The 2006 International Seminar of E-commerce
Academic and Application Research
Tainan, Taiwan, R.O.C, March 1-2, 2006

An Application of Coding Theory into Experimental Design – Construction Methods for Unequal Orthogonal Arrays –

Shigeichi Hirasawa *

Abstract— The relationship between coding theory and the orthogonal arrays is discussed in terms of the theory of Galois field. Since coding theory easily gives many codes with large minimum distance, it is useful to construct the orthogonal arrays with large strength which is applicable to experiments with high order interaction effects between factors. First, we review the result from the argument of coding theory, and starting from complete design, orthogonal design is introduced from the view-point of experimental design. Next, correspondence of parameters between error-correcting codes and orthogonal arrays is clarified. Finally, by using the construction methods for unequal error protection codes, orthogonal arrays are extended to those with unequal strength. Methods for constructing the orthogonal arrays with unequal strength based on coding theory is practically important, because most of all real problems which we usually treat must assume that the interaction effects between two or more factors of experiments are not equal. If the model for experiments is given, we can attain the same accuracy by the orthogonal design as that by complete design with fewer experiments.

Keywords—Experimental design, Orthogonal array, Error-correcting code, Galois field

1 Introduction

In the fundamentals in computer sciences, one of the most important theory is coding theory, or the theory of error-correcting codes (ECCs), which has the history of almost a half century [Hira99]. Research works in this area have been devoted and accumulated to apply them into actual systems such as the computer main memory, data transmission systems, deep space communication systems, the compact disc (CD) for music players, cellar phones and so on.

On the other hand, in the field of statistical data analysis, the experimental design has contributed to effectively analyze experimental data [Taka79]. Especially, orthogonal arrays (OAs) are important to construct methods for experiments and to analyze the data with taking into account of interaction effects between factors [HSS99].

Many methods for constructing the OAs have been given by techniques based on projective geometry (PG) [Taka79]. They are useful to apply to experiments with low order interaction effect, and they are difficult, however, to apply to those with higher one. Since we can easily obtain many codes with large minimum distance by coding theory [PW71][MS77], it is effective to construct the OAs with large strength based on ECCs, where the large minimum distance of the codes corresponds to large strength of the OAs which can treat high order interaction effects between factors of the experiments. Note that the purpose of introducing the OAs is to make the number of experiments reduce without degradation in accuracy of the estimation of parameters compared to complete design.

First, we show that there is a close relationship between the ECCs and the OAs through the theory of Galois field. Constructing methods for linear OAs given by those for linear codes are discussed, and the correspondence between parameters of the ECCs and those of the OAs is clarified.

Based on an idea of unequal error protection codes [MW67][Gils83], OAs are extended to those with unequal strength (UOAs) [SMH05]. If the model of the experiments assumes that there do not exist the all of interaction effects between L or fewer factors, then the UOAs can effectively eliminate needless experiments. The constructing algorithm for the UOAs is demonstrated. We show a few examples of the UOAs.

In section 2, we describe brief introduction of error-correcting codes, complete design, and orthogonal design. Section 3 discusses properties of error-correcting codes and orthogonal arrays. The correspondence between them is also discussed. In section 4, a new construction method for orthogonal arrays with unequal strength is proposed based on that for unequal error protection codes, and its examples are shown in section 5. Concluding remarks are stated and recent works and further research are notified in section 6.

Throughout this paper, we discuss linear OAs and UOAs with s level, where s is a prime power. Nonlinear OAs and UOAs can be constructed by nonlinear codes [HSS99][SMH05].

2 Preliminary

2.1 Error-Correcting Codes

In this section, we briefly review the results obtained by coding theory [PW71][MS77][Hira83][Hira99].

2.1.1 Codes and Minimum Distance

Definition 2.1 Suppose a q-ary code of length n, number of information symbols k, and (designed) minimum distance d denoted by an (n, k, d) code, then the code consists of q-ary vectors $\boldsymbol{x_1}, \boldsymbol{x_2}, \cdots, \boldsymbol{x_M}$ which are called codewords, where

$$\boldsymbol{x_m} = (x_{m1}, x_{m2}, \cdots, x_{mn}), m = 1, 2, \cdots, M, \quad (2.1)$$

* Department of Industrial and Management Systems Engineering, School of Science and Engineering, Waseda University, 3-4-1, Ohkubo, Shinjuku, Tokyo 169-8555 Japan Phone:+81-3-5286-3290. FAX:+81-3-5273-7215. E-mail:hirasawa@hirasa.mgmt.waseda.ac.jp

and

$$d = \min_{m,m'(m \neq m')} D_H(x_m, x_{m'}), \tag{2.2}$$

$$D_H(\boldsymbol{x_m}, \boldsymbol{x_{m'}}) = \Sigma_{i=1}^{n} d_H(x_{mi}, x_{mi'}),$$

$$d_H(a,b) = \begin{cases} 0, & a = b; \\ 1, & a \neq b, \end{cases}$$

and where q is a prime power. □

If the minimum distance of the code is not specified, we denote the code as an (n,k) code. The rate r of the (n,k,d) code is defined by $r = k/n$. M is the number of coedwords.

2.1.2 Linear Codes

A linear code C has the following property:

$$\forall \boldsymbol{x_i}, \boldsymbol{x_j} \in C, \exists \boldsymbol{x_\ell} = \boldsymbol{x_i} + \boldsymbol{x_j} \in C, \tag{2.3}$$

A generator matrix of an (n,k,d) (linear) code is given by

$$G = \begin{bmatrix} \boldsymbol{x_1} \\ \boldsymbol{x_2} \\ \vdots \\ \boldsymbol{x_k} \end{bmatrix}, \tag{2.4}$$

where $\boldsymbol{x_1}, \boldsymbol{x_2}, \cdots, \boldsymbol{x_k}$ are chosen to be mutually independent, where the rank of G is k. By a combination of row operations and column permutations, G can be lead to echelon canonical form G':

$$G' = [I_k, P], \tag{2.5}$$

which generates equivalent systematic code to the code C, where I_k denotes the identity matrix of dimension k. Denote the information symbols by $\boldsymbol{u} = (u_1, u_2, \cdots, u_k)$, then the codeword $\boldsymbol{x}$ is given by

$$\boldsymbol{x} = \boldsymbol{u}G. \tag{2.6}$$

A parity check matrix H of the code C generated by G' is given by

$$H = [-P^{\mathrm{T}}, I_{n-k}]. \tag{2.7}$$

Note that the following equation holds:

$$\forall \boldsymbol{x_m}, \boldsymbol{x_m} H^{\mathrm{T}} = 0. \tag{2.8}$$

To construct an (n,k,d) code, the following theorem is important (See Appendix B [Hira83]).

Theorem 2.1 Let H be a parity check matrix of the (n,k) code. The minimum distance of the code is at least d, if and only if every combination of $d-1$ or fewer columns of H is linearly independent.[1] □

For the later discussion, dual coeds must be defined. Since the row space of generator matrix G gives subspace C of dimension k, its null space is a vector space $C^\perp$ of dimension $n-k$.

[1] This condition is equivalent to that the sum of every combination of $d-1$ or fewer columns is non-zero.

Definition 2.2 (Dual codes) Let a code C be a subspace of n-tuples, then a dual code $C^\perp$ of a code C is a null space of C. □

Note that the generator matrix G of the code C is the parity check matrix $H^\perp$ of the code $C^\perp (G = H^\perp)$, and similarly $H = G^\perp$. If C is an (n,k) code, then $C^\perp$ is an $(n, n-k)$ code.

2.1.3 BCH Codes and RS Codes

We already have many linear codes with various parameters of n, k, and d which can be easily constructed.

Theorem 2.2 (BCH code) Let the roots of a generator polynomial $g(z)$ of the (n,k) BCH code over $GF(q)$ be $\alpha^{m_0}, \alpha^{m_0+1}, \cdots, \alpha^{m_0+d-1}$, where m_0 is any integer and α, any element of $GF(q^m)$. Then the minimum distance of the code is at least d. □

Theorem 2.3 (BCH bound) An (n,k,d) BCH code over $GF(q)$ has parameters such that:

$$n = q^m - 1,$$
$$n - k \leq m(d-1). \tag{2.9}$$

□

Corollary 2.1 (Binary BCH code bound) A binary (n,k,d) BCH code has parameters such that:

$$n = 2^m - 1,$$
$$n - k \leq m\lfloor (d-1)/2 \rfloor, \tag{2.10}$$

where $\lfloor a \rfloor$ implies the largest integer larger than or equal to a. □

For an (n,k,d) BCH code over $GF(q)$, letting $m = 1$, and $n = q-1$, we have RS code over $GF(q)$.

Corollary 2.2 (RS code) An (n,k,d) RS code over $GF(q)$ has parameters satisfying:

$$\begin{aligned} n &= q - 1, \\ k &\leq q - 1, \\ d &= n - k + 1. \end{aligned} \tag{2.11}$$

□

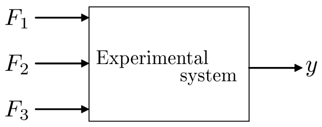
2.2 Experimental Design

First, we give a simple example to show the cases of experiments.

Example 2.1 (Experimental system) Let F_1, F_2, and F_3 be factors which may affect a ratio y of defective product, and let each factor have 2 levels, where F_1, F_2, and F_3 correspond to the choice of materials, machines, and temperatures, respectively as shown in Fig. 2.1. □

Suppose that we want to analyze how the level of factors affects the ratio of defective products. In this example, we assume that the model has three input factors with discrete variables and one output characteristic with continuous variable[2].

[2] If all input factors are continuous variables, then the regression analysis is applied, while input factors are composed of both continuous and discrete variables, then the variance analysis is used.



