

# 誤り訂正符号を用いた直積ファイルのディスク配置

## On Disk Allocation Methods for Cartesian Product Files by using Error Correcting Codes

平澤茂一<sup>†</sup> 斎藤友彦<sup>††</sup> 稲積宏誠<sup>†††</sup>

Shigeichi HIRASAWA<sup>†</sup> Tomohiko SAITO<sup>††</sup> Hiroshige INAZUMI<sup>†††</sup>

<sup>†</sup> サイバー大学 IT 総合学部, 早稲田大学 理工学術院 総合研究所  
青山学院大学 (<sup>††</sup> 理工学部, <sup>†††</sup> 社会情報学部)

<sup>†</sup> Information Technology and Business, Cyber University, Waseda Reseach Institute for Science and Engineering

<sup>††</sup> College of Science and Engineering / <sup>†††</sup> School of Social Informatics, Aoyama Gakuin University

要旨: 直積ファイルのディスク配置問題に誤り訂正符号の構造を導入し, 情報縮約論を用いたシステム評価モデルによりその性能を評価した.  $n$  個の属性を持つ  $q^n$  個の直積ファイルを仮定する. この  $q^n$  個の直積ファイルを  $G(\leq q^n)$  個のディスクに配置・記憶する. このとき部分照合質問(アクセス要求)に対し, 効率的なディスクの並列アクセスを可能とするファイルの配置問題を誤り訂正符号の構造により実現する. システム評価モデルを用いて,  $n$  が大きくなるときに望ましい性質を有することを示す. また, 部分照合質問の属性の出現に偏りがある場合, 不均一誤り訂正符号が適用できることを明らかにする.

**Abstract:** We have discussed disk allocation methods for Cartesian product files by introducing error correcting codes, and have clarified the performance of the methods by system evaluation models using rate distortion theory. Let us assume  $q^n$  Cartesian product files with  $n$  attributes and store  $q^n$  files into  $G(\leq q^n)$  disks. For a partial match query(access request), we present new disk allocation methods which are able to access in parallel to the disks as much as possible. Then we show that the proposed methods exhibit good property when  $n$  becomes large.

## 1 はじめに

1970年代後半, J. Pearl と A. Crolotte は質問回答システム(QAシステム)の記憶容量と誤りのトレードオフについて, 情報縮約論を用いて解析・評価した [10]. すなわち, 各種のQAシステムに対し, 小さな誤りを許容すれば大きな記憶容量を削減できる条件を明らかにした. このような条件を「フレキシブル(Flexible)」や「エラスティック(Elastic)」と呼んでいる. 筆者らはこれを拡張したトレードオフ評価モデルを提案し [8, 3, 6], 各種の情報システムに適用した. その結果, 情報システムを設計する際, システムの規模が大きくなるに従い相対的に効率がよい性質「効率的エラスティック(Effectively Elastic)」を持つか否かを判定することが出来る.

一方, 誤り訂正符号の構造は実験計画法 [2] やファイルのディスク配置問題に適用されることが知られており, 古くから研究されている [1].  $q^n$  個の直積ファイルを  $G(\leq q^n)$  個のディスクに分散配置する際,  $n$  個の直積ファイルの属性の内, 属性値が不定(\*で示す)の場合, 複数個のファイルを同時にアクセスする必要がある. そのとき, ディスクの並列アクセスを可能とする効率的なファイルの配置問題としてモデル化される.

本稿では, 誤り訂正符号の構造を用いた直積ファイルの分散配置にトレードオフ評価モデルを適用し, 直積ファイルの個数が増加するに従い望ましい条件を持つか否かを議論する. さらに, 属性値が不定となる属性の位置に偏りがある場合, 情報の位置により不均一な誤り訂正能力を持つ符号が効率よく適用できることを示す.

## 2 トレードオフ評価モデル

### 2.1 レート歪関数

情報縮約論は, 有歪の圧縮を対象とし, 情報伝送速度(rate)  $R$  と歪(distortion)  $D$  の関係を論じている. 両者の関係を表すレート歪関数  $R = R(D)$  は一般に下に凸な非増加関数で, わずかな歪を許容すれば大きなレートの減少が望める [4].

