2011 IEEE International Conference on Systems, Man and Cybernetics
Anchorage, AK, U.S.A., 09 - 12 October, 2011

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

Shigeichi Hirasawa + (Cyber University, Japan)
Tomohiko Saito (Aoyama Gakuin University, Japan)
Hiroshige Inazumi (Aoyama Gakuin University , Japan), and
Toshiyasu Matsushima (Waseda University , Japan)

+:e - mail:
shigeichi_hirasawa@cyber-u.ac.jp

サイバー大学
Cyber University

Out line of this paper

I. Introduction
II. Preliminaries
III. Disk Allocation of Cartesian Product Files
IV. Evaluation of Disk Allocation Methods
V. Numerical Results
VI. Concluding Remarks

1. QA (Question Answering) Systems by J. Perl and A. Crolotte in 1970's [9]
   - Rate-Distortion Theory
   - Flexible and Elastic
2. System Evaluation Model [4][5][6][7]
3. Error Correcting Codes [11][12]

サイバー大学 Cyber University

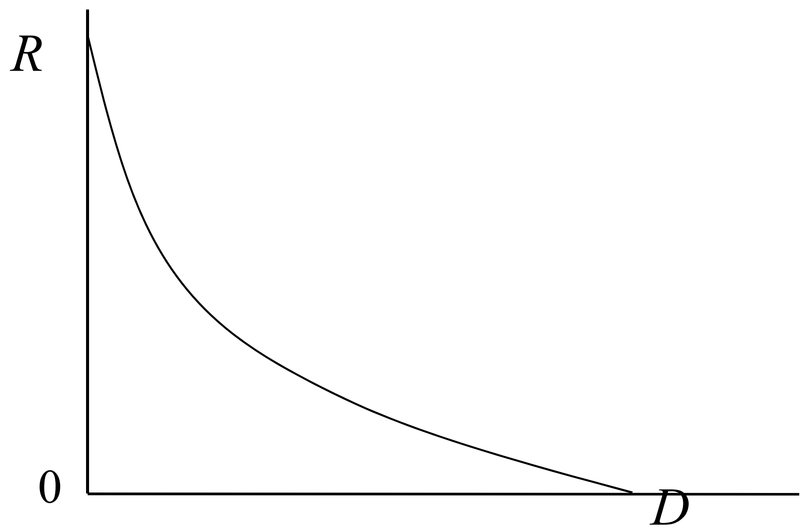
A. Out Line of Rate-Distortion Theory

The rate-distortion function: $R = R(D)$ **(1)**

R : rate

D: 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.



II. Preliminaries

B. System Evaluation Model

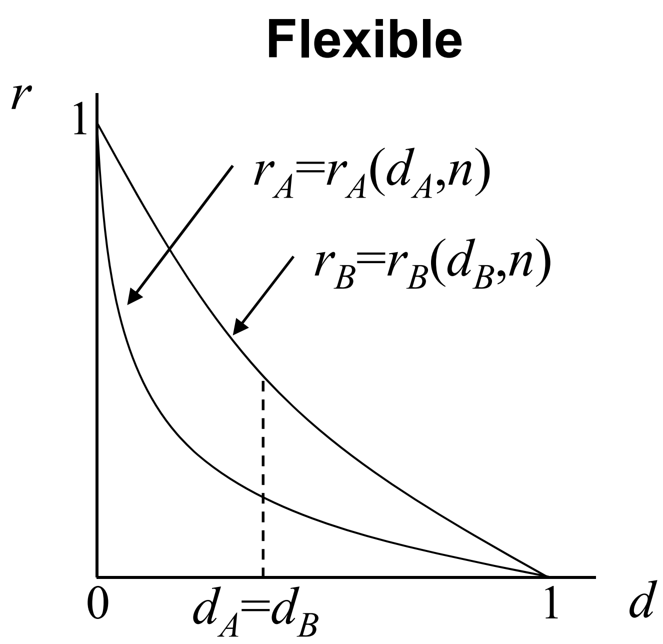
The system evaluation model: $r = r(d, n)$ **(2)**