F_1: F^1_0 (made in A company)
F^1_1 (made in B company)
F_2: F^2_0 (a new machine)
F^2_1 (an old machine)
F_3: F^3_0 (100 ℃)
F^3_1 (200 ℃)
y : Ratio of defective product

Figure 2.1: A model of the experiment system

2.2.1 Complete Design

We usually assume a mode for experiments based on hypothesis, assumption, or prior knowledge for the experimental system. The model is represented by a structure and its parameters, and is expressed by formula.

Example 2.2 (Model of experimental system) In Example 2.1 shown in Fig. 2.1, if we assume a model for which the interaction effects have all combinations of 2 factors (the 2nd order interaction), then we have the following equation:

$$\begin{aligned} y_{\nu_1,\nu_2,\nu_3} &= \mu + \alpha^1_{\nu_1} + \alpha^2_{\nu_2} + \alpha^3_{\nu_3} \\ &\quad + \alpha^{1,2}_{\nu_1,\nu_2} + \alpha^{1,3}_{\nu_1,\nu_3} + \alpha^{2,3}_{\nu_2,\nu_3} + e_{\nu_1,\nu_2,\nu_3}, \\ &(\nu_i \in \{0,1\},\ i \in \{1,2,3\}) \end{aligned} \tag{2.12}$$

where y_{ν_1,ν_2,ν_3} is the ratio of defective products which is given for level combination $F^1_{\nu_1}F^2_{\nu_2}F^3_{\nu_3}$. In Eq.(2.12) μ is a constant which has no relation with levels, and is called the central effect. $\alpha^i_{\nu_i}$ is the effect which appears when F_i is set for ν_i, and is called the main effect of F_i. $\alpha^{i_1,i_2}_{\nu_{i_1},\nu_{i_2}}$ is the effect which appears by combining $F^{i_1}_{\nu_{i_1}}$ with $F^{i_2}_{\nu_{i_2}}$, and is called the interaction effect of $F_{i_1}F_{i_2}$. e_{ν_1,ν_2,ν_3} is a random error. □

The complete design always requires experiments with all combinations of levels for each factor which is called a complete array as shown in Table 2.1 for Example 2.1.

Table 2.1: Experiment conditions and data

Experiment no.	F_1	F_2	F_3	y [%]
1	0	0	0	0.5
2	0	0	1	0.4
3	0	1	0	0.1
4	0	1	1	0.1
5	1	0	0	1.2
6	1	0	1	1.5
7	1	1	0	0.7
8	1	1	1	0.6

We must estimate parameters from the output so that maximum likelihood and minimum squire error criteria are satisfied. For Example 2.2, we have the following relations: For any $i \in \{1,2,3\}$,

$$\sum_{\nu_i \in \{0,1\}} \alpha^i_{\nu_i} = 0, \tag{2.13}$$

and for any $i_1, i_2 \in \{1,2,3\}$, $i_1 \neq i_2$,

$$\sum_{\nu_{i_2} \in \{0,1\}} \alpha^{i_1,i_2}_{\nu_{i_1},\nu_{i_2}} = 0 \quad for\ \ \forall \nu_{i_1} \in \{0,1\}, \tag{2.14}$$

$$\sum_{\nu_{i_1} \in \{0,1\}} \alpha^{i_1,i_2}_{\nu_{i_1},\nu_{i_2}} = 0 \quad for\ \ \forall \nu_{i_2} \in \{0,1\}. \tag{2.15}$$

Letting $\hat{\mu}$, $\hat{\alpha}^i_\phi$, $\hat{\alpha}^{i_1,i_2}_{\phi,\psi}$ be a estimator of μ, α^i_ϕ, $\alpha^{i_1,i_2}_{\phi,\psi}$, the estimation of parameters is completed by:

$$\hat{\mu} = \frac{1}{8} \sum_{(\nu_1,\nu_2,\nu_3) \in \{0,1\}^3} y_{\nu_1,\nu_2,\nu_3}, \tag{2.16}$$

$$\hat{\alpha}^i_\phi = \frac{1}{4} \sum_{(\nu_1,\nu_2,\nu_3) \in \{0,1\}^3, \nu_i = \phi} y_{\nu_1,\nu_2,\nu_3} \quad - \hat{\mu}, \tag{2.17}$$

and

$$\hat{\alpha}^{i_1,i_2}_{\phi,\psi} = \frac{1}{2} \sum_{\substack{(\nu_1,\nu_2,\nu_3) \in \{0,1\}^3, \\ \nu_{i_1} = \phi, \nu_{i_2} = \psi}} y_{\nu_1,\nu_2,\nu_3} - \hat{\mu} - \hat{\alpha}^{i_1}_\phi - \hat{\alpha}^{i_2}_\psi. \tag{2.18}$$

For example, $\hat{\alpha}^1_0 = \frac{1}{4}(y_{0,0,0} + y_{0,0,1} + y_{0,1,0} + y_{0,1,1}) - \hat{\mu}$ is calculated as follows:

$$\begin{array}{rl} y_{0,0,0} = & \mu + \alpha^1_0 + \alpha^2_0 + \alpha^3_0 + \alpha^{1,2}_{0,0} + \alpha^{1,3}_{0,0} + \alpha^{2,3}_{0,0} + e_{0,0,0}, \\ y_{0,0,1} = & \mu + \alpha^1_0 + \alpha^2_0 + \alpha^3_1 + \alpha^{1,2}_{0,0} + \alpha^{1,3}_{0,1} + \alpha^{2,3}_{0,1} + e_{0,0,1}, \\ y_{0,1,0} = & \mu + \alpha^1_0 + \alpha^2_1 + \alpha^3_0 + \alpha^{1,2}_{0,1} + \alpha^{1,3}_{0,0} + \alpha^{2,3}_{1,0} + e_{0,1,0}, \\ y_{0,1,1} = & \mu + \alpha^1_0 + \alpha^2_1 + \alpha^3_1 + \alpha^{1,2}_{0,1} + \alpha^{1,3}_{0,1} + \alpha^{2,3}_{1,1} + e_{0,1,1}, \\ \hline \hat{\alpha}^1_0 = & (\mu - \hat{\mu}) + \alpha^1_0 \qquad\qquad\qquad\qquad\qquad + \bar{e}^1_0, \end{array}$$

where $\bar{e}^1_0 = \frac{1}{4}(e_{0,0,0} + e_{0,0,1} + e_{0,1,0} + e_{0,1,1})$. This is because we assumed Eqs.(2.13), (2.14) and (2.15). When the output y for each experiment is given as Table 2.1, an estimated value of y is calculated as shown in Appendix A.

2.2.2 Orthogonal Design

The orthogonal design is used to reduce the number of experiments depending on a model of experiments, which is assumed usually by prior knowledge of objective systems.

Definition 2.3 [HSS99] An $M \times n$ array A with elements from $GF(s)$ is said to be an Orthogonal Array with s levels and strength τ, if every $M \times \tau$ subarray of A contains each τ-tuple based on $GF(s)$ exactly same times as row. We will denote such an array by $OA(M, n, s, \tau)$. □

Example 2.3 In Example 2.1 shown in Fig. 2.1, if we assume no interaction effect for all factors (the 1st order interaction), then we have the following equation:

$$y_{\nu_1,\nu_2,\nu_3} = \mu + \alpha^1_{\nu 1} + \alpha^2_{\nu_2} + \alpha^3_{\nu_3} + e_{\nu_1,\nu_2,\nu_3}.$$

In this case, $OA(4, 3, 2, 2)$ as shown in Table 2.2 is enough to estimate parameters, hence the number of experiments decreases. Table 2.2 is called an orthogonal array for Example 2.1. □

Table 2.2: Experiment conditions and data

Experimental no.	F_1	F_2	F_3	y(%)
1	0	0	0	0.5
2	0	1	1	0.1
3	1	0	1	1.5
4	1	1	0	0.7

The estimation of parameters is also followed:

$$\hat{\mu} = \frac{1}{|\bar{A}|} \sum_{(\nu_1,\nu_2,\nu_3)\in\bar{A}} y_{\nu_1,\nu_2,\nu_3}, \quad (2.19)$$

$$\hat{\alpha}^i_\phi = \frac{1}{|\bar{A}^i_\phi|} \sum_{(\nu_1,\nu_2,\nu_3)\in\bar{A}^i_\phi} y_{\nu_1,\nu_2,\nu_3} - \hat{\mu}, \quad (2.20)$$

where $\bar{A}$ is the set of the rows of $OA(4,3,2,2)$ and $\bar{A}^i_\phi = \{(\nu_1,\nu_2,\nu_3)|(\nu_1,\nu_2,\nu_3) \in \bar{A}, \nu_i = \phi\}$. For example, $\hat{\alpha}^1_0 = \frac{1}{2}(y_{000} + y_{011}) - \hat{\mu}$ is given as follows:

$$\begin{aligned} y_{0,0,0} &= \mu \ + \alpha^1_0 + \alpha^2_0 + \alpha^3_0 + e_{0,0,0}, \\ y_{0,1,1} &= \mu \ + \alpha^1_0 + \alpha^2_1 + \alpha^3_1 + e_{0,1,1}, \\ \hline \hat{\alpha}^1_0 &= (\mu - \hat{\mu}) + \alpha^1_0 \qquad\qquad + \bar{e}^1_0, \end{aligned}$$

where, $\bar{e}^1_0 = \frac{1}{2}(e_{0,0,0} + e_{0,1,1})$. This is because we assumed Eq.(2.13).

Let $F_1, F_2, \ldots, F_n$ denote the n factors to be included in the experiment. We assume that each factor has s levels, so we can describe the set of levels as $GF(s)$, where s is a prime power.

