System Evaluation of Disk Allocation Methods for Cartesian Product Files by using Error Correcting Codes

Shigeichi Hirasawa*, Tomohiko Saito†, Hiroshige Inazumi‡ and Toshiyasu Matsushima§
*Faculty of Information Technology and Business, Cyber University
1-11, 4F, KitaYamabushi-cho, Shinjuku-ku, Tokyo 162-0853 Japan
Email: shigeichi_hirasawa@cyber-.ac.jp
†College of Science and Engineering, Aoyama Gakuin University, Sagamihara, Kanagawa 229-8558, Japan
‡School of Social Informatics, Aoyama Gakuin University, Sagamihara, Kanagawa 229–8558, Japan
§School of Fundamental Science and Engineering, Waseda University, Tokyo 169-8555, Japan

***Abstract*—We discuss disk allocation methods for Cartesian product files by introducing error correcting codes, and have clarified the performance of the methods by system evaluation models developed by using rate distortion theory. Let us assume q^n Cartesian product files with n attributes and q actual values in each attribute, and store q^n files into $G(\leq q^n)$ disks. For a partial match access request, we represent new disk allocation methods which able to access the disks in parallel as much as possible, where the partial match access request includes an indefinite case (don't care: "$*$") in some attributes and the $*$ requires to access the files with corresponding to the attribute for the all actual attribute values. In this paper, we propose to apply unequal error protection codes to the case where the probabilities of occurrence of the $*$ in the attributes for a partial match access request are not the same. We show the disk allocation methods have desirable properties as n becomes large.**

***Index Terms*—disk allocation, Cartesian product files, error correcting codes, unequal error protection codes, system evaluation model, flexible, elastic, rate distortion theory, Chernoff bound**

I. Introduction

In the latter decade of 1970's, J. Pearl and A. Crolotte have discussed on the trade-off between amount of memory and error in QA (Question Answering) systems based on rate-distortion theory [9]. They clarified the conditions such that we can reduce the amount of memory, if the small error rate can be tolerated. These conditions are called "flexible" or "elastic". We have extended these conditions to be more useful by generalized trade-off evaluation model, called system evaluation model [4], [5] and have applied it to various information systems [6], [7]. By using this model, we can decide whether the system has "effective elastic" condition or not, where the "effectively elastic" implies the relatively effective property as the system size becomes large.

On the other hand, it is well known that the structure of linear error correcting codes can be used to experimental design [3], and disk allocation for files [1]. Let us assume the q^n Cartesian product files with n attributes and q actual values in each attribute, and store q^n files into $G(\leq q^n)$ disks. For a partial match access request (PMAR), we intend to access the disks in parallel as much as possible by using disk allocation methods, where the PMAR includes an indefinite case (don't care: "$*$") in some attributes and the $*$ requires to access the files for corresponding attribute to the all actual attribute values.

In this paper, first we apply the system evaluation model to the disk allocation methods, and discuss whether the disk allocation methods satisfy the (effective) elastic condition or not. Next, we extend them into the case where the probability of the occurrence of the $*$ at i-th attribute X_i, $\Pr(z_i = *)$ is not uniform for i, where z_i is the i-th actual attribute value. We propose new disk allocation methods based on unequal error protection (UEP) codes for PMAR, and we show that the proposed methods are superior compared to those based on equal error protection (EEP) codes [1] from the view-point of the flexible condition.

II. Preliminaries

A. Out Line of Rate-Distortion Theory

Rate-distortion theory discusses on information compression by the trade-off property between rate and distortion [2]. The rate-distortion function can be written by:

$$R = R(D), \tag{1}$$

where R is the rate defined by $R = (1/n)\log M$, and M is the number of code words, n, the code length, and D, the distortion. The $R = R(D)$ is usually a convex downward and non-increasing function of D. The function $R = R(D)$ suggests us that we can decrease the rate drastically with tolerating a slightly growth of the distortion by proper source encoding.

