersecting spheres. Such coverings can be obtained via perfect codes. When the geometrical interpretation of spherical covers is considered in the models of search of similarities, besides the perfect codes their other possible extensions can be considered and applied, such as nearly perfect codes, strongly uniformly packed codes, quasi perfect codes or coverings by spheres with different radii [4], etc. We brought a brief description of such coverings.

A code C will be called perfect [3-4], if $\mathrm{r_C} = \mathrm{R_C}$. It is known [3, 6] that in binary space nontrivial perfect codes can have only the following two parameter sets.

(I) $(2^m - 1, , 2^{2^m - m - 1}, 3)1$,

(II) $(23, 2^{12}, 7)3$,

here (I) corresponds to the parameters of Hemming codes and (II) refers the case of Golay codes.

Let us consider some generalizations of perfect codes. Let we have a code C, with minimum distance d represented as $2\mathrm{t} + 1$ or $2\mathrm{t} + 2$ (for odd and even d correspondigly). And we suppose that the covering radius $\mathrm{R} \le \mathrm{t} + 1$. Let us denote $\mathrm{D} = \{\mathrm{x/d(x, C)} \ge \mathrm{t}\}$. For $\mathrm{x} \in \mathrm{E^n}$ denote by $\mathrm{A_i(x)}$ the number of codewords of C located at distance i from x. For $\mathrm{x} \in \mathrm{D}$ denote $\mathrm{a(x)} = \mathrm{A_t(x)} + \mathrm{A_{t+1}(x)}$.Note that $\mathrm{A_t(x)} = 1$ or 0. Having $\mathrm{d_C} \ge 2\mathrm{t} + 1$ and $\mathrm{R_C} \le \mathrm{t} + 1$, we may reduce that $\mathrm{a(x)} \le \left[\frac{\mathrm{n+1}}{\mathrm{t+1}}\right]$. Denote by a the average value of a(x) for all $\mathrm{x} \in \mathrm{D}$.

Then $a = \frac{\sum_{c\in C}|O_t^n(c)\cup O_{t+1}^n(c)|}{2^n-|C|\sum_{i=0}^{t-1}\binom{n}{i}} = \frac{|C|\left(\binom{n}{t}+\binom{n}{t+1}\right)}{2^n-|C|\sum_{i=0}^{t-1}\binom{n}{i}}$. The code C will be called nearly perfect [3, 4] if a(x) achieves the possible maximum value $\left[\frac{n+1}{t+1}\right]$ for all $x \in D$, i.e. for nearly perfect codes it takes place the following equality: $|C|\left(\sum_{i=0}^{t-1}\binom{n}{i} + \frac{\binom{n}{t}+\binom{n}{t+1}}{\left[\frac{n+1}{t+1}\right]}\right) = 2^n$. The following parameter sets of nearly perfect codes are known:

(III)$(2^m - 2, 2^{2^m-m-2}, 3)2$;

(IV)$(2^{2m} - 1, 2^{2^{2m}-4m}, 5)3$.

Here (III) corresponds to the parameters of shortened Hemming codes and (IV) corresponds to parameters of punctured Preparata codes. In [7] proved that nearly perfect codes can have only the one of mentioned parameter sets.

The code C will be called strongly uniformly packed if $a(x) = a$ for all $x \in D$ [4]. The parameters of strongly uniformly packed codes are known too [4].

The code C will be called quasi-perfect if $R = r + 1$ [3, 4]. Many families of quasi perfect codes are known for the covering radius ≤ 4 [4, 8-13] but the general problem of existence of quasi-perfect codes by the given parameters isn't completely solved yet [8]. Also the nearly perfect codes appear as a special class of quasi-perfect codes.

Let $i \geq 1$ and $R_1, \dots, R_i$ are integers, $C = \bigcup_{j=1}^{i} C_i$. Code C will be called perfect i radius code if the spheres with radii $R_1, \dots, R_i$ respectively centered at points of code sets $C_1, \dots, C_i$ do not intersect and their union covers the whole space [4]. These structures are another candidate that we may apply in model of best match search below, but there are not known exhausting results also about existence of such codes [4].