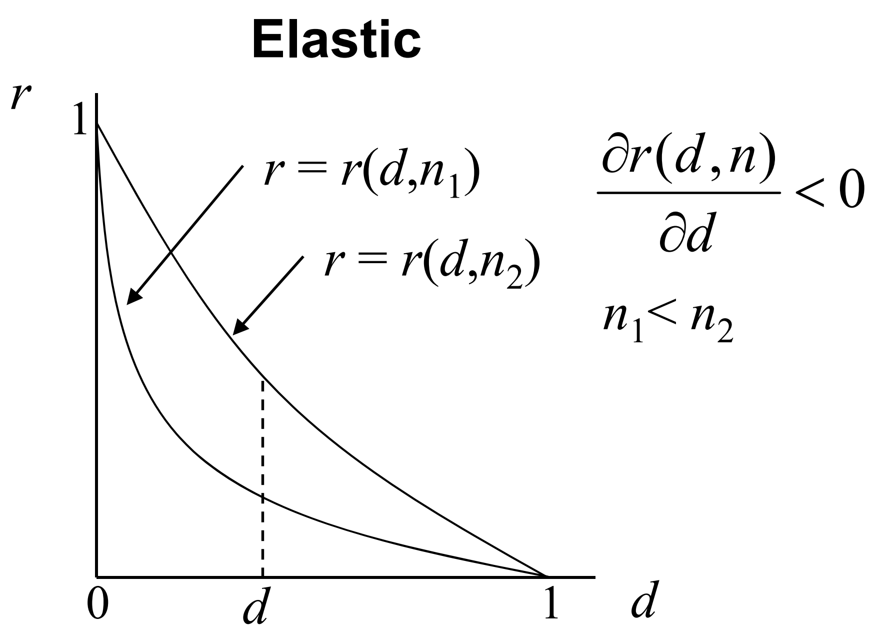
r : the cost of the system ($= R/R_{max}$)

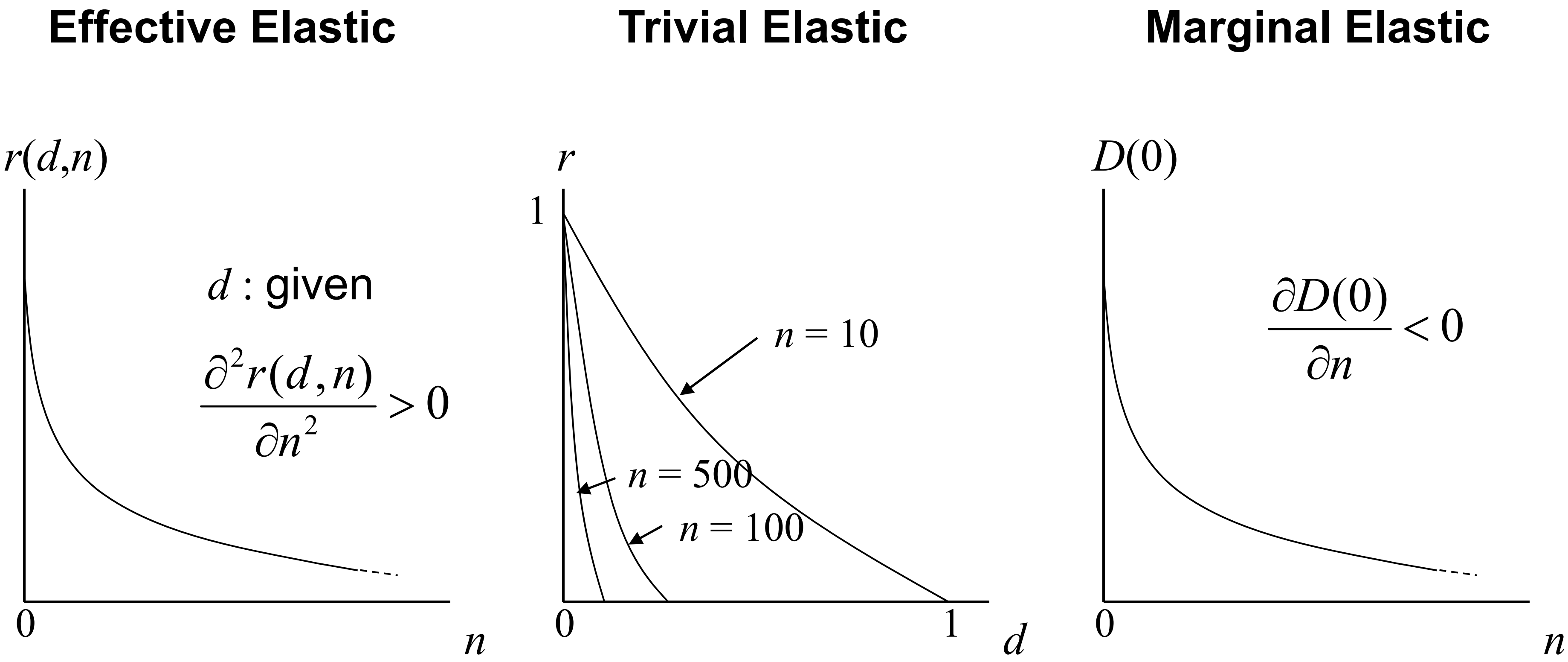
d: degradation of the performance of the system ($d=D/D_{max}$)

n: the system size

[Definition 1]