圧縮することにより, 人間の目や耳では認識できない程度の歪を許せば帯域を大きく低減できる. 現在の携帯電話や地上デジタル放送などは, この性質を巧に利用している.

### 2.2 システム評価モデル

J. Pearl と A. Crolotte はレートと歪の間のトレードオフ関係に注目し, QAシステムなどの情報システムの評価に適用するモデルを提案した [10]. 筆者らはさらにこれを拡張し [3, 6], 大規模システムの設計に際し構成をどのようにすればよいか, あるいは  $n$  が増加するとともに, 相対的に効率が良くなる性質を持つかどうかを判定できる基準を提案した. ここで,  $n$  は情報システムの規模を示すパラメータで, レートはシステムのコストに, 歪は性能劣化の程度に対応する.

まず,  $R = R(D)$  を  $D$  軸,  $R$  軸ともそれぞれ  $D_{max}$ ,  $R_{max}$  で正規化する. すなわち,  $r = R/R_{max}$ ,  $d = D/D_{max}$  とし, 関数

$$r = r(d, n), \quad (1)$$

について考える. システムを評価するために, 次の性質を定義する. システム評価関数  $r = r(d, n)$  について

- (1) 2つのシステム  $A, B$  を考え, それぞれの関数を  $r = r_A(d, n)$ ,  $r = r_B(d, n)$  とする. ある  $n(> 1)$  に対し, 任意の  $d(0 < d < 1)$  において,  $r_A(d, n) > r_B(d, n)$  のとき, システム  $B$  はシステム  $A$  より「フレキシブル」(Flexible) であるという [10].
- (2) 任意の  $d(0 < d < 1)$  において, 関数  $r = r(d, n)$  の  $r$  が  $n$  の減少関数のとき, 「エラスティック」(Elastic) であるという [10].
- (3) (2)において, さらに  $r$  が  $n$  の下に凸な関数のとき, 「効果的エラスティック」(Effectively Elastic) であるという [6].
- (4) 関数  $d = d(0, n)$  の  $d$  が  $n$  の減少関数のとき, 「トリivialエラスティック」(Trivial Elastic) という [10]. ただし,  $d = d(r, n)$  は  $r = r(d, n)$  の逆関数である.
- (5) 任意の  $r(0 < r < 1)$  において, 関数  $d = d(r, n)$  の  $d$  が  $n$  の下に凸な減少関数のとき, 「マージナルエラスティック」(Marginal Elastic) であるという [6].

### 3 直積ファイルのディスク配置問題

#### 3.1 直積ファイル

$X_1, X_2, \dots, X_n$  を属性の集合とする．各属性  $X_i (i = 1, 2, \dots, n)$  が領域  $Z_i = \{0, 1, \dots, q-1\}$  で与えられる ( $q \geq 2$ ) 個の属性値を持つ直積ファイル (Cartesian product files) を考える．このとき,  $q$  元直積ファイルは  $q^n$  個のファイルで構成される．ここで,  $q^n$  個のそれぞれのファイルをバケット (bucket) という．

#### 3.2 部分照合アクセス要求

$q$  元直積ファイルに対するアクセス要求  $Q$  を次式で表す．

$$Q = (X_1 = z_1, X_2 = z_2, \dots, X_n = z_n), \quad (2)$$

ここで

$$z_i \in \{0, 1, \dots, q-1, *\}, \quad (3)$$

である．すなわち,  $z_i$  は属性  $X_i$  の領域  $Z_i$  における実現値  $\{0, 1, \dots, q-1\}$  のいずれか, または \* である．ここで, 記号 \* は不定 (don't care) で,  $* = \{0, 1, \dots, q-1\}$  を表す． $\|Z_i\| = q$  であるから,  $z_i$  は  $q+1$  の値をとる．このように, \* を除く他の属性値は特定されるから, 一部分だけ属性値を指定することになり, 部分照合アクセス要求といわれる．