4. The Complexity of the Algorithm. Suppose we have an $[n, k]$ code C with covering radius R and $C = \{c_1, c_2, \dots, c_{2^k}\}$. We define a hash function $h: E^n \longrightarrow C$, associated to the code C in the following way:

$$h_C(x) = \{c_i / \, d(x, c_i) = min_{c\in C}\{d(x, c)\}\}. \quad (1)$$

As it follows from (1) $h_C(x)$ could be multivalued function because the blocks B_i are spheres of radius R, and they can intersect (recall that $B_i = \{x \in E^n / h_C(x) = c_i\}$, $i \in \{1, \dots, 2^k\}$). When the code C is perfect the mentioned blocks do not intersect and their union covers the unit cube. The formula below for complexity of algorithm is brought for the case corresponding to Hamming code. We also consider hash functions associated to codes in some sense "near" to perfect codes. Such property have also the so called quasi-perfect codes [3, 4]. Indeed the algorithm is proposed for balanced hash coding schemes where different blocks B_i do not intersect, but we will also consider the algorithm for the case of intersecting blocks. In this case when blocks intersect we create the list in a similar way to the basic case and then these lists are also intersecting. Repeated element bring some redundancy (in terms of memory). The formal expression of complexity of algorithm is then

brought for the particular case of extended Hamming code. To write a formula of complexity of the algorithm, for $x \in E^n$ let us consider the following table:

	x	p_1 F_1	p_2 F_2	⋯	$p_{2^{2^n}}$ $F_{2^{2^n}}$	probability subset
	B_1	a^x_{11}	a^x_{12}	⋯	$a^x_{1\,2^{2^n}}$	
Blocks	B_2	a^x_{21}	a^x_{22}	⋯	$a^x_{2\,2^{2^n}}$	
	⋮	⋮	⋮	⋯	⋮	
	B_{2^k}	$a^x_{2^k 1}$	$a^x_{2^k 2}$	⋯	$a^x_{2^k\,2^{2^n}}$	

$F_1, F_2, \dots, F_{2^{2^n}}$ are all subsets of vertexes of unit cube and each F_i could be generated with the corresponding probability p_i. We will use the values a^x_{ij} putting them in the cells corresponding to block B_i and subset F_j, where

$$a^x_{ij} = \begin{cases} 1 & if\ B_i is considered\ in\ case\ of\ set\ F_i\ and\ vertex\ x, \\ 0 & otherwise. \end{cases}$$

As we mentioned in section 2.1, the complexity of algorithm will be represented as

$$\alpha(h_C) = \frac{1}{2^n}\sum_{x\in E^n}\sum_{1\le i\le 2^k}\sum_{1\le j\le 2^{2^n}} p_j a^x_{ij}.$$

Let us denote $\Phi_x(B_i) = \sum_{1\le j\le 2^{2^n}} p_j a^x_{ij}$. As we can see $\Phi_x(B_i)$ is the probability that the block B_i will be considered by the algorithm when the vector x is requested. Then

$$\alpha(h_C) = \frac{1}{2^n}\left[\sum_{x\in E^n}\sum_{1\le i\le 2^k}\Phi_x(B_i)\right],$$

It is easy to understand that for a fixed query x the block B_i will be examined if the sphere $S^n_{d(x,B_i)-1}$ does not contain any vector belonging to F. In that case all blocks B_l such that $d(x,B_l) \le d(x,B_i) - 1$, will be examined. Let j vary over all possible distances between vector x and blocks B_i. Denote by $T_x(j)$ the number of blocks located at distance $\le j$ from vector x, then

$$\alpha(h_C) = \frac{1}{2^n}\sum_{x\in E^n}\sum_{0\le j\le n} T_x(j)V(j). \qquad (2)$$

where $V(j)$ denotes the probability that the nearest vector in F is located at distance j from x. Recall that [2]

$$V(j) = (1-(1-p)^{\binom{n}{j}})(1-p)^{\sum_{l=0}^{j-1}\binom{n}{l}}.$$

As $d(x,C_i) = w(x+c_i)$, then the number of vectors located at distance i is equal to A_i^{x+C}. The sphere with centre c_i and radius R will be located in a distance ≤ j from vector x if and only if $d(x,c_i) \le j + R$. Therefore

$$T_x(j) = \sum_{i=0}^{j+R} A_i^{x+C}. \qquad (3)$$

We consider that $A_i^{x+C} = 0$ when $i > n$.