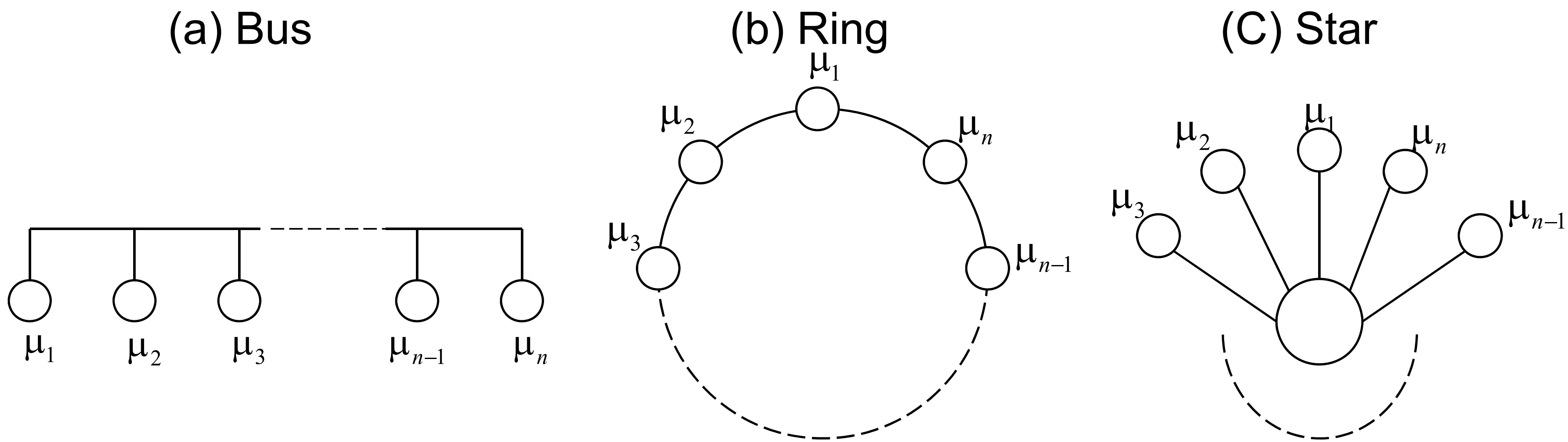

Effective Elastic
Trivial Elastic
Marginal Elastic
r(d,n)
d : given
∂²r(d,n)/∂n² > 0
0
n
r
1
n = 10
n = 500
n = 100
0
1
d
D(0)
∂D(0)/∂n < 0
0
n

II. Preliminaries

[Example] Distributed Database in Computer networks [6][7]

- r : the redundancy of the file duplication
- d : the access cost to the files
- n : the number of the nodes