B. System Evaluation Model

Generally, the rate R discussed in the previous subsection corresponds to the cost of a system, and distortion D, degradation of the performance of the system [7]. By extending

[1]We use "EEP codes" to show the conventional error correcting codes to contrast with UEP codes, although we have no such technical term.

the trade-off evaluation model, we have proposed the system evaluation model [5], where we have introduced a variable n as the system size.

Let the rate R be normalized by the maximum of R, $R_{\max}$, and the distortion D, by the maximum of D, $D_{\max}$, then we have $r = R/R_{\max}$, and $d = D/D_{\max}$, and the normalized function introducing n:

$$r = r(d, n). \tag{2}$$

For evaluation of the systems, we define the following properties to the (normalized) system evaluation function (2):

Definition 1:

1) *Flexible* [9]: The system A with $r = r_A(d, n)$ is flexible compared to the system B with $r = r_B(d, n)$, if $r_A(d, n) < r_B(d, n)$ for arbitrary d $(0 < d < 1)$, and n $(n > 1)$.
2) *Elastic* [9]: The system with $r = r(d, n)$ is elastic, if $r = r(d, n)$ is a decreasing function of n for arbitrary d $(0 < d < 1)$.
3) *Effective elastic* [5]: The system is effective elastic, if the system is elastic and $r = r(d, n)$ is a convex downward function of n.
4) *Trivial elastic* [9]: The system with $r = r(d, n)$ is trivial elastic, if $d = d(0, n)$ is a decreasing function of n, where $d = d(r, n)$ is the inverse function of $r = r(d, n)$.
5) *Marginal elastic* [5]: The system with $d = d(r, n)$ is marginal elastic, if $d = d(0, n)$ is a convex downward function of n.

□

4) Trivial elastic and 5) Marginal elastic are sometimes observed depending on the structure of systems (See [6]).

III. Disk Allocation of Cartesian Product Files

A. Cartesian Product Files

Let a set of attributes be denoted by $X_1, X_2, \cdots, X_n$, where an actual attribute value of X_i $(i = 1, 2, \cdots, n)$ is given by a domain $Z_i = \{0, 1, 2, \cdots, q-1\}$. Then the Cartesian product files are constructed by q^n buckets, and each bucket is specified by n-tuple $(z_1, z_2, \cdots, z_n)$, $z_i \in Z_i = \{0, 1, 2, \cdots, q-1\}$.

B. Partial Match Access Request

A partial match access request (PMAR) $\boldsymbol{Q}$ to q-ary Cartesian product file is given by:

$$\boldsymbol{Q} = (X_1 = z_1, X_2 = z_2, \cdots, X_n = z_n), \tag{3}$$

where $z_i \in \{0, 1, 2, \cdots, q-1, *\}$. The symbol $*$ shows an indefinite value, hence $* = \{0, 1, 2, \cdots, q-1\}$. This implies that the $*$ at the attribute X_i of $\boldsymbol{Q}$ requires to access the files with the all actual attribute values in Z_i. Thus all attributes are specified except for X_i, then we call such access request, PMAR.

Example 1: (PMAR) Letting $q = 2$, and $n = 4$, an example of the Cartesian product files is shown in Table I. If the PMAQ is given by:

$$\boldsymbol{Q} = (0, 0, *, 1), \tag{4}$$

then we must access the buckets of $(0, 0, 0, 1)$ and $(0, 0, 1, 1)$, since X_3 of $\boldsymbol{Q}$ requires don't care (whether married or not as shown in Table I). □

TABLE I
Example of Cartesian product file

X_1(Sex)	X_2(Income \$/year)	X_3(Married)	X_4(Age)
0 (Male)	0 (100k≤)	0(No)	0 (<20)
1 (Female)	1 (<100k)	1(Yes)	1 (20≤)

C. Disk Allocation Methods

Let us consider a disk allocation problem of the Cartesian product files. The problem is to partition the q^n buckets into G $(G \le q^n)$ disks so that we can access disks in parallel to simultaneously different buckets. This problem can be effectively solved by standard array appeared in coding theory.