1. Case $\tau = 2$ (the 1st order interaction)
   If we can assume that there is no interaction effect, we have
   $$y_{\nu_1,\nu_2,\ldots,\nu_n} = \mu + \alpha^1_{\nu_1} + \alpha^2_{\nu_2} + \ldots + \alpha^n_{\nu_n} + e_{\nu_1,\nu_2,\ldots,\nu_n}, \quad (2.21)$$
   we can reduce the number of experiments by using an OA with strength $\tau = 2$, i.e., $OA(M,n,s,2)$. When we use an OA to experimental design, each column corresponds to the factor in the experiment, and each row, to the level combination of the factors.
2. Case $\tau =$ 2-4 (the 1st and the 2nd order interaction)
   We consider some interaction effects of two factors. Let $I \subset \{1,2,\ldots,n\}^2$ be the set whose element is a pair of indices of two factors in which there may be interaction effect. When we can assume that
   $$y_{\nu_1,\nu_2,\ldots,\nu_n} = \mu + \alpha^1_{\nu_1} + \alpha^2_{\nu_2} + \ldots + \alpha^n_{\nu_n} + \sum_{(i_1,i_2)\in I} \alpha^{i_1,i_2}_{\nu_{i_1},\nu_{i_2}} + \ldots + e_{\nu_1,\nu_2,\ldots,\nu_n}, \quad (2.22)$$
   we need an $M \times n$ array A which satisfies the following three conditions;
   (1) The array A has strength 2.
   (2) The array A partially has strength 3, that is, for any i_1, i_2 ($(i_1,i_2) \in I$), every $M \times 3$ subarray, which contains two columns that correspond to F_{i_1} and F_{i_2}, contains each 3-tuple based on $GF(s)$ exactly same times as row.
   (3) The array A partially has strength 4, that is, for any i_1,i_2,i_3,i_4 ($(i_1,i_2),(i_3,i_4) \in I$), $M \times 4$ subarrays, which contains four columns that correspond to F_{i_1}, F_{i_2}, F_{i_3}, and F_{i_4}, contains each 4-tuple based on $GF(s)$ exactly same times as row.

   In the special case, if there are all interaction effects of two factors, we need an OA with strength 4. Generally, if there are all interaction effects of L factors, we need an OA with strength $\tau = 2L$.

Definition 2.4 (L-th order interaction model) Let there exist the interaction effects of all combinations of ℓ factors for $\ell = 1,2,\cdots,L$, then a model is called the L-th order interaction model, where the following equation holds:

$$\begin{aligned} y_{\nu_1,\nu_2,\ldots,\nu_n} = \mu &+ \sum_{i_1\in\{1,2,\ldots,n\}} \alpha^{i_1}_{\nu_{i_1}} \\ &+ \sum_{(i_1,i_2)\in\{1,2,\ldots,n\}^2} \alpha^{i_1,i_2}_{\nu_{i_1},\nu_{i_2}} + \ldots \\ &+ \sum_{(i_1,i_2,\ldots,i_L)\in\{1,2,\ldots,n\}^L} \alpha^{i_1,i_2,\ldots,i_L}_{\nu_{i_1},\nu_{i_2},\ldots,\nu_{i_L}} \\ &+ e_{\nu_1,\nu_2,\ldots,\nu_n}, \end{aligned} \quad (2.23)$$

□

If $L = 0$, then y is represented by only a central effect μ (and a random error) . If $L = 1$, then y is represented by μ and main effects αs.

Theorem 2.4 If an experiment system is assumed to be the L-th order interaction model, then the optimum experimental design is given by $OA(M,n,s,2L)$. □

Our problem to be solved is to derive the smallest M for given n, s, and t.

3 Error-Correcting Codes(ECCs) and Orthogonal Arrays(OAs)

3.1 Properties of Orthogonal Arrays

In the following, unless mentioned explicitly, we will consider the case that $s = 2$ for simplicity. An $OA(M,n,2,\tau)$ is said to be linear if the rows of $OA(M, n,2,\tau)$ form a linear vector space. If an $OA(M,n,2,\tau)$ is linear, $OA(M,n,2,\tau)$ has a basis for the linear vector space. This basis is given in the form of $(\log_2 M) \times n$ matrix called a generator matrix.

Theorem 3.1 [HSS99] Let A be an $M \times n$ linear array with binary elements, and G be a generator matrix of A. Then A is an $OA(M,n,2,\tau)$ if and only if any τ columns of G are linearly independent over $GF(2)$. □

Theorem 3.2 [HSS99] An $M \times n$ array A with binary elements is an $OA(M, n, 2, \tau)$ if and only if

$$\sum_{\boldsymbol{v}:\text{row of A}} (-1)^{\boldsymbol{w} \cdot \boldsymbol{v}^{\mathrm{T}}} = 0,$$

for all binary vectors $\boldsymbol{w}$ of length n containing w 1's, for all w in the range $1 \leq w \leq \tau$, where the sum is over all rows $\boldsymbol{v}$ of A. □

3.2 Orthogonal Arrays and Error-Correcting Codes

Let $W_H(\boldsymbol{w})$ be the Hamming weight of a vector $\boldsymbol{w} = (w_1, w_2, \ldots, w_n)$. We consider the case of $q = 2$ as well as OAs.

C is said to be linear if C is a linear vector subspace. If C is linear, C has the dual code $C^\perp$. Let $d^\perp$ be the minimal distance of $C^\perp$. Then $d^\perp$ is said to be the dual distance of C.

Theorem 3.3 [HSS99] If C is a binary (n, k, d) code over $GF(2)$ with dual distance $d^\perp$, then the codewords of C form the rows of an $OA(M, n, 2, d^\perp - 1)$. Conversely, the rows of a linear $OA(M, n, 2, \tau)$ form an (n, k, d) linear code over $GF(2)$ with dual distance $d^\perp \geq \tau + 1$. If the OA has strength τ but not $\tau + 1$, then $d^\perp = \tau + 1$ hold. □

Example 3.1 Let $C = \{000, 011, 101, 110\}$. This is a binary $(3, 2, 2)$ code. Then $C^\perp = \{000, 111\}$, so the dual distance of C $d^\perp = 3$. Therefore, the OA corresponding to the code C, that is in Table 2.2, is an $OA(4, 3, 2, 2)$. □

3.3 Correspondence between ECCs and OAs

Let G be a $k \times n$ matrix over $GF(2)$ which generates an $OA(M, n, 2, \tau)$. Then any τ columns of G are linearly independent over $GF(2)$. While a code generated by the parity check matrix G is the dual code $C^\perp$ of the code C which is generated by the generator matrix G. From Theorems 3.1 and 3.3, we have the Table 3.1 which shows a correspondence of parameters between ECCs and OAs. Technical terms and symbols (variables) are used which are generally used in each field. If there is no confusing, we use the same symbols.

3.4 OAs from ECCs

(1) Binary Hamming codes

A binary (n, k, d) Hamming code is a class of the binary BCH codes with $d = 3$, hence its parity check matrix is given by a $(n - k) \times n$ matrix whose columns consist of all distinct non-zero vectors over $GF(2)$.

Example 3.2 A parity check matrix H of the $(7, 4, 3)$ Hamming code is given by:

$$H = \begin{bmatrix} 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.1}$$

then we have a generator matrix G:

$$G = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 \end{bmatrix} \tag{3.2}$$

A dual code $C^\perp$ of the code C is a maximum length sequence $(7, 3, 4)$ code. □

Generally, a dual code of the $(n, k, 3)$ Hamming code C is a $(n, n-k, 2^{n-k-1})$ code $C^\perp$ [HSS99]. We then have $OA(2^{n-k}, n, 2, 2)$ from the code $C^\perp$, and $OA(2^k, n, 2, 2^{n-k-1} - 1)$.

(2) RS codes

The parameters of an (n, k, d) RS code over $GF(q)$ $(q > 2)$ are given by Eq.(2.11). Since all RS code are MDS codes, $d = n - k - 1$ holds.

Theorem 3.4 [HSS99] Let code C be an $(n, \tau, n - \tau + 1)$ RS code over $GF(q)$ which forms $OA(s^\tau, n, s, \tau)$. Then the dual code $C^\perp$ is an $(n, n - \tau, \tau + 1)$ RS code which forms $OA(s^{n-\tau}, n, s, n - \tau)$, where $q = s$ is a prime power. □

4 Unequal Error Protection Codes (UEPCs) and Orthogonal Arrays with Unequal Strength (UOAs)

4.1 Orthogonal Arrays with Unequal Strength

Definition 4.1 An $M \times n$ array A with elements from $GF(s)$ is said to be an OA with s levels and strength $\boldsymbol{\tau} = (\tau_1, \tau_2, \ldots, \tau_n)$ if every $M \times \tau_i$ subarray of A, which contains i-th column of A, contains each τ_i-tuple based on $\{0, 1\}$ exactly same times as row. We will denote such an array by $OA(M, n, 2, \boldsymbol{\tau})$. Then we will call an $OA(M, n, 2, \boldsymbol{\tau})$ OA with unequal strength if the components of $\boldsymbol{\tau}$ are not mutually equal. □

When $OA(M, n, 2, (\tau_1, \tau_2, \ldots, \tau_n))$ is applied to experimental design, we can estimate the interaction effects of at most $\lfloor \frac{\tau_i}{2} \rfloor$ factors which contains i-th factor. There are many cases that UOAs reduce more numbers of experiments than OAs with equal strength. For example, let F_1, F_2 and F_3 be the factors to be included in the experiment. Suppose we know that there are the interaction effects of F_1F_2 and F_1F_3 but not F_2F_3. If an $OA(M_1, 3, 2, 4)$ is used, we can estimate not only the interaction effects of F_1F_2, F_1F_3 but F_2F_3, although we need not estimate the interaction effect of F_2F_3. On the other hand, If an $OA(M_2, 3, 2, (4, 2, 2))$ is used, we can not estimate the interaction effect of F_2F_3. Therefore, UOA can reduce the number of experiments.