##### 例 1 部分照合アクセス要求

$q = 2, n = 4$  の場合の例を表 1 に示す．もし, アクセス要求が

$$Q = (0, 0, *, 1) \text{ の場合, } Q = (0, 0, 0, 1) \text{ and } (0, 0, 1, 1)$$

を示す．ここで,  $z_3 = *$  は既婚者か未婚者を問わないことを示している．このように, \* は該当する属性のすべての属性値をもつバケットを照合しなければならない．

表 1: 直積ファイルの例

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

#### 3.3 ファイルの分散配置問題

$q^n$  個の直積ファイルを  $G$  個のディスクに配置する問題を考える．3.2 で述べた部分照合アクセス要求に対し, 複数個のファイルを同時に (並列に) アクセスしたい．そのために符号理論で知られた標準配列 (standard array) を用いる．

例 2  $q = 2, n = 6, G = 8, q^n = 64$  の場合, 構成法の例を表 2 に示す．

$$Q = (0, *, 1, *, 0, 0,) \text{ の場合, } Q = (001000) \text{ and } (001100) \text{ and } (011000) \text{ and } (011110)$$

が 1 回で並列にアクセスされる．

$Q$  によって指定されたバケットの集合を  $S(Q)$  とする．もし,  $z_i = *$  で,  $z_j = *, (i \neq j)$  のときは, アクセス時間を低減するために  $q^2$  個のバケットを並列にアクセスしたい． $Q$  に生じた \* の個数を  $w (= \# \text{ of } *; 0 \leq w \leq n)$  とする．明らかに,  $G = q^n$  は最大で,  $G = 1$  は最小である．このとき, 符号理論 [5] で知られた構成法により, 符号長  $n$ , 情報記号数  $k$ , 最小距離  $d$  の  $(n, k, d)$  符号を用いて, 次の結果が知られている．

補題 1 [1] もし,  $0 \leq w < d$  ならば,  $q$  元  $(n, k, d)$  符号に基づくディスク配置法は最適である．すなわち, 1 回で  $q^w$  個のバケットを並列アクセス可能である．

表 2:  $q = 2, n = 6, G = 8, q^n = 6$  の場合

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

$$Q = (0, *, 1, *, 0, 0) \rightarrow \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 \end{bmatrix}$$

### 4 ファイル配置問題の評価

#### 4.1 ファイル配置問題の定式化

$q$  元  $(n, k, d)$  符号  $C$  を用いて,  $q^n$  個のバケットを  $G = q^{n-k}$  個のディスクに収納する．

まず, 与えられた部分照合アクセス要求  $Q$  に対し,  $Q$  で指定されたバケットの集合  $S(Q)$  を回数  $J$  でアクセス可能とする．一方, 符号  $C$  を用いて  $J = 1$  でアクセス可能なバケットの集合を  $S(C)$  とする．このとき評価損失 (歪測度)  $\rho$  を

$$\rho = \begin{cases} 0, & J = 1 \quad (S(Q) \subseteq S(C)), \\ 1, & J \geq 2 \quad (S(Q) \supset S(C)), \end{cases} \quad (4)$$

とする．これを用いて, 符号  $C$  によるシステムのアクセス性能<sup>1</sup>  $\nu$  は次式で与えられる<sup>2</sup>．

$$\nu = 0 \times \Pr(J = 1) + 1 \times \Pr(J \geq 2). \quad (5)$$

一方, 符号  $C$  によるシステムのコスト<sup>3</sup>  $G$  はディスクの個数に対応する．

$$G = q^{n-k}. \quad (6)$$

明らかに,  $G_{max} = q^n$  であるから, 正規化したコストは次式で与えられる．

$$g = G/G_{max} = q^{-k}. \quad (7)$$

#### 4.2 均一の場合

\* の生起確率  $\Pr(*) = p$  とし, 属性によらず均一とする．このとき補題 1 より

$$\Pr(J \geq 2) = \Pr(w \geq d), \quad (8)$$

で与えられる．式 (8) を式 (5) に代入すれば, アクセス性能  $\nu$  が求められる．

#### 4.3 不均一の場合

\* の生起が属性の位置により異なる場合を考える．図 1 に示した不均一誤り訂正 (Unequal Error Protection: UEP) 符号  $C_u$  を用いて以下のように定式化される．