Example 2: (Standard array) Letting $q = 2$, $n = 6$, $G = 8$, and $q^n = 64$, a construction method obtained by standard array is shown in Table II. We easily see that the following PMAR $\boldsymbol{Q}$:

$$\boldsymbol{Q} = (0, *, 1, *, 0, 0) \tag{5}$$

can be accessed to the all required buckets of $(0, 0, 1, 0, 0, 0)$, $(0, 0, 1, 1, 0, 0)$, $(0, 1, 1, 0, 0, 0)$, and $(0, 1, 1, 1, 0, 0)$ in parallel at once by Table II. □

Let a set of the buckets required by PMAQ $\boldsymbol{Q}$ be $S(\boldsymbol{Q})$. If $z_i = *$, and $z_j = *$, $(i \ne j)$, we hope to access simultaneously q^2 disks in parallel so that we can decrease the access time. Obviously, the maximum value of G, $G_{\max} = q^n$, and the minimum value of G, $G_{\min} = 1$. It is known that coding theory gives the following Lemma by using q-ary (n, k, d) code C, where n is the code length, k, the number of information symbols, and d, minimum distance.

Lemma 1: [1] Let the number of the $*$ occurred in $\boldsymbol{Q}$ be w $(0 \le w \le n)$. If $0 \le w < d$, then a disk allocation method based on a q-ary (n, k, d) code is the optimum. □

Lemma 1 states that the q-ary (n, k, d) code can give the method for accessing the q^w buckets in parallel at once, if $w < d$.

IV. Evaluation of Disk Allocation Methods

A. Formulation of Disk Allocation Methods

Let the q^n bucket be stored in q^{n-k} disks [2] by using an (n, k, d) code C. For given PMAR $\boldsymbol{Q}$, we let the number of access time to disks for $S(\boldsymbol{Q})$ be J. While we let a set of bucket accessible by using code C with $J = 1$ be $S(C)$. Then we give the evaluation loss (distortion measure) as the following definition.

Definition 2: The evaluation loss ρ is given by:

$$\rho = \begin{cases} 0 & (J = 1) \\ 1 & (J \ge 2). \end{cases} \tag{6}$$

□

[2]Note that the number of disks G equals to the number of coset of codes C.

TABLE II
STANDARD ARRAY FOR $q = 2, n = 6$, AND $G = 8$

disk#	bucket#							
0	000000	100110	010101	110011	001111	101001	011010	111100
1	100000	000110	110101	010011	101111	001001	111010	011100
2	010000	110110	000101	100011	011111	111001	001010	101100
3	001000	101110	011101	111011	000111	100001	010010	110100
4	000100	100010	010001	110111	001011	101101	011110	111000
5	000010	100100	010111	110001	001101	101011	011000	111110
6	000001	100111	010100	110010	001110	101000	011011	111101
7	000011	100101	010110	110000	001100	101010	011001	111111

$$Q = (0, *, 1, *, 0, 0) \longrightarrow \begin{matrix} 0\,0\,1\,0\,0\,0 \\ 0\,0\,1\,1\,0\,0 \\ 0\,1\,1\,0\,0\,0 \\ 0\,1\,1\,1\,0\,0 \end{matrix}$$

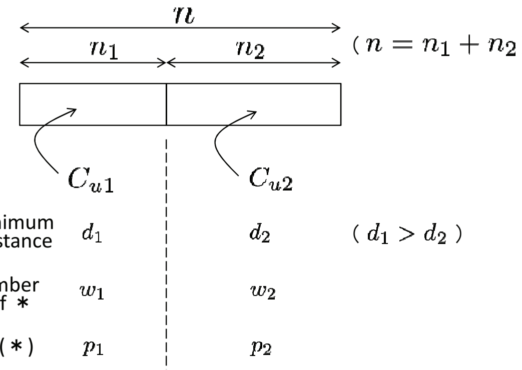

Fig. 1. 2-split UEP code $C_u[(n_1, n_2), M, (d_1, d_2)]$

The ρ is roughly defined to evaluate the access performance, it is, however, enough to evaluate the access time as described below. In (6), $J = 1$ implies $S(\boldsymbol{Q}) \subseteq S(C)$, and $J \geq 2$, $S(\boldsymbol{Q}) \supset S(C)$.