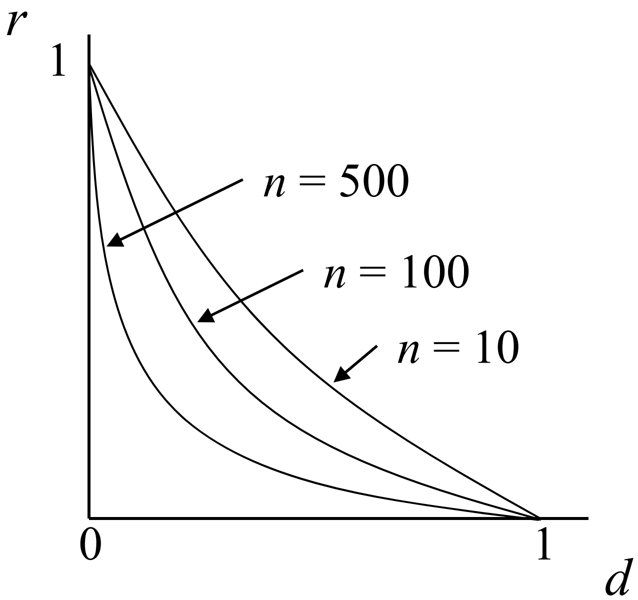
Network Topology



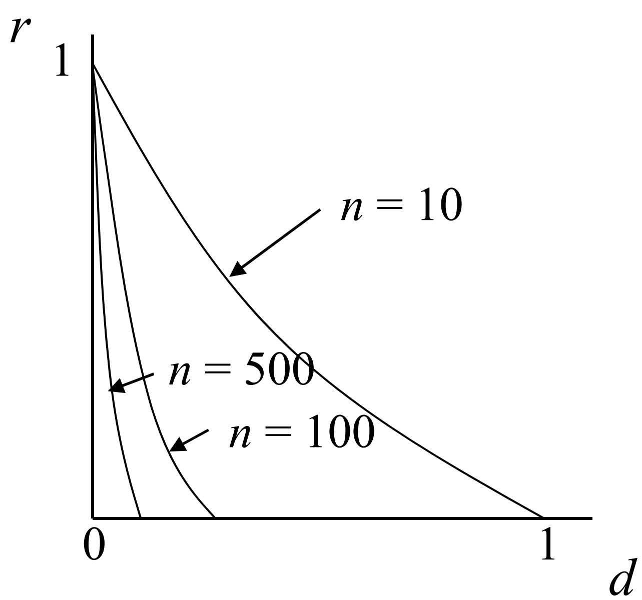
μ_i : node i

II. Preliminaries

(a), (b) Elastic



(c) Trivial Elastic



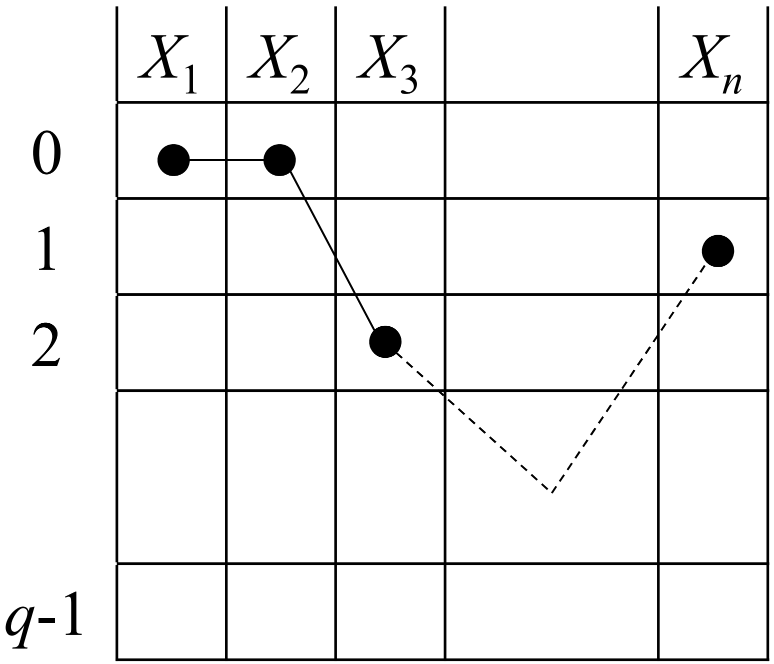
A. Cartesian Product Files

q-ary product files

Attributes: $X_1, X_2, \ldots, X_n$

Domain: $Z_1, Z_2, \ldots, Z_n,$

Actual attribute value: $z_i \in Z_i = \{0, 1, \ldots, q\text{-}1\}$



q^n bucket

Path shows a bucket(0, 0, 2,…, 1)

B. Partial Match Request (PMR)

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

where $z_i \in \{0, 1, \ldots, q\text{-}1, *\}$

$*$: don't care ($* = \{0, 1, \ldots, q\text{-}1\}$)

[Example 1] (PMR)

$q=2$, $n=4$, $G=4$ (G: the number of the files)