いま, 長さ  $n = n_1 + n_2$ ,  $C_{u1}(C_{u2})$  の部分の最小距離を  $d_1(d_2)$  とする符号語数  $M$  の不均一誤り訂正符号  $C_u$  のプロファイル  $[(n_1, n_2), M, (d_1, d_2)]$  で表す． $w_1(w_2), p_1(p_2)$  をそれぞれ  $C_{u1}(C_{u2})$  に生じた \* の個数と \* の生起確率とする．このとき, 次の補題が成り立つ．

<sup>1</sup>歪  $D$  に相当する．

<sup>2</sup>確率による加重平均であるから, 既に正規化されている．

<sup>3</sup>レートに相当する．

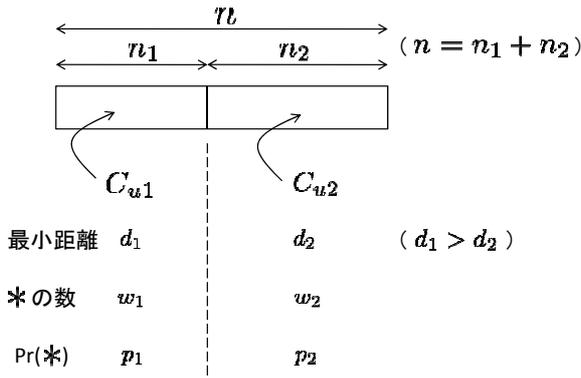


図 1: UEP 符号  $C_u : [(n_1, n_2), M, (d_1, d_2)]$  の構造

補題 2 [9]

- (1)  $w_1 = 0$  のとき,  $w_2 < d_2$  ならば  $J = 1$  を達成する .
- (2)  $w_1 \geq 1$  のとき,  $w_1 + w_2 < d_1$  ならば  $J = 1$  を達成する .

その結果, 次の定理が得られる .

定理 1 符号  $C_u$  を用いて,  $J = 1$  でアクセス可能なバケット集合  $S(C_u)$  を考える . 2 回以上のアクセスを必要とする確率  $\Pr(J \geq 2)$  は次式で与えられる .

$$\Pr(J \geq 2) = \Pr(w_1 = 0)\Pr(w_2 \geq d_2) + \sum_{s=1}^{n_1} \Pr(w_1 = s)\Pr(w_2 = d_1 - s). \quad (9)$$

4.2 と同様, 式 (9) を式 (5) に代入すれば  $\nu$  が求められる .

#### 4.4 評価関数の計算法

4.2, 4.3 で, コスト  $g$  とアクセス性能  $\nu$  を求めた . 両者を結び付けるために, 符号のレート  $R$  (または  $M$ ) と最小距離  $d$  (または  $d_1, d_2$ ) の関係を用いなければならない . そのために

- (1) LP 上界式 [11]:  $M \geq f(n_1, n_2, d_1, d_2)$
- (2) BCH 符号や RS 符号などの具体的符号: [例]  $d = n - k + 1$  (RS 符号)
- (3) Gilbert 下界式:  $d/n \leq H^{-1}(1 - R)(n \rightarrow \infty)$ ,

を適用する . ただし,  $R = k/n$ ,  $R = (1/n)\log M$  である . この内, ここでは (1)(3) を用いた . なお, (1) LP 上界式は与えられた符号パラメータに対し, 符号語数  $M$  を LP を解いて求めるものでタイトな限界式であり, 均一誤り訂正符号  $C$  と不均一誤り訂正符号  $C_u$  のいずれのケースも計算可能である<sup>4</sup> .