Using (6), the (average) access performance of the method constructed by the code C, ν is given by:

$$\nu = 0 \times \Pr(J = 1) + 1 \times \Pr(J \geq 2) = \Pr(J \geq 2). \tag{7}$$

Note that ν is already normalized. On the other hand, the cost G of the method constructed by the code C, i.e., the number of disks G is given by:

$$G = q^{n-k}. \tag{8}$$

Obviously, the maximum of G, $G_{\max} = q^n$, we have the normalized cost g as:

$$g = G/G_{\max} = q^{-k}. \tag{9}$$

B. Equal Probability Case

Let the probability of occurrence of the $*$ at the i-th attribute in PMAR $\boldsymbol{Q}$ be $\Pr(z_i = *)$, and let it be uniform distribution such that $\Pr(z_i = *) = \Pr(*) = p$ independent of i. Then from Lemma 1, we have

$$\Pr(J \geq 2) \leq \Pr(w \geq d). \tag{10}$$

Substitution of (10) into (7) gives ν, simply $\nu \leq \Pr(w \geq d)$.

C. Unequal Probability Case

Let us consider a case where $\Pr(z_i = *)$ is distinct of i. Then in such case, the following unequal error protection (UEP) codes [8], [12], [11] can play an important role to construct the disk allocation methods. We illustrate a UEP code C_u in Fig. 1, where $\Pr(z_i = *) = p_1$ for $i = 1, 2, \cdots, n_1$, $\Pr(z_j = *) = p_2$ for $j = n_1 + 1, n_1 + 2, \cdots, n_2$, and $n = n_1 + n_2$. A simplest case as shown in Fig. 1 is called the 2-split UEP code, where n_1 (n_2), d_1 (d_2), and p_1 (p_2) are the code length, the minimum distance, and the probability of occurrence of the $*$ in the 1st part (2nd part) of the UEP code C_u, respectively, and M is the number of the code words, i.e. the rate R is given by $R = (1/n) \log M$.

Lemma 2: [8] The $[(n_1, n_2), M, (d_1, d_2)]$ UEP code can access the disks with $J = 1$ as follows:

1) When $w_1 = 0$, then $J = 1$ if $w_2 < d_2$.
2) When $w_1 \geq 1$, then $J = 1$ if $w_1 + w_2 < d_1$.

□

From Lemma 2, we have the following theorem.

Theorem 1: Suppose a set of buckets $S(C_u)$ accessible to the disks with $J = 1$ using the code C_u. Then the probability of the access time with $J \geq 2$ satisfies:

$$\Pr(J \geq 2) \leq \Pr(w_1 = 0) \Pr(w_2 \geq d_2) + \sum_{s=1}^{n_1} \Pr(w_1 = s) \Pr(w_2 \geq d_1 - s), \tag{11}$$

where w_1 (w_2) is the number of the $*$ in the 1st part (2nd part) of the UEP code C_u.

Proof: Using the inverse of the sufficient conditions stated in 1), and 2) in Lemma 2, we can easily derive (11). □

Substitution of (11) into (7) gives ν.

D. Calculation for Evaluation

In the previous subsection, we derive the access performance ν, and the cost g. Note that ν and g can be given by the functions of parameters (n, k, d) for a code C, and $[(n_1, n_2), M, (d_1, d_2)]$ for a UEP code C_u. Then we simply denote the functions by $\nu = \nu(d, n)$, and $g = g(k, n)$ for the code C, and by $\nu = \nu(d_1, d_2, n_1, n_2)$, and $g = g(M, n)$ for the code C_u.

To compute the functions ν and g, we can use

- The LP upper bound [12] [3]: $M \leq f(d_1, d_2, n_1, n_2)$
- The Gilbert lower bound [10]: $d/n \leq H^{-1}(1 - R), n \to \infty$
- Constructive codes such as BCH codes and RS codes.

[3]The LP upper bound gives an upper bound on M (or R) solving the existence area of a code C and C_u by linear programming for given code parameters. Hence the computed results are given by a table.