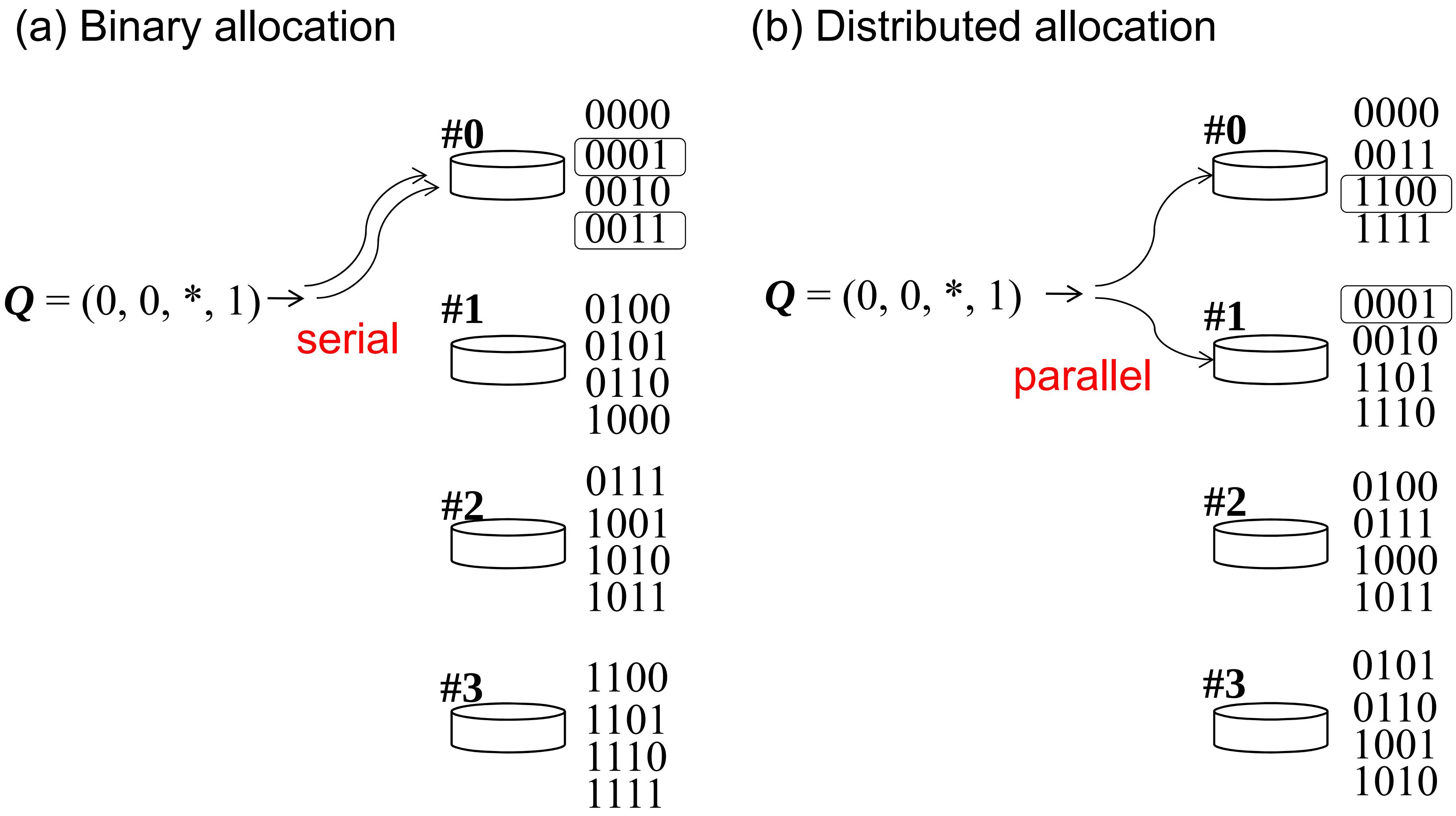
Table I

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 ≤)

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

$*$: married or unmarried

III. Disk Allocation of Cartesian Product Files



G: the number of the files, ($G_{max} = q^n$)

[Example 2] (Standard Array) Distributed Allocation Method of Product Files by Error Correcting Codes

q=2, n=6, G=8, and q^n=64

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

サイバー大学 Cyber University

Standard array of the binary (6, 3, 3) code

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

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

→$\boldsymbol{Q}$ = (001000), (001100), (011000), (011100)

Coding Theory gives:

[Lemma 1] [1] Let the number of the * occurred in $\boldsymbol{Q}$ be w ($0 \leq w \leq n$). If $0 \leq w < d$, 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 $w < d$.

J : the number of access times to disks for $S(Q)$

w : the number of * occurred in Q

$0 \leq w < d$ ➔ accessible to q^w buckets by J=1 (in parallel)

サイバー大学 Cyber University

A. Formulation of Disk Allocation Methods

J : the number of access times to disks for $S(\boldsymbol{Q})$

ρ : evaluation loss

$S(\boldsymbol{Q})$: the set of buckets required for $\boldsymbol{Q}$

$S(C)$: the set of accessible by using code C

[Definition 2] The evaluation loss ρ is given by:

$$\rho = \begin{cases} 0, & J = 1\ (S(\mathbf{Q}) \subseteq S(C)), \\ 1, & J \geq 2\ (S(\mathbf{Q}) \supset S(C)), \end{cases} \qquad (6)$$

□

$J = 1$ ➔ $S(Q) \subseteq S(C)$

$J \geq 2$ ➔ $S(Q) \supset S(C)$

サイバー大学 Cyber University

A. Formulation of Disk Allocation Methods

ν : access performance

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

$$= \Pr(J \geq 2) \quad \textbf{(7)}$$

g : cost

G: the number of disks $G = q^{n-k}$ **(8)** , $G_{\max} = q^{n}$

$$g = G / G_{\max} = q^{-k} \quad \textbf{(9)}$$

B. Equal Probability Case (EEP Codes C)

p : the probability of the occurrence of * , p=Pr(*)