4.2 UOAs and UEPCs

The separation $(d_1, d_2, \ldots, d_n)$ of linear code C is defined by

$$\begin{aligned} d_i = \min\{dist(\boldsymbol{u}, \boldsymbol{v}) \mid \boldsymbol{u} = (u_1, u_2, \ldots, u_n), \\ \boldsymbol{v} = (v_1, v_2, \ldots, v_n), \boldsymbol{u}, \boldsymbol{v} \in C, u_i \neq v_i\}, \\ for \quad i = 1, 2, \ldots, n. \end{aligned}$$

If a linear code C has the separation whose components are not mutually equal, the code C is called an unequal error protection codes. Let $(d_1^\perp, d_2^\perp, \ldots, d_n^\perp)$ be the separation of $C^\perp$ which is the dual code of C. Then we will call $(d_1^\perp, d_2^\perp, \ldots, d_n^\perp)$ the dual separation of C.

Table 3.1: Correspondence of parameters between ECCs and OAs

ECCs	OAs	Notes
# of codewords: M	# of experiments (runs): M	
Codewords:$\boldsymbol{x_1}, \boldsymbol{x_2}, \cdots, \boldsymbol{x_M}$	Array $A = (\boldsymbol{v_1}^{\mathrm{T}}, \boldsymbol{v_2}^{\mathrm{T}}, \cdots, \boldsymbol{v_M}^{\mathrm{T}})^{\mathrm{T}}$	$\boldsymbol{x} = \boldsymbol{v}$
Alphabet size: q	# of levels: s	$p = s$
Code length: n	# of factors: n	
Dual distance $d^{\perp}$	strength: τ	$\tau = d^{\perp} - 1$

Theorem 4.1 If C is a binary (n, k, d) code over $GF(2)$ with dual separation $(d_1^{\perp}, d_2^{\perp}, \ldots, d_n^{\perp})$, then the codewords of C form the row of an $OA(2^k, n, 2, (d_1^{\perp}-1, d_2^{\perp}-1, \ldots, d_n^{\perp}-1))$. Conversely, the rows of a linear $OA(M, n, 2, (\tau_1, \tau_2, \ldots, \tau_n))$ form an $(n, \log_2 M, d)_2$ linear code over $GF(2)$ with dual separation $(d_1^{\perp}, d_2^{\perp}, \ldots, d_n^{\perp})$, where $d_i^{\perp} \geq \tau_i + 1$, $i = 1, 2, \ldots, n$. If the OA has strength τ_i but not $\tau_i + 1$, $d_i^{\perp} = \tau_i + 1$ $(i = 1, 2, \ldots, n)$. □

We show two construction methods of UOAs. These are derived from UEPCs.

Construction Method 1 Let there be two generator matrices of OAs; G_1 is the generator matrix for a linear $OA(M_1, n_1, 2, \tau)$, and G_2 is the one for a linear $OA(M_2, n_2, 2, \tau')$, where $\tau' \leq \tau$. Let G_1 and G_2 be joined as submatrices of G where G_1 and G_2 overlap, as shown in Fig.4.1. The OA with generator matrix G is a $(M_1 M_2) \times (n_1 + n_2 - n_{0L})$ array. Let $n_{0L} \leq \tau'/2$.

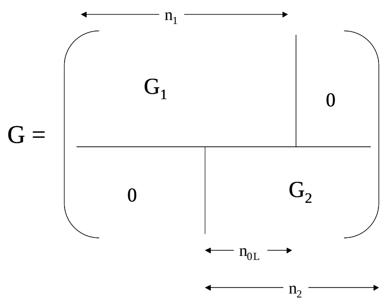


Figure 4.1: Construction method of UOA

Theorem 4.2 An OA by Construction Method 1 is an $OA(M_1 M_2, n_1 + n_2 - n_{0L}, 2, (\tau_1, \tau_2, \ldots, \tau_n))$, where

$$\begin{aligned}
&\tau_i \geq \tau \quad (i = 1, 2, \ldots, n_1 - n_{0L}),\\
&\tau_i \geq \tau' \quad (i = n_1 + 1, n_1 + 2, \ldots, n_1 + n_2 + n_{0L}),\\
&\tau_i \geq \tau + \tau' - n_{0L} \quad (i = n_1 - n_{0L} + 1, \ldots, n_1).
\end{aligned}$$

□

Construction Method 2 Let α denote a primitive element of the field $GF(2^{2m})$. Then $\beta = \alpha^{2^m+1}$ is a primitive element of the field $GF(2^m)$ which is a subfield of the field $GF(2^{2m})$. Consider an OA with 2 levels which have the generator matrix

$$G = \begin{bmatrix} 1 & \alpha & \cdots & \alpha^{2^m} & \alpha^{2^m+1} & \alpha^{2^m+2} & \cdots & \alpha^{2^{2m}-2} \\ 1 & 0 & \cdots & 0 & \beta^3 & 0 & \cdots & 0 \end{bmatrix}. \tag{4.1}$$

The OA with generator matrix G is a $2^{3m} \times (2^{2m} - 1)$ array, and its strength is at least 2.

Theorem 4.3 Let m be an odd integer. Then the OA with the generator matrix in (4.1) is an $OA(2^{3m}, (2^{2m} - 1), 2, (\tau_1, \tau_2, \ldots, \tau_n))$, where

$$\begin{aligned}
\tau_i &= 4 \quad (i = 1 + j(2^m + 1), j = 0, 1, \ldots, 2^m - 2),\\
\tau_i &\geq 2 \quad (\text{otherwise}).
\end{aligned}$$

□

5 Examples of UOAs

In this section, we show some examples of UOAs by Construction Method 1 and 2. And we compare them with optimal OAs with equal strength.

Firstly, we compare the following OAs;

- (Equal) optimal $M \times n$ OAs with 2 levels and equal strength 4 that is in [HSS99] $(n = 11, 12, \ldots, 32)$.
- (Method 1) $M \times n$ OAs with 2 levels and partially strength 4 by Construction Method 1 $(n = 11, 12, \ldots, 32)$: G_1 in Construction Method 1 is a generator matrix for an optimal $M_1 \times n_1$ linear OA with 2 levels and equal strength 3 that is in [HSS99] $(n_1 = 9, 10, \ldots, 30)$, G_2 is a generator matrix for a linear $OA(4, 3, 2, 2)$, and $n_{0L} = 1$.

The number of rows of each OA is shown in Table 5.1. Then, the number of rows of UOAs by Construction Method 1 is fewer than that of OAs with equal strength at many n's. Therefore, these UOAs can reduce more number of experiments than OAs with equal strength under partial interaction effects.

Next, we compare the following OAs;

- The OA with equal strength that has generator matrix
$$G = \begin{bmatrix} 1 & \alpha & \cdots & \alpha^{2^m+1} & \cdots & \alpha^{2^{2m}-2} \\ 1 & \alpha^2 & \cdots & \alpha^{2^{m+1}+2} & \cdots & \alpha^{2^{2m+1}-4} \end{bmatrix}.$$
This is an $OA(4096, 63, 2, 4)$. This OA is derived from BCH codes.
- The UOA with by Construction Method 2, where let $m = 3$ in Construction Method 2. This is $OA(512, 63, 2, (\tau_1, \tau_2, \ldots, \tau_{63}))$, where $\tau_i = 4$ $(i = 1 + 9j, j = 0, 1, \ldots, 6)$, $\tau_i \geq 2$ (otherwise).

Then, the number of rows of the UOA by Construction Method 2 is fewer than that of the OA with equal strength. Therefore, the UOA can reduce more number

Table 5.1: The number of rows of OAs

n	Equal	Method 1
16	256	128
17	256	128
18	256	128
19	256	256
20	512	256
21	512	256
22	512	256
23	512	256
24	1024	256
25	1024	256
26	1024	256
27	1024	256
28	1024	256
29	1024	256
30	1024	256
31	1024	256
32	1024	256

of experiments than the OA with equal strength under partial interaction effects.

6 Concluding Remarks

We have discussed the construction methods of orthogonal arrays from those of error correcting codes. The relation between them is also clarified. Although coding theory and orthogonal arrays have analogous problems, the subjects have studied almost separately. As future discussions, powerful extension to non-linear cases and mixed orthogonal effect cases are remained. An approach by projective geometry to construct orthogonal arrays is also necessary.

Acknowledgement

The author would like to thank Mr. T. Saito for his valuable support to write this paper.

This work was partially supported by Waseda University Grant for Special Research Project no:2005B-189.

References

[Gils83] W. J. Van Gils, "Two Topics on Linear Unequal Error Protection Codes: Bounds on Their Length and Cyclic Codes Classes," *IEEE Trans. Inform. Theory,* vol.IT-29, no.6, pp.866-876, Nov. 1983.

[Hira83] S.Hirasawa, "Introduction to coding theory and some applications," *Seminar on Information Systems* Taipei, R.O.C., Aug. 1983.

[Hira99] S. Hirasawa, Introduction to Coding Theory (in Japanese), Baifukan, Tokyo, 1999.

[HSS99] A. S. Hedayat, N. J. A. Sloane, and J. Stafken, *Orthogonal Arrays –Theory and Applications–*, Springer, NY, 1999.

[MS77] F. J. MacWilliams, and N. J. A. Sloane, *The Theory of Error-Correcting Codes*, North-Holland, Amsterdam, North-Holland Pub. Co., Netherlands, 1979.

[MW67] B. Masnic, and J. k. Wolf, "On linear unequal error protection codes ," *IEEE Trans. Infrom. Theory*, vol.IT-3, no.4, pp.600-607, Oct. 1967.

[PW71] W. W. Peterson, and E. J. Weldon, Jr. *Error Correcting Codes*, 2nd ed., The MIT Press, Cambridge, MA, 1971.