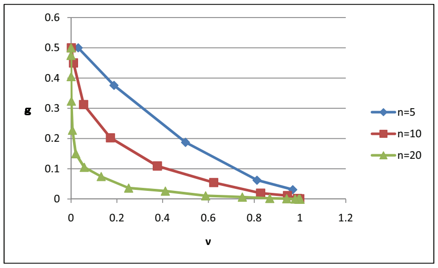

Fig. 2. LP upper bound by codes C ($p = 0.5$): BD-EEP

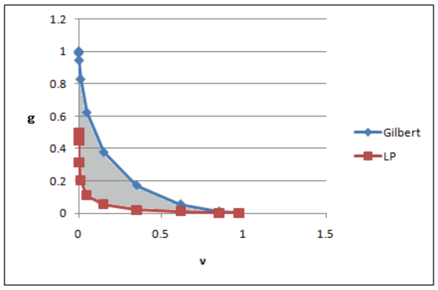

Fig. 3. LP upper bound and Gilbert lower bound by codes C ($p = 0.3$, and $n = 10$): BD-EEP

where $R = k/n$, or $R = (1/n) \log M$ holds. We have no bound up to now corresponding to the Gilbert lower bound for codes C_u. On the other hand, the LP upper bound can apply to both codes C and codes C_u. Note that the LP upper bound gives the lower bound on g, since $g = 1/M$, and similarly the Gilbert lower bound, the upper bound on g, since $g = q^{-k}$.

V. Numerical Results

A. Binomial Distribution (BD)

Assuming that the probability of occurrence of the $*$ in a disk allocation method constructed by the code C or the code C_u is given by the binomial distribution (BD) function, we have the following results (See Appendix A):

1) *Cases by (EEP) codes C:* Fig. 2 shows a case of $p = 0.5$, where we have used the LP upper bound. The difference between the LP upper bound and the Gilbert lower bound is depicted in Fig. 3 for a case of $p = 0.3$, and $n = 10$.

2) *Cases by (UEP) codes C_u:* Fig. 4 shows a case of $p_1 = 0.5$, and $p_2 = 0.25$, where we have used the LP upper bound. The difference between codes C and codes C_u is illustrated in Fig. 5 for a case of $p_1 = 0.5$, and $p_2 = 0.25$. Fig. 6 shows g as a function of n using the LP upper bound by codes C and by codes C_u.

B. Chernoff Bound (CB)

To compute the access performance ν, we use the Chernoff bounding (CB) techniques which are known in information theory [2] to evaluate coding systems. We have the following results (See Appendix B) without assuming any probability distribution function of occurrence of the $*$:

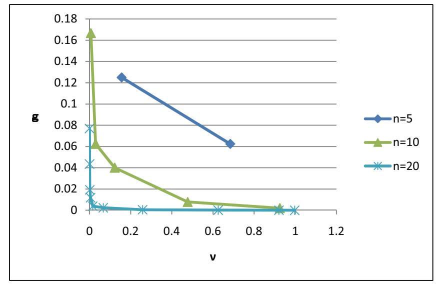

Fig. 4. LP upper bound by codes C_u ($p_1 = 0.5$, and $p_2 = 0.25$): BD-UEP

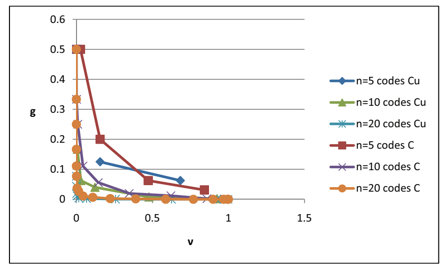

Fig. 5. LP upper bound by codes C and codes C_u ($p_1 = 0.5$, and $p_2 = 0.25$) : BD-EEP, BD-UEP

1) *Cases by (EEP) codes C:* Fig. 7 shows a case of $p = 0.3$, where we have used the LP upper bound. The difference between a binomial distribution and the Chernoff bound is illustrated in Fig. 8 for a case of $p = 0.3$, and $n = 10$.

2) *Cases by (UEP) codes C_u:* Fig. 9 shows the difference between a binomial distribution and the Chernoff bound for a case of $p_1 = 0.5$, and $p_2 = 0.25$, where we have used the LP upper bound.