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

i. e,

$$\nu \leq \Pr(w \geq d)$$

IV. Evaluation of Disk Allocation Method

C. Unequal Probability Case (UEP Codes C_u)

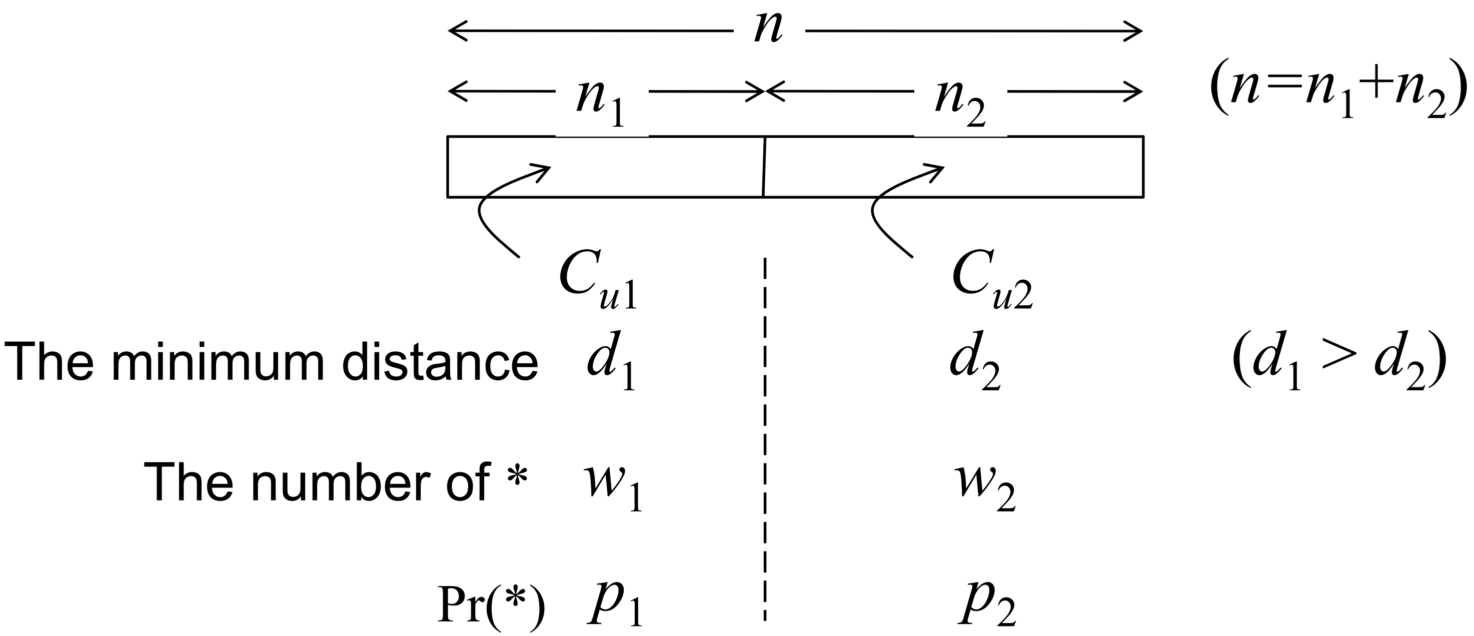


Fig. 1: 2-split UEP code $C_u[(n_1,n_2), M, (d_1,d_2)]$

$\Pr(z_i = *) = p_1$ for $i = 1,2, \ldots, n_1$

$\Pr(z_j = *) = p_2$ for $j = n_1+1, n_1+2, \ldots, n_2$

$n = n_1 + n_2$

IV. Evaluation of Disk Allocation Method

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

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

□

1. $w_1=0$, $w_2<d_2$ ➔ J=1
2. $w_1\geq 1$, $w_1+w_2<d_1$ ➔ J=1

IV. Evaluation of Disk Allocation Method

サイバー大学 Cyber University

[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:

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

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

D. Calculation for Evaluation

Access performance: $\nu = \nu(n, d)$

$\delta = d / n$ vs. $R = k / n$

Cost: $g = g(k, n)$

1. LP upper bound [11]: $M \leq f(n_1, n_2, d_1, d_2)$
2. Gilbert lower bounds: $d/n \geq H^{-1}(1-R)$ $(n \rightarrow \infty)$
3. Actual parameters of BCH codes, RS codes