[SMH05] T. Saito, T. Matstushima, and S. Hirasawa, "A note on construction of orthogonal arrays with unequal strength from error-correcting codes," to appear in *IEICE Trans. Fundamentals.*

[Taka79] I. Takahashi, *Combinatorial Theory and its Applications* (in Japanese),Chapter 2, Iwanami-shoten, Tokyo, 1979.

Appendix A

We show how to calculate an estimated value of y, when complete design and orthogonal design are used.

(1) Complete design

In Example.2.1, we assume the model

$$\begin{aligned} y_{\nu_1,\nu_2,\nu_3} &= \mu + \alpha^1_{\nu_1} + \alpha^2_{\nu_2} + \alpha^3_{\nu_3} \\ &\quad + \alpha^{1,2}_{\nu_1,\nu_2} + \alpha^{1,3}_{\nu_1,\nu_3} + \alpha^{2,3}_{\nu_2,\nu_3} + e_{\nu_1,\nu_2,\nu_3}, \\ &(\nu_i \in \{0,1\},\ i \in \{1,2,3\}), \end{aligned}$$

and the output y for each experiment is given as Table 2.1. Then, by Eqs.(2.16) and (2.17),

$$\begin{aligned} &\hat{\mu} = 0.500 \\ &\hat{\alpha}^1_0 = -0.225, \quad \hat{\alpha}^2_0 = 0.125, \quad \hat{\alpha}^3_0 = 0.100, \\ &\hat{\alpha}^1_1 = 0.225, \quad \hat{\alpha}^2_1 = -0.125, \quad \hat{\alpha}^3_1 = -0.100. \end{aligned}$$

and by Eq.(2.18),

$$\begin{aligned} &\hat{\alpha}^{1,2}_{0,0} = 0.000, \quad \hat{\alpha}^{1,3}_{0,0} = -0.025, \quad \hat{\alpha}^{2,3}_{0,0} = 0.025, \\ &\hat{\alpha}^{1,2}_{1,0} = 0.000, \quad \hat{\alpha}^{1,3}_{1,0} = 0.025, \quad \hat{\alpha}^{2,3}_{1,0} = -0.025, \\ &\hat{\alpha}^{1,2}_{0,1} = 0.000, \quad \hat{\alpha}^{1,3}_{0,1} = 0.025, \quad \hat{\alpha}^{2,3}_{0,1} = -0.025, \\ &\hat{\alpha}^{1,2}_{1,1} = 0.000, \quad \hat{\alpha}^{1,3}_{1,1} = -0.025, \quad \hat{\alpha}^{2,3}_{1,1} = 0.025. \end{aligned}$$

And, the estimated value of y is as follows.

$$\begin{aligned} \hat{y}_{\nu_1,\nu_2,\nu_3} &= \hat{\mu} + \hat{\alpha}^1_{\nu_1} + \hat{\alpha}^2_{\nu_2} + \hat{\alpha}^3_{\nu_3} \\ &\quad + \hat{\alpha}^{1,2}_{\nu_1,\nu_2} + \hat{\alpha}^{1,3}_{\nu_1,\nu_3} + \hat{\alpha}^{2,3}_{\nu_2,\nu_3}, \\ &(\nu_i \in \{0,1\},\ i \in \{1,2,3\}). \end{aligned}$$

(2) Orthogonal design

In Example.2.1, we assume the model

$$\begin{aligned} y_{\nu_1,\nu_2,\nu_3} &= \mu + \alpha^1_{\nu_1} + \alpha^2_{\nu_2} + \alpha^3_{\nu_3} + e_{\nu_1,\nu_2,\nu_3}, \\ &(\nu_i \in \{0,1\},\ i \in \{1,2,3\}). \end{aligned}$$

and the output y for each experiment is given as Table 2.2. Then by Eqs.(2.19) and (2.20),

$$
\begin{aligned}
&\hat{\mu} = 0.500 \\
&\hat{\alpha}_0^1 = -0.200, \quad \hat{\alpha}_0^2 = 0.100, \quad \hat{\alpha}_0^3 = 0.100, \\
&\hat{\alpha}_1^1 = 0.200, \quad \hat{\alpha}_1^2 = -0.100, \quad \hat{\alpha}_1^3 = -0.100.
\end{aligned}
$$

And, the estimated value of y is as follows.

$$
\begin{aligned}
\hat{y}_{\nu_1,\nu_2,\nu_3} &= \hat{\mu} + \hat{\alpha}_{\nu_1}^1 + \hat{\alpha}_{\nu_2}^2 + \hat{\alpha}_{\nu_3}^3 \\
&(\nu_i \in \{0,1\},\ i \in \{1,2,3\}).
\end{aligned}
$$

資訊網路技術研討會

(1)電腦網路系統

Computer Network System

主講人：八星禮剛 (R. Yatsuboshi)

日本早稻田大學工學博士

富士通公司系統研究所主任

(2)編碼理論及應用

Coding Theory and Its Application

主講人：平澤茂一 (S. Hirasawa)

日本大阪大學工學博士

日本早稻田大學教授

主辦：中華民國電腦學會

教育部

協辦：中國工程師學會青年工程師勵進會

地點：中國工程師學會（仁愛路二段一號三樓）

時間：中華民國72年8月30日～9月1日

CODING THEORY AND ITS RECENT TOPICS

Shigeichi Hirasawa

Department of Industrial Engineering and Management
School of Science and Engineering
Waseda University
3-4-1, Ohkubo, Shinjuku
Tokyo, 160 JAPAN

This lecture note was published in the text book at the Seminar on Information Systems, which was held at Taipei, Taiwan from Aug. 30 through Sep. 1, 1983, sponsored by the Ministry of Education, Taiwan and by the Information Processing Society of Taiwan, R.O.C.

Coding Theory and Its Recent Topics

Shigeichi Hirasawa

Waseda University
3-4-1, Ohkubo, Shinjuku
Tokyo, 160 JAPAN

Summary

In this lecture, coding theory, information theory, and their applications are reviewed. This lecture note is composed of two parts. Part I is a review of coding theory as tutorials. The algebraic structures for linear codes are focused. Then applications of error correcting code to computer storage systems are given. In Part II, a survey on concatenated codes and product code is described connecting with information theory, especially Shannon's channel coding theorem. Recent topics on both codes and their generalizations are also given along with the discussions.

The table of contents of this lecture note is shown as follows:

Contents

An Introduction to Coding Theory and Some Applications

Shigeichi Hirasawa*

Abstract

In this note, discussed are an introduction to coding theory as tutorials. The most important class of codes, linear codes are described by using matrix representation. Bounds on error correction capabilities of linear codes are also shown. Applications of error-correcting codes to computer main storage systems are given as a practical example.

* Professor, Department of Industrial Engineering and Management, School of Science and Engineering, Waseda University, 3-4-1, Ohkubo, Shinjuku, Tokyo 160 JAPAN

I. Introduction

Information theory was established by Shannon[1] for the study of quantitative arguments of information, and gave an impact to research areas on coding schemes as coding theory. Although both information theory and coding theory deal with fundamental problem of channel coding system which achieves reliable communication over noisy channels, there is a significant difference between them; clearly different approaches were made. The former has statistical view-point, while the latter, constructive and combinatorial view-point. Actually, coding theory has been developed strongly supported by modern algebra.

Coding theory has given us a lot of efficient error correcting codes and their decoding methods [2],[3],[4],[5],[6],[7],[8]. The main subject of codes is to correct errors over noisy channels. The channel might be deep space communication link, a satellite communication link or telephone line. Since some kind of electromagnetic waves and noise on the reading and writing head of the tape would cause to error, the output of the channel (received data) is different from the input of the channel (transmitted data). Therefore main storage medium such as constracted by LSI (Large Scale Integration) memory and external storage medium such as magnetic tape can also be considered to be the channel. A model of coding and decoding system is shown in Fig.1.1, where we assume that the channel is binary symmetric with cross-over probability p as also shown in the Figure. This model can be rewritten for main storage system as shown in Fig.1.2.

The main functions of the coding and decoding system are summarized as follows;

(i) the reliability of the system
; measured by the probability of decoding error P_e,

(ii) the efficiency of the system
; measured by the code rate r,

(iii) the cost of the system
; measured by the decoding complexity χ.

In Section II of this note, as preliminary consideration, some notations, definitions, and concepts used for coding theory are shown. As the most important class of codes, linear codes are discussed in Section III. Error correction capability of codes are reviewed as bounds in Section IV. Some applications of error correcting codes to computer storage systems are described in Section V. Section VI is comments for further studies.

Although coding theory has two main class of codes; one is called block codes and the other, convolutional codes. In this note, however, we restrict our discussions to only the block code. We assume that symbols from the source and to the sink are binary.

II. Preliminaries

In this section, we introduce notations and definitions for coding theory.

Consider two binary vectors $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ of length n such that

$$\boldsymbol{x}_i=(x_1^{(i)}, x_2^{(i)}, \ldots, x_n^{(i)}), \tag{2.1.a}$$

$$\boldsymbol{x}_j=(x_1^{(j)}, x_2^{(j)}, \ldots, x_n^{(j)}), \tag{2.1.b}$$

where

$$x_m^{(i)},\ x_m^{(j)} \in GF(2),\ m=1,\ 2,\ \ldots,\ n.$$

Definition 2.1 [Hamming distance]: The Hamming distance $d_H(.,.)$ between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ is defined by

$$d_H(\boldsymbol{x}_i,\boldsymbol{x}_j)=\sum_{m=1}^{n} d_H(x_m^{(i)},x_m^{(j)}), \tag{2.2}$$

where

$$d_H(a,b)=\begin{cases} 0, & a=b; \\ 1, & a\neq b. \end{cases} \tag{2.3}$$

The Hamming distance is the number of position in which the vectors differ among n symbols. It is also defined by (2.3) for the non-binary code. The other distance function, Lee distance has been also used in coding theory. Above two distance measures coincide in the binary case.