C. Discussions

Since codes C and codes C_u exist discretely, the curves as shown in Figs.2-9 obtained by calculation for given parameters are not continuous and smooth. Hence we will roughly discuss evaluation results [4].

1) *Elastic and effective elastic:* We see that from Figs. 2, and 5, the disk allocation methods constructed by both codes C and codes C_u are *elastic*. As illustrated in Fig. 6, where we show the cost g as a function of system size n, we also see that the disk allocation methods constructed by both codes C and codes C_u are *effective elastic*.

2) *Flexible:* Fig. 5 also shows the disk allocation methods constructed by both codes C and codes C_u [5] for a case of $p_1 = 0.5$, and $p_2 = 0.25$, where the codes C are chosen to be the optimum for given parameters. This figure tells us that the disk allocation method constructed by codes C_u is *flexible*

[4]We have evaluated the methods based on codes C and codes C_u in various cases. Although results obtained in these cases are omitted here, they support to hold the properties such as elastic, effective elastic, or flexible.

[5]It sometimes happens that as seen in Fig. 5, codes C are superior compared to codes C_u ($n = 5$, for $\nu = 0.75$). The reason is that generally we have more effective (EEP) codes C than (UEP) codes C_u for given parameters.

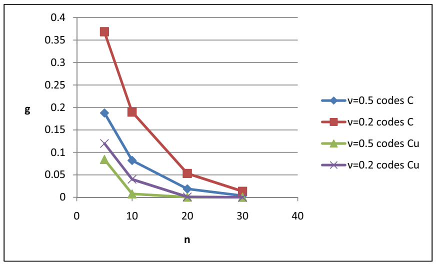

Fig. 6. LP upper bound by codes C ($p = 0.5$, $\nu = 0.2$, and $\nu = 0.5$) and codes C_u ($p_1 = 0.5$, $p_2 = 0.25$, $\nu = 0.2$, and $\nu = 0.5$): BD-EEP, BD-UEP

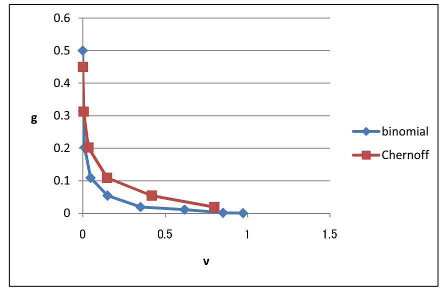

Fig. 8. Binomial distribution and Chernoff bound by codes C using LP upper bound ($p = 0.3$, and $n = 10$): BD-EEP, CB-EEP

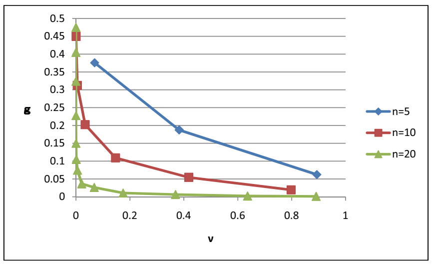

Fig. 7. LP upper bound by codes C ($p = 0.3$): CB-EEP

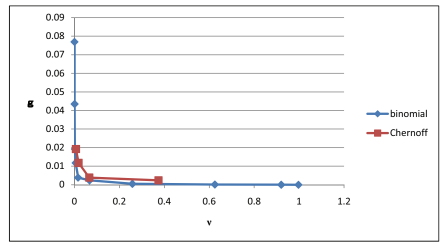

Fig. 9. Binomial distribution and Chernoff bound by codes C_u using LP upper bound ($p_1 = 0.5$, $p_2 = 0.25$, and $n = 20$): BD-UEP, CB-UEP

compared to that by codes C, since the curves of the former are lower than those of the latter. This result implies that it is better to construct a disk allocation method based on codes C_u than codes C for given parameters p_1 and p_2.

3) *Bounding techniques:* We use the bounding results from coding theory given by the LP upper bound and the Gilbert bound. These two bounds guarantee the existence of disk allocation methods as shown by shaded area in Fig. 3.