#### 4.5 数値計算結果

\* の生起が二項分布に従うと仮定する . このとき

<sup>4</sup> $g = q^{-k} = 1/M$  であるから, LP 上界式による  $M$  の上界は  $g$  の下界を与える . 同様に Gilbert 下界式による  $d$  の下界は  $g$  の上界を与える . なお, (2)(3) は均一誤り訂正符号しか求まっていない .

#### (1) 均一誤り訂正符号 $C$ による場合

図 2 に  $p = 0.3$ , 図 3 に  $p = 0.5$  で LP 上界式を用いた場合, 図 4 に  $p = 0.3$ , Gilbert 下界式を用いた場合を示す .

図 5 に LP 上界式と Gilbert 下界式を比較した結果を示す .  $p = 0.3$  で,  $n = 10$  の場合である .

#### (2) 不均一誤り訂正符号 $C_u$ による場合

図 6 に  $p_1 = 0.5, p_2 = 0.25$  で, LP 上界式を用いた場合を示す . さらに, 図 7 に同一の条件  $p_1 = 0.5, p_2 = 0.25$  で, LP 上界式を用いて均一誤り訂正符号  $C$  と不均一誤り訂正符号  $C_u$  の性能を比較した結果を示す .

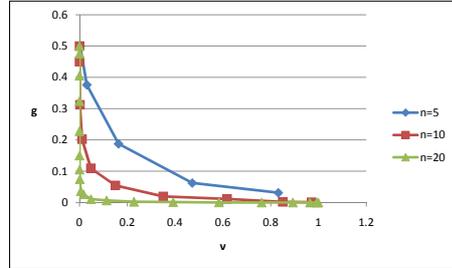


図 2: LP 上界式  $p = 0.3$

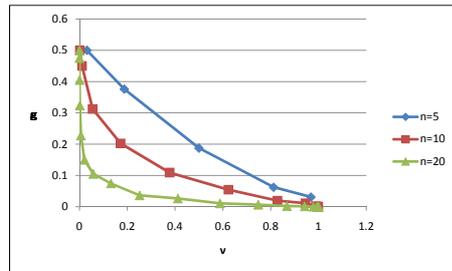


図 3: LP 上界式  $p = 0.5$

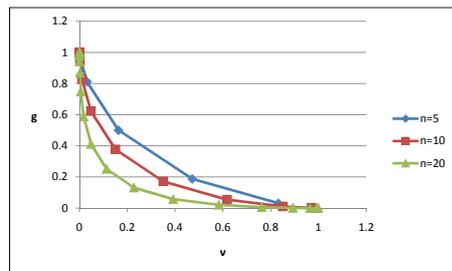


図 4: Gilbert 下界式  $p = 0.3$

## 5 考察

符号構成が離散的にしか存在しないため, 与えたパラメータの数値計算結果は滑らかな曲線とはならない . そのため, 残念ながらグラフ上で十分な考察が出来ない微妙な部分がある . したがって, おおよその性質を述べるにとどまる .

### 5.1 均一誤り訂正符号 $C$ による場合

- (1.1) 図 2, 図 3 より, LP 上界式から求めた曲線からエラスティックであることが分かる .
- (1.2) 同様に図 4 より, Gilbert 下界式から求めた曲線からもエラスティックであることが分かる .
- (1.3)  $p = 0.3$  の場合,  $n = 10$  について両者を併記した図 5 より, 符号  $C$  の存在する範囲は 2 つの曲線の間であることが分かる .