5. Case of Hamming code and extended Hamming code. Denote by $\mathcal{H}_m$ the Hamming code of length $n = 2^m - 1$. As we know [3], $\mathcal{H}_m$ is $[2^m - 1, 2^m - m - 1, 3]1$ perfect code. The parity check matrix of $\mathcal{H}_m$ is the following:

$$H_{\mathcal{H}_m} = \begin{pmatrix} 0 & 0 & \cdots & 1 \\ \vdots & \vdots & \cdots & \vdots \\ 0 & 1 & \ddots & 1 \\ 1 & 0 & \cdots & 1 \end{pmatrix}, \quad (4)$$

The code $\mathcal{H}_m$ has two types of cosets: the code $\mathcal{H}_m$ itself and $e_i + \mathcal{H}_m$, where $supp(e_i) = \{i\}$, $i = 1, \dots, n$. Coset weight spectra in these cases are respectively

$$A_j^{\mathcal{H}_m} = \frac{1}{2^m}\left(K_j^n(0) + (2^m - 1)K_j^n(2^{m-1})\right), \quad (5)$$

$$A_j^{e_i+\mathcal{H}_m} = \frac{1}{2^m}\left(\binom{2^m - 1}{j} - K_j^n(2^{m-1})\right). \quad (6)$$

From (2),(3),(5) and (6) follows:

Proposition 1. *The complexity of algorithm for the hash function defined by* $[2^m - 1, 2^m - m - 1, 3]1$ *Hamming code* $\mathcal{H}_m$ *is:*

$$\alpha(h_{\mathcal{H}_m}) = \frac{1}{2^m}\sum_{0 \le j \le 2^m - 1} V(j)\left(\sum_{i=0}^{j+1}\left(A_i^{\mathcal{H}_m} + (2^m - 1)A_i^{e_i+\mathcal{H}_m}\right)\right). \quad (7)$$

Let us consider the extended Hamming code, which we denote by $\widehat{\mathcal{H}_m}$.It is known [3], that $\widehat{\mathcal{H}_m}$ is $[2^m, 2^m - m - 1, 4]2$ quasi-perfect code, and its parity check matrix is:

$$H_{\widehat{\mathcal{H}_m}} = \begin{pmatrix} 1 & 1 & 1 & \cdots & 1 \\ 0 & 0 & 0 & \cdots & 1 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 1 & \cdots & 1 \\ 0 & 1 & 0 & \cdots & 1 \end{pmatrix}.$$

It could be obtained, that the code has three types of cosets

(a) $\widehat{\mathcal{H}_m}$,

(b) $e_i + \widehat{\mathcal{H}_m}$, where $car(e_i) = \{i\}$, $i \in \{1, \dots, n\}$,

(c) $g_i + \widehat{\mathcal{H}_m}$, where $car(g_i) = \{1, i\}$, $i \in \{2, \dots, n\}$.

Coset weight spectra in these cases are respectively

$$A_j^{\widehat{\mathcal{H}_m}} = \frac{1}{2^{m+1}}\left(K_j^n(0) + (2^{m+1} - 2)K_j^n(2^{m-1}) + K_j^n(2^m)\right) \quad (8)$$

$$A_j^{e_i+\widehat{\mathcal{H}_m}} = \frac{1}{2^{m+1}}\binom{2^m}{j}\left(1 - (-1)^j\right) \quad (9)$$

$$A_j^{g_i+\widehat{\mathcal{H}_m}} = \frac{1}{2^{m+1}}\left(\binom{2^m}{j}\left(1 + (-1)^j\right) - 2K_j^{2^m}(2^{m-1})\right) \quad (10)$$

Keeping in mind this and the fact that each coset contains $2^{2^m - m - 1}$ vectors and the number of cosets of first, second and third types is equal to 1, 2^m and $2^m - 1$ respectively from (2) and (3) we get:

Proposition 2. For the hash function defined by $[2^m, 2^m - m - 1, 4]2$ *extended Hamming code* $\widehat{\mathcal{H}_m}$ *the complexity of algorithm is:*

$$\alpha(h_{\widehat{\mathcal{H}_m}}) = \sum_{0 \le j \le 2^m} V(j)\left(\sum_{i=0}^{j+2}\left(\frac{1}{2^{m+1}}A_i^{\widehat{\mathcal{H}_m}} + \frac{1}{2}A_i^{e_i+\widehat{\mathcal{H}_m}} + \frac{2^m - 1}{2^{m+1}}A_i^{g_i+\widehat{\mathcal{H}_m}}\right)\right). \quad (11)$$

1- Institute for Informatics and Automation Problems of NAS RA, lasl@sci.am

2- Yerevan State University, hdanoyan@yandex.ru

L. H. Aslanyan[1], H. E. Danoya[2]

Complexity of Elias Algorithm for Hash Functions Based on Hamming and Extended Hamming Codes

The procedure of finding the set of all “nearest neighbors” in a set, known as the Elias algorithm is addressed. In connection to it the hash coding schemes associated with the n-dimensional unit cube coverings by non-intersecting spheres of the same radius is considered. Such coverings, in particular, can be obtained via perfect codes. We get a formula presentation for complexity of the search algorithm in case of Hamming codes. As such coverings are possible in very simple cases and we consider coverings by intersecting spheres of the same radius. These can be obtained via quasi-perfect codes. A formula of complexity of algorithm for extended Hamming codes is obtained ~~as~~.