Definition 2.2: A binary (n,k,d) code is a set of M binary vectors $\boldsymbol{x}_i$, i=1,2,...,M (see Fig.2.1), where $M=2^k$, $\boldsymbol{x}_i$'s are codewords, n is the code length, k is the number of information symbols, and d is the minimum distance of the code defined by

$$d=\min_{1\le i,j\le M} d_H(\boldsymbol{x}_i,\boldsymbol{x}_j),\ i\neq j. \tag{2.4}$$

The rate r is defined by

$$r=k/n. \tag{2.5}$$

Example 2.1: The even (or odd) parity code shown in Fig.2.2 is the (9,8,2) code.

Example 2.2: The repetition code of length n is the (n,1,n) code;

for n=5, 0 0 0 0 0

1 1 1 1 1 .

Example 2.3: The r out of n code is shown in Fig.2.3 for n=5 and r=2, where $M=\binom{n}{r}$.

Definition 2.3: The weight $w_H(.)$ of $\boldsymbol{x}_i$ is the number of nonzero symbols $x_m^{(i)}$ among n symbols:

$$w_H(\boldsymbol{x}_i)=\sum_{m=1}^{n} w_H(x_m^{(i)}) \tag{2.6}$$

where

$$w_H(a)=\begin{cases} 0, & a=0; \\ 1, & a=1,\ (a\neq 0) \end{cases} \tag{2.7}$$

From definition 2.1, we can easily get

$$d_H(\boldsymbol{x}_i,\boldsymbol{x}_j)=w_H(\boldsymbol{x}_i-\boldsymbol{x}_j). \tag{2.8}$$

Example 2.4:

$d_H(01001011,01110010)=w_H(00111001)=4.$

Theorem 2.1: Consider an (n,k,d) code. Then the code can detect all patterns of d-1 or fewer errors. Similarly, the code can correct all patterns of t or fewer errors, where d is at least 2t+1. (See Fig.2.4)

From Theorem 2.1, we see that the (n,k,d) code corrects all patterns of t or fewer errors and simultaneously detects those of d' or fewer errors, where $d'\geq t$, $d=t+d'+1$. The even parity code of Example 2.1 can detect one error, exactly speaking all odd number of errors. The repetetion code of Example 2.2 for length n=5 can correct 2 errors. Next, we shall discuss the decoding rule.

Definition 2.4: Let all of the codewords $\boldsymbol{x}_i$, $i=1,2,\ldots,M$ of the (n,k,d) code be used equally likely. And let $\boldsymbol{x}_i$ be transmitted and $\boldsymbol{y}$ be received. Then the maximum likelihood decoding (MLD) algorithm is to decode $\boldsymbol{y}$ into $\boldsymbol{x}_j$ by finding $\boldsymbol{x}_j$ such that

$$\max_j \Pr(\boldsymbol{y}|\boldsymbol{x}_j).$$

If $i\neq j$, a decoding error occurs. If binary symmetric channel (BSC) with p is assumed, then MLD algorithm is equivalent to the minimum distance decoding (MDD) algorithm: Let

$$d_H(\boldsymbol{x}_j,\boldsymbol{y})=e, \tag{2.9}$$

then the probability that $\boldsymbol{x}_j$ is transmitted and $\boldsymbol{y}$ is received, $\Pr(\boldsymbol{y}|\boldsymbol{x}_j)$ is given by

$$\Pr(\boldsymbol{y}|\boldsymbol{x}_j)=p^e(1-p)^{n-e}. \tag{2.10}$$

Therefore, if e is minimized Pr(.) is maximized. Thus we have MDD algorithm such that decode $\boldsymbol{y}$ into $\boldsymbol{x}_j$, where $\boldsymbol{x}_j$ is

the closest codeword from $\boldsymbol{y}$. Usual algebraic decoding algorithm, however, is to find $\boldsymbol{x}_j$ from $\boldsymbol{y}$, if $d_H(\boldsymbol{x}_j,\boldsymbol{y})\leq[(d-1)/2]$[1]. This is called the bounded distance decoding (BDD) algorithm.

III. Linear Codes

In this section, assuming that the code is linear, we shall describe the algebraic code structures by using matrices.

3.1. Properties of Linear Codes

Let F be the field composed of two elements, i.e., GF(2), where the addition and multiplication for GF(2) are shown in Fig.3.1.

Definition 3.1: An (n,k,d) linear code is a linear subspace of $F^n=\{0,1\}^n$.

If $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are codewords of a linear code, then $\boldsymbol{x}_i+\boldsymbol{x}_j$ should also be a codeword, since all set of codeword is a subspace.

Theorem 3.1: The minimum distance for a linear code equals the minimum weight of the nonzero codewords.

Example 3.1: The codes given by Example 2.1 and 2.2 are linear but that by Example 2.3 is not linear.

3.2. Generator Matrix and Parity Check Matrix

If the dimension of the linear subspace is k, we can pick k linearly independent codewords from the (n,k,d) linear code. We let these codewords be $\boldsymbol{x}_1,\boldsymbol{x}_2,\ldots,\boldsymbol{x}_k$. Then we have a generator matrix G such that

$$G=\begin{bmatrix}\boldsymbol{x}_1\\ \boldsymbol{x}_2\\ \vdots\\ \boldsymbol{x}_k\end{bmatrix}, \tag{3.1}$$

1. [x] denotes the greatest integer less than of equql to x.

where the code is the row space of G, and G is a $k \times n$ matrix of rank k. By usual matrix manipulations, we have the following theorem.

Theorem 3.2: A generator matrix G of an (n,k,d) linear code can be given by the canonical form:

$$G = [\ I\ ,\ P\ ], \tag{3.2}$$

where I is a $k \times k$ unit matrix, and P is a $k \times (n-k)$ matrix.

Example 3.2: The generator matrices for the codes given by Example 2.1 and 2.2 are shown in Fig.3.2.

Let the data sequence of length k to be encoded be $\mathbf{v}$, then the corresponding codeword $\mathbf{x}$ is given by

$$\mathbf{v} = (v_1,\ v_2,\ \ldots,\ v_k), \tag{3.3}$$

$$\begin{aligned} \mathbf{x} &= \mathbf{v}G \\ &= (x_1,\ x_2,\ \ldots,\ x_n), \end{aligned} \tag{3.4}$$

where $v_m = x_m$, $m=1,2,\ldots,k$, since G is composed of the $k \times k$ unit matrix I. A code of this type is called a systematic code as shown in Fig.3.3.

Definition 3.2: Two codes are equivalent, if the difference between them is only in the order of their coordinates.

Theorem 3.3: Every linear code is equivalent to a systematic code.

Note that equivalent codes have the same capability of error-correction. So we use the form of (3.2) as G. If V is an (n,k,d) linear code, then V is the null space of H given by the following theorem, where $GH^T = 0$.

Theorem 3.4: A parity check matrix H of an (n,k,d) linear code can be given by

$$H = [\ -P^T\ ,\ I\ ], \tag{3.5}$$

where I is an $(n-k) \times (n-k)$ unit matrix and P^T is a transposed $(n-k) \times k$ matrix of P given by (3.2).

We can show that if G is the generator matrix for V, then H is that for $V^{\perp}$, where $V^{\perp}$ is the dual code of V. As above discussion, we can get the codeword $\mathbf{x}$ by (3.4). Let us now decode the received sequence $\mathbf{y}$ into some codeword from the (n,k,d) code by using the parity check matrix H.

Definition 3.3: The syndrome $\mathbf{s}$ of $\mathbf{y}$ is given by

$$\mathbf{s} = \mathbf{y}H^T. \tag{3.6}$$

Letting $\mathbf{x}$ be transmitted and a noise vector $\mathbf{e}$ be added by the channel, then we have

$$\mathbf{y} = \mathbf{x} + \mathbf{e}, \tag{3.7}$$

where

$$\mathbf{e} = (e_1,\ e_2,\ \ldots,\ e_n),\ e_m \in GF(2),\ m=1,\ 2,\ \ldots,\ n. \tag{3.8}$$

Thus

$$\mathbf{s} = \mathbf{y}H^T = (\mathbf{x}+\mathbf{e})H^T = \mathbf{e}H^T. \tag{3.9}$$

From (3.8), we can interpret $\mathbf{s}$ as the sum of the columns of H.

Example 3.3: Letting G and H be given by

$$G = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}, \tag{3.10}$$

$$H = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}, \tag{3.11}$$

where

$$P = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{bmatrix}, \tag{3.12}$$

we have

$$\mathbf{s}=(0\ 0\ 0) \quad \text{if } e=(0,0,0,\ldots,0),$$
$$\mathbf{s}=(1\ 1\ 0) \quad \text{if } e=(1,0,0,\ldots,0), \tag{3.13}$$
$$\mathbf{s}=(1\ 0\ 1) \quad \text{if } e=(0,1,0,\ldots,0),$$

and so on. Therefore $\mathbf{s}$ equals the m-th column of H, where $e_m=1$.

Next, we now state an important theorem which gives the method for choosing the columns of the parity check matrix H to obtain the minimum distance d.

Theorem 3.5: Consider an (n,k,d) binary linear code. Then its parity check matrix H is an $(n-k)\times n$ matrix for which any d-1 or fewer columns are lineary independent over GF(2).