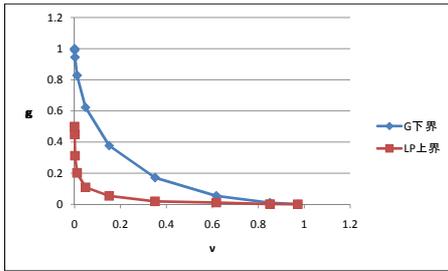


図 5: LP 上界式と Gilbert 下界式  $p = 0.3, n = 10$

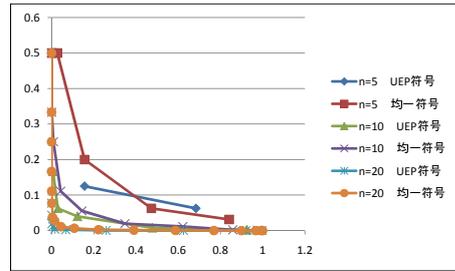


図 7: LP 上界式 (UEP, 均一符号)  $p_1 = 0.5, p_2 = 0.25$

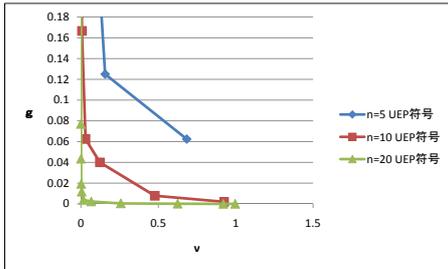


図 6: LP 上界式 (UEP 符号)  $p_1 = 0.5, p_2 = 0.25$

## 5.2 不均一誤り訂正符号 $C_u$ による場合

- (2.1) 図 6 より, LP 上界式から求めた曲線からエラスティックであることが分かる.
- (2.2) 図 7 より, LP 上界式から求めた符号  $C$  (均一) による場合と符号  $C_u$  による場合 (不均一) の曲線を比較すると, 後者がよりフレキシブルであることが分かる<sup>5</sup>. このことは, 予め \* の生起が属性の位置により不均一であることが分かっている場合, 均一誤り訂正符号より不均一誤り訂正符号を用いた方が効率が良いことを示している<sup>6</sup>.

## 6 むすび

均一・不均一誤り訂正符号を用いて, 直積ファイルのディスク配置を行った場合, 数値計算の結果, トレードオフにあるシステム評価関数は

- (1) 下に凸の関数である.
- (2) システムの規模を示すパラメータ  $n$  が増加するにしたがい, 効率が良くなる. すなわち, エラスティックである.
- (3) \* の生起が属性により不均一なときは, 不均一誤り訂正符号を用いることが有効である. すなわち, フレキシブルである.

であることを明らかにした.  
今後の課題として,  $n$  が大きくなる場合の振舞いについて解析する必要がある. また, 二項分布を仮定しない確率計算で離散的な符号のパラメータから滑らかな陽関数表現  $g = F(v, n)$  を導出し, 直接効率のエラスティック性を持つかについて検討したい.

<sup>5</sup> $n = 5$  の [(1, 4), 16, (2, 1)] 不均一誤り訂正符号  $C_u$  のケースはパラメータの選択が難しいため, グラフ上で逆転している. また, 両者で同一のコスト  $g$  を選ぶことが出来ない.

<sup>6</sup>通常, 不均一誤り訂正符号より均一誤り訂正符号の方が構成方法に富んでいる. そのため, 後者の方が効率のよい符号が得られるとされている. 上記逆転したケースはそのような場合で,  $d_1 = 2, d_2 = 1$  の不均一誤り訂正符号と  $d = 2$  の均一誤り訂正符号のレートが同一となり, 不均一誤り訂正符号を構成したとき,  $d_2 = 1$  によるレートの増加が望めなかったことによる.

## 謝辞

熱心な議論を頂いた早稲田大学・松嶋・後藤研 D ゼミの皆様へ感謝します.

## 参考文献

- [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] A. S. Hedayat, N. J. A. Sloane and J. Stufken, Orthogonal Arrays: Theory and Applications, Springer, New York, 1999.
- [3] 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. 1900.
- [4] 平澤茂一, 情報理論, 培風館, 東京, 1996.
- [5] 平澤茂一, 西島利尚, 符号理論入門, 培風館, 東京, 1999.
- [6] 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.
- [7] 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.
- [8] H. Imazumi, Studies on the Evaluations for Information Systems based on Rate Distortion Theory, 早稲田大学博士論文, 1989年11月.
- [9] B. Masnick and J. Wolf, "On linear unequal error protection codes," IEEE Trans. Inform. Theory, vol. IT-3, pp. 600-607, Oct. 1967.
- [10] 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.
- [11] 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.