where $R=k/n$, or $R=(1/n) \log M$, M is the number of code words

サイバー大学 Cyber University

A. Binomial Distribution (BD)

Pr(*): Binomial Distribution

(1) Cases by EEP codes C

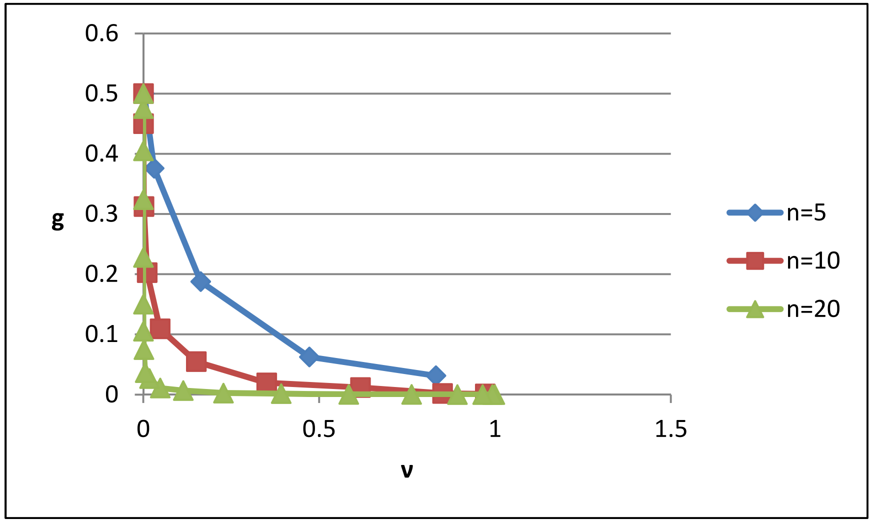


Fig. 2: LP upper bound by Codes C (p=0.5)

(Elastic)

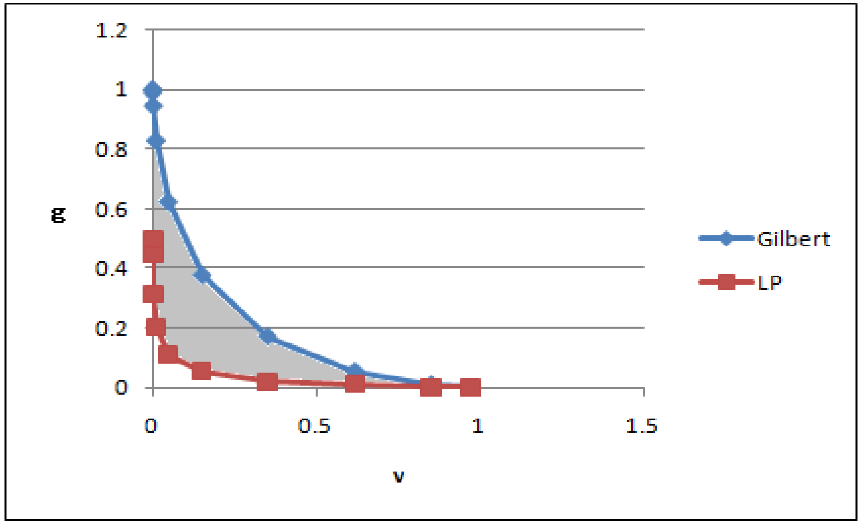


Fig. 3 LP upper bound and Gilbert lower bound by codes C (p=0.3 and n=10)

(Existence of the codes)

V. Numerical Results

A. Binomial Distribution (BD)

(2) Case by UEP codes C_u

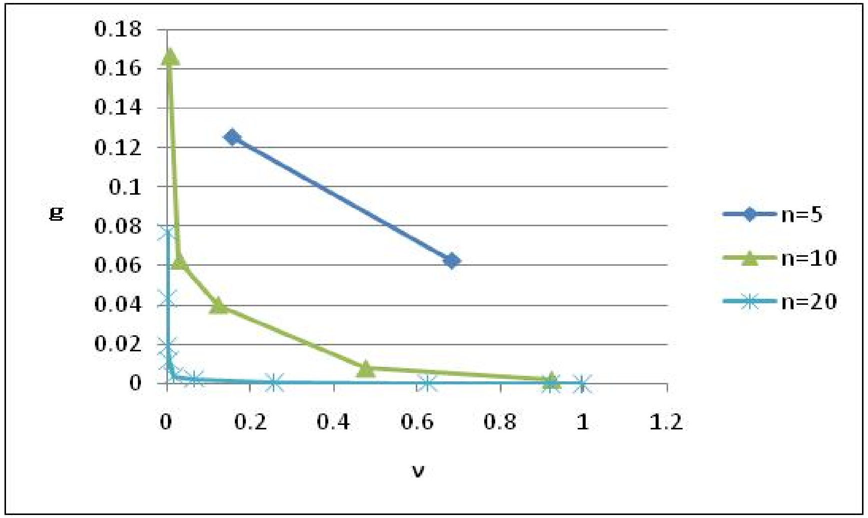


Fig. 4: LP upper bound by codes C_u (p_1=0.5 and p_2=0.25)

(Elastic)