Լ. Հ. Ասլանյան, Հ. Է. Դանոյան

Էլեասի ալգորիթմի բարդությունը Հեմմինգի և ընդլայնված Հեմմինգի կոդերով սահմանված հաշ-ֆունկցիաների համար

Ուսումնասիրման առարկան բազմության տրված էլեմենտի «ամենամոտ հարևանների» գտնելու հայտնի Էլեասի ալգորիթմն է: Դրա հետ կապված դիտարկվում են հաշ-կոդավորման սխեմաներ՝ ասոցիացված n-չափանի միավոր խորանարդի՝ միևնույն շառավղով չհատվող գնդերով ծածկույթների հետ: Այդպիսի ծածկույթներ ստացվում են կատարյալ կոդերի միջոցով: Բերված է որոնման ալգորիթմի բարդության բանաձև Հեմմինգի կոդի դեպքում: Քանի որ նման ծածկույթներ գոյություն ունեն եզակի դեպքերում, դիտարկում ենք նաև ծածկույթներ միևնույն շառավղով հատվող գնդերի տեսքով: Այդպիսի ծածկույթներ մասնավորապես ստացվում են քվազիկատարյալ կոդերի միջոցով: Բերված է ալգորիթմի բարդության բանաձև ընդլայնված Հեմմինգի կոդի դեպքում:

Л. А. Асланян, А. Э. Даноян

Сложность алгоритма Элеаса для хеш-функциий определенных кодами Хемминга и расширенными кодами Хемминга

Известен алгоритм нахождения всех «ближайших соседей» к данной точке из данного множества. В связи с этим рассматриваются схемы хеш-кодирования, ассоцированные с покрытиями n-мерного единичного куба с непересекающими шарами равного радиуса. Такие покрытия получаются с помощью совершенных кодов. Поскольку такие покрытия существуют в единичных случаях, мы рассматриваем покрытия с пересекающми шарами равного радиуса. Такие покрытия в частности получаются с помощью квазисовершенных кодов. Приведена формула сложности алгоритма для случая расширенных кодов Хемминга.

References

1. *Knuth D. E.*, The Art of Computer Programming, V. 3 / Sorting and Searching, second edition, Eddison-Wesley, 780p., 1998
2. *R.L. Rives*t On The Optimality of Elias's Algorithm for Performing Best-Match Searches, Information Processing 74, North-Holland Publishing Company, pp. 678-681, 1974.
3. *F. J. Mac-Williams, N. J. Sloane*, "The theory of error-correcting codes", North Holland Publishing Company, 762p., 1977
4. *Cohen G., Honkala I., Litsyn S., Lobstein A.*, "Covering Codes", North-Holland Mathematical Library, vol. 54, 542p. 1997
5. *Асланян Л. А.* -Проблемы кибернетики. 1979. Вып. 36. С. 85-127.
6. *Zinov'ev V. A., Leont'ev V. K.* - Problems of Control and Info. Theory, 2(2). 1973. P. 123-132.
7. *Lindstrom K.*- Ann. Univ. 1975. Turku.Ser A,169, P. 3-28.
8. *Baicheva T., Bouyukliev I., Dodunekov S.* - IEEE Transactions of Information Theory.2008. V. 54, 9, P. 4335-4339.
9. *Gabidulin E. M., Davydov A. A., Tombak L. M.*- IEEE Trans. Inf. Theory. 1991. V. 37, 1, P. 219–224.
10. *Davydov A. A., Tombak L. M.* - Probl. Inf. Transm. 1989. V. 25, 4, P. 265–275.
11. *Davydov A. A., Drozhzhina-Labinskaya A. Yu.,* - IEEE Trans. Inf. Theory. 1994. V. 40, 4, P. 1270–1279.
12. *Etzion T., Mounits B.* - IEEE Trans. Inform. Theory. 2005. V. 51, 11, P. 3938-3946.
13. *Etzion T., Greenberg G.,* - IEEE Trans. Inf. Theory. 1993. V. 39, 1, P. 209–214.