The proof of this theorem can be understood from the derivation of Theorem 4.2 [G-V bound] described later. However, we can easily proof it by the fact that if the sum of d columns of H is zero, then there is a codeword of weight d, since $\mathbf{x}H^T=0$, and the code is linear, where the weight of a nonzero codeword $\mathbf{x}$ of the (n,k,d) code is at least d. This suggests us that any d-1 or fewer error vector $\mathbf{e}$ satisfies $\mathbf{e}H^T\neq 0$; hence we can correct errors from syndrome $\mathbf{s}=\mathbf{e}H^T$, if $\mathbf{s}$'s are all distinct for corresponding error vectors $\mathbf{e}$'s.

Finally, we give the following theorem to evaluate the performance of the code.

Theorem 3.6: Let an (n,k,d) code be decoded by the algorithm such that the decoder corrects any t or fewer errors; otherwise the decoder will either fail to decode or decode in error. Then the probability of correct decoding P_c over the binary symmetric channel with cross-over probability p is given by

$$P_c=\sum_{i=0}^{t}\binom{n}{i}p^i(1-p)^{n-i}, \tag{3.14}$$

where $t=[(d-1)/2]$.

3.3. Hamming Codes

The binary Hamming code is the most important and the best known code [9].

Definition 3.4: Consider a parity check matrix H which has m rows and 2^m-1 columns, where the column vectors are all possible patterns of length m except the all 0 patterns. The (n,k,d) Hamming code is thus defined by H.

Theorem 3.7: The (n,k,d) Hamming code is capable of correcting all single error, where the parameters are given as follows:

$$n=2^m-1, \tag{3.15.a}$$
$$k=2^m-m-1, \tag{3.15.b}$$
$$d=3. \tag{3.15.c}$$

Example 3.4: The parity check matrix H of (7,4,3) Hamming code is given by

$$H=\begin{bmatrix} 1&0&1&0&1&0&1\\ 0&1&1&0&0&1&1\\ 0&0&0&1&1&1&1 \end{bmatrix}, \tag{3.16}$$

where the columns of H are rearranged so that the syndrome shows the location of an error. For example, if e=(0001000), then s=(001).

Corollary 3.1: The (n,k,d) modified (or extended) Hamming code is a single-error-correcting, double-error-detecting code, whose parameters are given by

$$n=2^m, \tag{3.17.a}$$
$$k=2^m-m-1, \tag{3.17.b}$$
$$d=4, \tag{3.17.c}$$

where the code can be obtained by adding to the original Hamming code one parity check symbol.

IV. Error-Correction Capability Bounds

In this section, we shall show some of upper and lower bounds on error-correction capability.

If $d \geq 2t+1$, the code is capable of correct all patterns of t or fewer errors. The number of patterns within t or fewer errors for each of the M codewords is given by

$$1+\binom{n}{1}+\binom{n}{2}+\ldots+\binom{n}{t},$$

thus

$$M \leq 2^n / \sum_{i=0}^{t} \binom{n}{i}. \tag{4.1}$$

Letting $M=2^k$, we have the following theorem.

Theorem 4.1 [Hamming bound]: Any $(n,k,2t+1)$ code satisfies

$$n-k \geq \log\left[\sum_{i=0}^{t} \binom{n}{i}\right]. \tag{4.2}$$

An asymptotic formula for (4.2) as $n \to \infty$, is given by

$$1-k/n \geq H(t/n), \tag{4.3}$$

where we have used [2]

$$\lim_{n\to\infty} \frac{1}{n}\log \sum_{i=0}^{j} \binom{n}{i} = H(j/n). \tag{4.4}$$

Definition 4.1: If a code satisfies (4.2) with equality, then the code is called perfect code.

Only the three following linear perfect códes are known:

(i) The repetition code [See Example 2.2],

(ii) The Hamming code [See Theorem 3.5],

(iii) The (23,12,7) Golay code [10] [See following Example 4.1].

Example 4.1: Note that

$$\binom{23}{0}+\binom{23}{1}+\binom{23}{2}+\binom{23}{3}=2^{11}, \tag{4.5}$$

and P^T of the parity check matrix is shown in Fig.4.1.

From (3.8), letting $e^{(i)}$ be defined by

$$e^{(i)}=e, \tag{4.6}$$

$$\begin{cases} e_i=1; \\ e_m=0,\ m=1,2,\ldots,n,\ m\neq i, \end{cases} \tag{4.7}$$

we have the following equations to obtain the G-V bound of Theorem 4.2.

(i) First, select arbitrarily the first nonzero column of H.

(ii) Second, it is possible to choose the second column of H such that

$$e^{(2)}H^T \neq 0, \tag{4.8.a}$$

$$e^{(2)}H^T \neq e^{(1)}H^T, \tag{4.8.b}$$

if $2^{n-k} > 2$.

(iii) Next, it is also possible to choose the third column of H such that

$$e^{(3)}H^T \neq 0, \tag{4.9.a}$$

$$e^{(3)}H^T \neq e^{(2)}H^T, \tag{4.9.b}$$

$$e^{(3)}H^T \neq (e^{(1)}+e^{(2)})H^T, \tag{4.9.c}$$

if $2^{n-k} > 4$.

(iv) Finally, it is also possible to choose the n-th column of H such that

$$e^{(n)}H^T \neq 0, \tag{4.10.a}$$

$$e^{(n)}H^T \neq e^{(i_1)}H^T,\ i_1=1,2,\ldots,n-1; \tag{4.10.b}$$

$$e^{(n)}H^T \neq (e^{(i_1)}+e^{(i_2)})H^T,\ i_1,i_2=1,2,\ldots,n-1,\ i_1\neq i_2; \tag{4.10.c}$$

$$e^{(n)}H^T \neq (e^{(i_1)}+e^{(i_2)}+e^{(i_3)})H^T,\ i_1,i_2,i_3=1,2,\ldots,n-1,\ i_1\neq i_2,\ i_2\neq i_3,\ i_3\neq i_1; \tag{4.10.d}$$

$$\vdots$$

$$e^{(n)}H^T \neq (e^{(i_1)}+e^{(i_2)}+\ldots+e^{(i_{d-2})})H^T,\ i_1,i_2,\ldots,i_{d-2}=1,2,\ldots,n-1, \tag{4.10.e}$$

where $i_1, i_2, \ldots, i_{d-1}$ are all different from each other, if

$$2^{n-k} > 1+\binom{n-1}{1}+\binom{n-1}{2}+\ldots+\binom{n-1}{d-2}, \qquad (4.10.f)$$

since any d-1 or fewer columns of H for an (n,k,d) code are linearly independent. This is inversely the proof of Theorem 3.5. Thus we have the following theorem.

Theorem 4.2 [Gilbert-Varsharmov (G-V) bound]: It is possible to construct an (n,k,d) code which satisfies

$$n-k > \log\left[\sum_{i=0}^{d-2}\binom{n-1}{i}\right]. \qquad (4.11)$$

As the similar formula to (4.3), we have

$$1-k/n \geq H(d/n), \qquad (4.12)$$

where we have used (4.4) and

$$H[(d-2)/(n-1)] \simeq H(d/n), \qquad (4.13)$$

as $n \to \infty$.

Letting d=3, or t=1, we get $2^{n-k} \geq 1+n$ from Theorem 4.1 and simultaneously $2^{n-k} > n$ from Theorem 4.2. Therefore, $n \leq 2^{n-k}-1$ is the necessary and sufficient condition for t=1, i.e., it coincides (3.15.a) with equality, since Hamming code is a perfect code.

These bounds together with another bounds are shown in Fig.4.2.

V. Applications to Computer Storage System

The (n,k,d) modified Hamming codes are widely used for main storage system as single-error-correcting, double-error-detecting (SEC/DED) code. Since usually k is chosen to be a multiple of 8 (=byte), the codes are shortened versions of the modified Hamming code. By making the s leading information symbols identically 0 and omitting them from codewords, we always have an (n-s,k-s,d) code from an (n,k,d) code. Thus

we usually use

the (22,16,4) code,
the (39,32,4) code,
and the (72,64,4) code.

Next, letting the (22,16,4) code be chosen as an example, we shall describe on the encoding and decoding process. Note that such encoder and decoder have already been available by a LSI (e.g., Am 2960, AMD Inc.[11]), which uses combinatorial circuit because of high seed operation required for correcting errors (e.g., the time required for detecting errors is only 30ns, and that for correcting errors, 50ns). Furthermore, it is possible to get encoder and decoder for long code by cascaded connection of LSI's.

The generator matrix G of the LSI is given by

$$G=[\ I,\ P\ ] \qquad (5.1)$$

where P is shown in Fig.5.1. The parity check H of this code can be easily obtained from (3.5). Thus, as we have already shown as Examples 3.3 and 3.4, the syndrome can give information where the error occurs. The error correction table from the syndrome is shown in Table 5.1.

Finally, we shall show an error characteristics of a Dynamic RAM as an example [12]. Table 5.2. shows typical error rate.

Random error correcting codes are used in main storage. On the other hand, we can usually observe a characteristic of a burst channel for external storage medium, such as a magnetic disc and a magnetic tape. Therefore, burst error correcting codes such as Fire codes are used for external storage systems.

VI. Comments for Further Studies

This note is only an introduction, or the one of the shortest course to coding theory. There are many other important codes, such as BCH codes, Reed-Solomon codes, Goppa codes, and so on. For further studies, modern algebra as a tool for researches are necessary; theory of group, ring, ideal, field and especially of Galois field. Peterson's book [2] is a little bit old but is still useful guide to coding theory. The other books such as [3],[4],[5],[6],[7], and [8] are also recommended for reading. Recent surveies [13],[14] are interesting for practical use.

References

[1] C.E. Shannon, "A mathematical theory of communication," Bell syst. Tech. J., Vol.27, pp.379-423, July 1948.

[2] W.W. Peterson, Error correcting codes. 1st Ed. MA: The M.I.T. Press, 1961.

[3] S. Lin, An introduction to error-correcting codes. Englewood Cliffs, New Jersey: Prentice-Hall Inc., 1970.