We also use Chernoff bounding techniques for calculating ν without assuming the probability distribution for occurrence of the $*$. On the other hand, assuming the binomial distribution for it, the difference between them is depicted in Figs. 8 and 9 for codes C. We see that the Chernoff bound is a good upper bound on g enough to show the properties such as elastic, effective elastic, or flexible, although n is relatively small, and p_1 and p_2 are relatively large [6].

VI. Concluding Remarks

In our another paper [11], we have precisely analyzed the performance of disk allocation methods from view-point of the average access time and the number of disks. In this paper, our approach is slightly different compared to [11], since we are interested in an application of system evaluation model. As the results obtained by the system evaluation model of the disk allocation methods constructed by error correcting codes, we can conclude that the $g = g(\nu, n)$ is:

(i) an *elastic* function
(ii) an *effective elastic* function and
(iii) if the occurrence of the $*$ is not uniform, disk allocation methods constructed by UEP codes C_u are *flexible* compared to those by codes C.

The (i) states that tolerating a small access performance degradation in ν introduces a drastic saving of the cost g, and this property can be effectively enhanced by the number of attribute n becomes large as seen in (ii). We can also remark that the UEP codes C_u are useful for the disk allocation methods if the probabilities of occurrences of the $*$ are not uniform as stated in (iii).

We note that the LP upper bound can give the case of $L(>2)$-split UEP codes, hence we can discuss the disk allocation methods in general. Furthermore, although we have only discuss in this paper on cases for $q = 2$, cases for $q \geq 3$ are straight forward, since we already have q-ary codes C and codes C_u, and their upper bounds, where $q(\geq 2)$ is a prime power.

To compute a function such as $g = F(\nu, n)$ using the Chernoff bounding techniques by analytically and to show the effective elastic directly will be rest as a further research.

Acknowledgment

One of the present authors, S. H. would like to thank Professor S. Oishi of Waseda University for giving him a chance to submit this paper. The authors also thank all the members of Matsusima Lab. and Goto Lab. of Waseda University for their fruitful comments.

[6]It is known that the Chernoff bound is always valid, and is usually tight upper bound when n becomes large and p, p_1, and p_2 are small [2].

APPENDIXES

A. Derivation by Binomial Distribution

By the (n, k, d) code C, we have from (10):

$$\nu = \Pr(J \geq 2) \leq \Pr(w \geq d) = \sum_{i=d}^{n} \binom{n}{i} p^i (1-p)^{n-i}, \tag{12}$$

and

$$g = q^{-k}. \tag{13}$$

While by the $[(n_1, n_2), M, (d_1, d_2)]$ code C_u, we have from Theorem 1, (11):

$$\begin{aligned}\nu &= \Pr(J \geq 2) \\ &\leq (1-p_1)^{n_1} \sum_{i=d_2}^{n_2} \binom{n_2}{i} p_2^i (1-p_2)^{n_2-i} \\ &\quad + \sum_{s=1}^{n_1} \Bigg\{ \binom{n_1}{s} p_1^s (1-p_1)^{n_1-s} \\ &\quad\quad \times \sum_{j=d_1-s}^{n_2} \binom{n_2}{j} p_2^j (1-p_2)^{n_2-j}, \end{aligned} \tag{14}$$

and

$$g = 1/M. \tag{15}$$

B. Derivation by Chernoff Bound [10]

By the (n, k, d) code C, we have from (10):

$$\nu = \Pr(J \geq 2) \leq \Pr(w \geq d) \leq \exp[-nE(\lambda, p)], \tag{16}$$

where

$$E(\lambda, p) = -\lambda \ln p - (1-\lambda) \ln(1-p) - H(\lambda), \tag{17}$$

$$H(x) = -x \ln x - (1-x) \ln(1-x), \tag{18}$$

and

$$\lambda = \frac{d}{n}, \quad \lambda > p. \tag{19}$$

While by the $[(n_1, n_2), M, (d_1, d_2)]$ code C_u, we have from Theorem 1, (11):