[4] W.W. Peterson and E.J. Weldon,Jr., Error correcting codes. 2nd Ed. MA: The M.I.T. Press, 1972.

[5] E.R. Berlekamp, Algebraic coding theory. NY: McGraw-Hill Book Co., 1968.

[6] N.J.A. Sloane, A short course on error correcting codes. 3rd Printing, CISM Courses and Lectures No.188, Udline, Italy, 1975.

[7] H. Miyakawa et al., Coding theory. (in Japanese), Tokyo: Shokodo Inc., 1973.

[8] T. Kasami et al., Coding theory. (in Japanese), Tokyo: Corona Inc., 1975.

[9] R.W. Hamming, "Error detecting and error correcting codes," Bell syst. Tech. J., Vol.29, pp.147-160, Apr. 1950.

[10] M.J.E. Golay, "Note on digital coding," Proc. IRE, Vol.37, p.657, June 1949.

[11] "Am 2960 Fast error detection and correction for memories," The Am 2960 family data book, pp.2/312-2/327, AMD Inc., CA. 1979.

[12] "Am 2960 Boots memory reliability,' AMD Tech. Rep., Jan. 1980.

[13] V.K. Bhargava, "Forward error correction schemes for digital communications," IEEE Comm. Magazine, pp.11-19, Jan. 1983.

[14] E.R. Berlekamp, "The technology of error-correcting codes," Proc. IEEE, Vol.68, pp.564-593, May 1980.

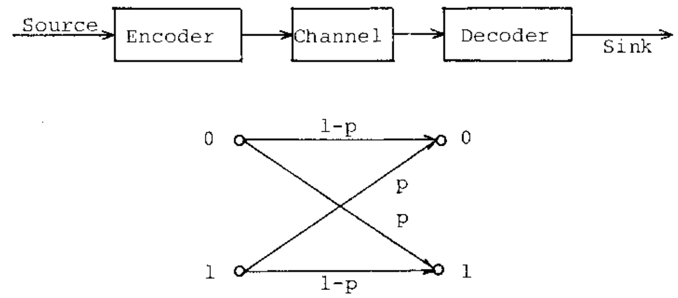
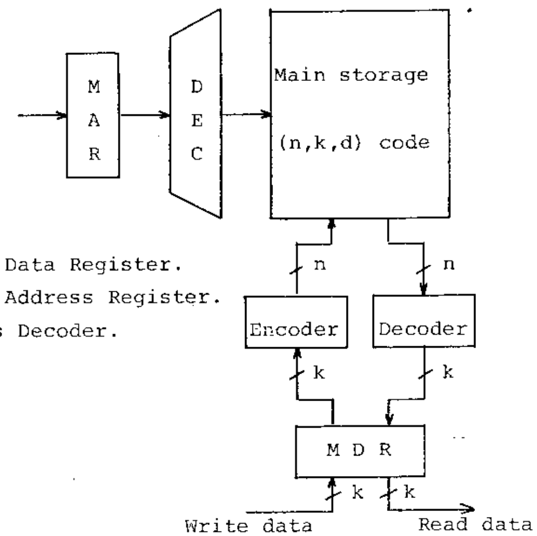


Fig.1.1. A model of coding and decoding system.



MDR: Memory Data Register.
MAR: Memory Address Register.
DEC: Address Decoder.

Fig.1.2. A model of main storage system.

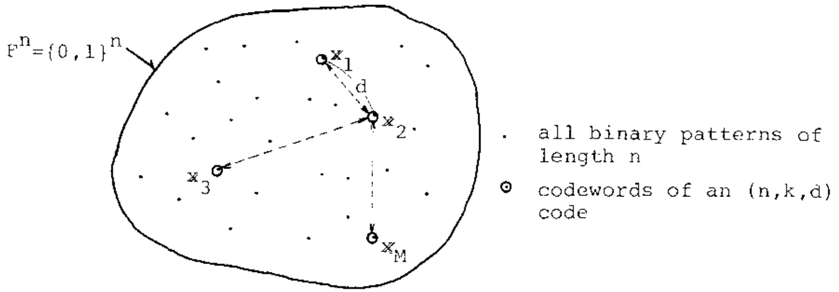


Fig.2.1. Concept of code.

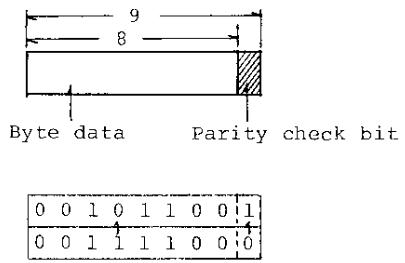


Fig.2.2. Even parity check code (Even weight code).

11000
00011
00101
00110
01001
01010
01100
10001
10010
10100

n=5, r=2

Fig.2.3. Example of r out of n code (Constant weight code).

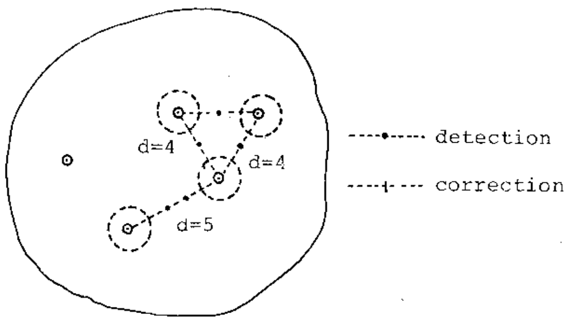


Fig.2.4. Error detection and correction.

+	0	1
0	0	1
1	1	0

·	0	1
0	0	0
1	0	1

Fig.3.1. Addition and multiplication for GF(2).

(i) Parity check code: $G = \begin{bmatrix} 100000001 \\ 010000001 \\ 001000001 \\ 000100001 \\ 000010001 \\ 000001001 \\ 000000101 \\ 000000011 \end{bmatrix}$

(ii) Repetition code: $G = [1,1,1,1,1]$

Fig.3.2. Generator matrices for examples.

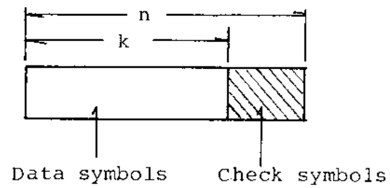


Fig.3.3. Codeword of systematic code.

$$\begin{bmatrix}
1&0&0&1&1&1&0&0&0&1&1&1\\
1&0&1&0&1&1&0&1&1&0&0&1\\
1&0&1&1&0&1&1&0&1&0&1&0\\
1&0&1&1&1&0&1&1&0&1&0&0\\
1&1&0&0&1&1&1&0&1&1&0&0\\
1&1&0&1&0&1&1&1&0&0&0&1\\
1&1&0&1&1&0&0&1&1&0&1&0\\
1&1&1&0&0&1&0&1&0&1&1&0\\
1&1&1&0&1&0&1&0&0&0&1&1\\
1&1&1&1&0&0&0&0&1&1&0&1\\
0&1&1&1&1&1&1&1&1&1&1&1
\end{bmatrix}$$

Fig.4.1. P^T of parity check matrix of the (23,12,7) Golay code [7].

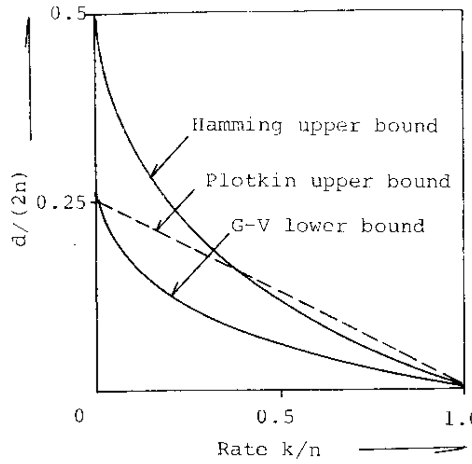


Fig.4.2. Bounds on minimum distance (asymptotic distance ratio)[2].

$$P=\begin{bmatrix} 0&1&1&1&0&0\\ 1&1&0&1&0&0\\ 1&1&0&0&1&0\\ 1&0&0&0&1&0\\ 0&1&1&0&1&0\\ 1&0&0&1&1&0\\ 0&1&0&1&1&0\\ 0&0&1&1&1&0\\ 1&1&0&0&0&1\\ 1&0&1&0&0&1\\ 0&1&1&0&0&1\\ 1&0&0&1&0&1\\ 0&1&0&1&0&1\\ 0&0&1&1&0&1\\ 1&0&0&0&1&1\\ 0&0&1&0&1&1 \end{bmatrix}$$

Fig.5.1. The matrix of P[10].

Table 5.1.
The error correction procedure from the syndrome [11].

s_6	0	1	0	1	0	1	0	1
s_5	0	0	1	1	0	0	1	1
$s_1 s_2 s_3$ \ s_4	0	0	0	0	1	1	1	1
0 0 0	*	22	21	T	20	T	T	M
0 0 1	19	T	T	16	T	14	9	T
0 1 0	18	T	T	M	T	13	7	T
0 1 1	T	11	5	T	1	T	T	M
1 0 0	17	T	T	15	T	12	6	T
1 0 1	T	10	4	T	M	T	T	M
1 1 0	T	8	3	T	2	T	T	M
1 1 1	M	T	T	M	T	M	M	T

* - no errors detected.
Number - the location of the single bit-in-error.
T - two errors detected.
M - three or more errors detected.

$s=(s_1,s_2,\ldots,s_6)$.

Table 5.2.
Error characteristics of Dynamic RAM [12].

Bits/chip	Typical error rate (%/10^3Hr)	
	Soft	Hard
1K	0.001	0.0001
4K	0.02	0.002
16K	0.10	0.011
64K(predicted)	0.13	0.016

Soft errors: non-repeating, single-bit error.
Hard errors: permanent error.