$$\begin{aligned}\nu &= \Pr(J \geq 2) \\ &\leq (1-p_1)^{n_1} \Pr(w_2 \geq d_2) \\ &\quad + \sum_{s=1}^{\alpha} 1 \cdot \Pr(w_2 \geq d_1 - s) \\ &\quad + \sum_{s=\alpha+1}^{n_1} \Pr(w_1 \geq s) \Pr(w_2 \geq d_2 - s) \end{aligned} \tag{20}$$

$$\begin{aligned} &\leq \exp[n_1 \ln(1-p_1) - n_2 E(\lambda_{a,1}, p_2)] \\ &\quad + \sum_{s=1}^{\alpha} \exp[-n_2 E(\lambda_{b,1}, p_2)] \\ &\quad + \sum_{s=\alpha+1}^{n_1} \exp[-n_1 E(\lambda_{b,2}, p_1) - n_2 E(\lambda_{b,3}, p_2)], \end{aligned} \tag{21}$$

where

$$\lambda_{a,1} = \frac{d_2}{n_2}, \quad 1 \geq \lambda_{a,1} > p_2, \tag{22}$$

$$\lambda_{b,1} = \frac{d_1 - s}{n_2}, \quad 1 \geq \lambda_{b,1} > p_2, \quad (s = 1, 2, \cdots, \alpha) \tag{23}$$

$$\lambda_{b,2} = \frac{s}{n_1}, \quad 1 \geq \lambda_{b,2} > p_1, \quad (s = \alpha + 1, \cdots, n_1) \tag{24}$$

$$\lambda_{b,3} = \frac{d_1 - s}{n_2}, \quad 1 \geq \lambda_{b,3} > p_2, \quad (s = \alpha + 1, \cdots, n_1) \tag{25}$$

and $\alpha = n_1 p_1$.

REFERENCES

[1] T. Fujiwara, U. Ito, T. Kasami, and J. Okui, "Performance analysis of disk allocation methods using error-correcting codes," *IEEE Trans. Inform. Theory*, vol. IT-37, no. 2, pp.379-384, Mar. 1991.

[2] R. G. Gallger, *Information Theory and Reliable Communication*, Jhon Wiley & Sons, New York, 1968.

[3] A. S. Hedayat, N. J. A. Sloane and J. Stufken, *Orthogonal Arrays: Theory and Applications*, Springer, New York, 1999.

[4] S. Hirasawa, and H. Inazumi, "A system evaluation model by using information theory," *30th Joint National Meeting*, ORSA/TIMS, MB35.3, Philadelphia, PE. USA, Oct. 1990.

[5] S. Hirasawa, and H. Inazumi, "A model for syatem evaluation based on information theory," *The 2000 International Conference of Management Science and Decision Making*, Tamkang University, Taipei, ROC, June 2000.

[6] H. Inazumi, M. Kochiya, and S. Hirasawa, "On the trade-offs between the file redundancy and the communication costs in distributed database systems," *IEEE Trans. SMC*, vol. SMC-19, no. 1, pp.108-112, Jan. 1989.

[7] H. Imazumi, Studies on the evaluations for information systems based on rate distortion theory, Dr. Eng. Dissertation, Waseda University, Nov. 1989.

[8] B. Masnick and J. Wolf, "On linear unequal error protection codes," *IEEE Trans. Inform. Theory*, vol. IT-3, pp. 600-607, Oct. 1967.

[9] J. Pearl, and A. Crolotte, "Storage space versus validity of answers in probabilistic question answering systems," *IEEE Trans. Inform. Theory*, vol. IT-26, no. 6, pp.633-640, Nov. 1979.

[10] W. W. Peterson, *Error-Correcting Codes*, MIT Press, Cambridge Mass., 1961.

[11] T. Saito, H. Inazumi, T. Matsushima, and S. Hirasawa, "Disk allocation methods for Cartesian product files by using error correcting codes"(in Japanese), *IEICE Technical Report*, vol. IT2010-55, pp.7-12, 2011.

[12] T. Saito, Y. Ukita, T. Matsushima, and S. Hirasawa, "A linear programming bound for unequal error protection codes," *Proceedings of the 2010 Australian Communications Theory Workshop*, pp.24-29, 2010.