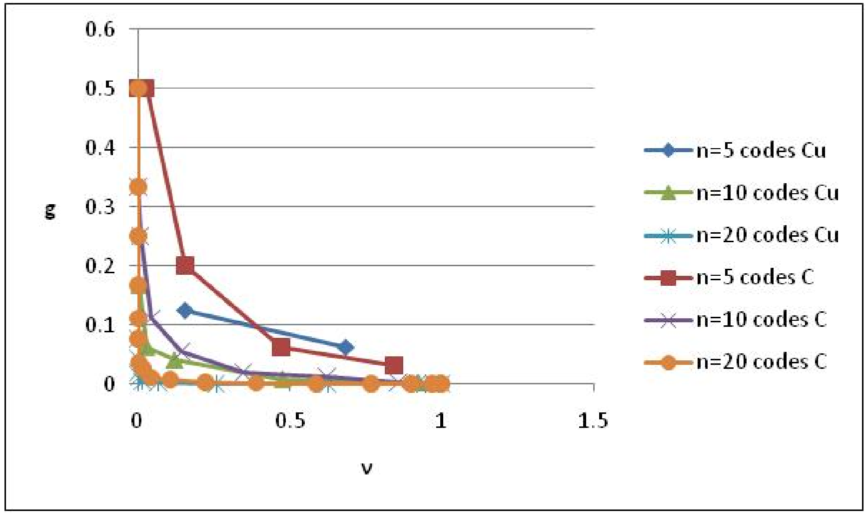


Fig. 5 LP upper bound by codes C and C_u (p_1=0.5 and p_2=0.25)

(Elastic, and Flexible)

サイバー大学 Cyber University

A. Binomial Distribution (BD)

(2) Cases by UEP codes C_u

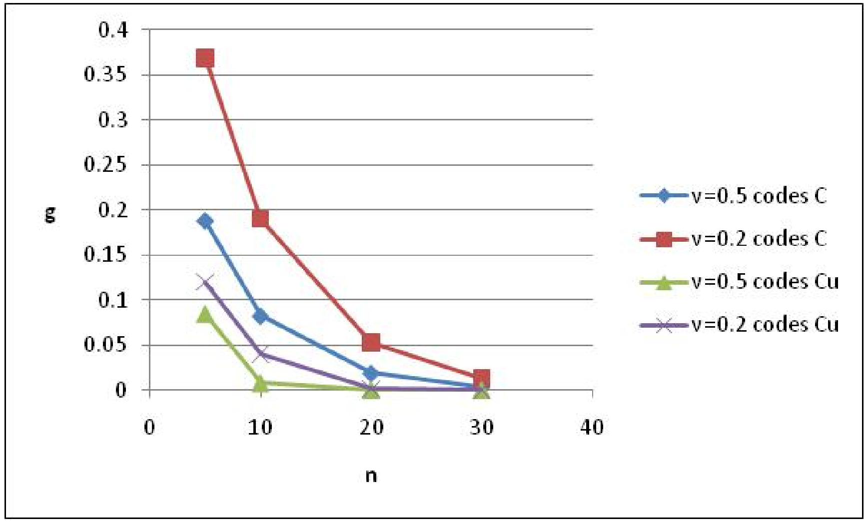


Fig. 6: LP upper bound by codes C (p=0.5, v=0.2, and v=0.5) and codes C_u (p_1=0.5, p_2=0.25 , v=0.2, and v=0.5)

(Effective Elastic)

V. Numerical Results

サイバー大学 Cyber University

B. Chernoff Bound (CB)

(1) Case by EEP codes C

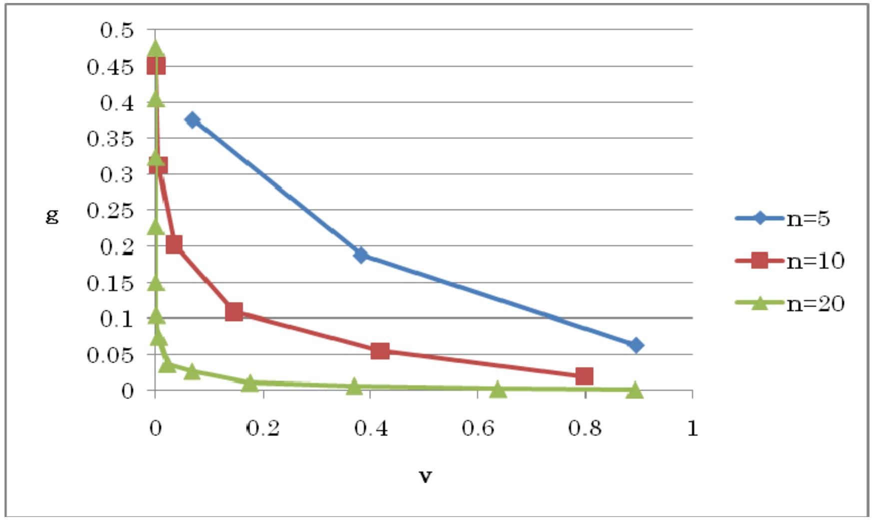


Fig. 7: LP upper bound by codes C (p=0.3)

(Elastic)

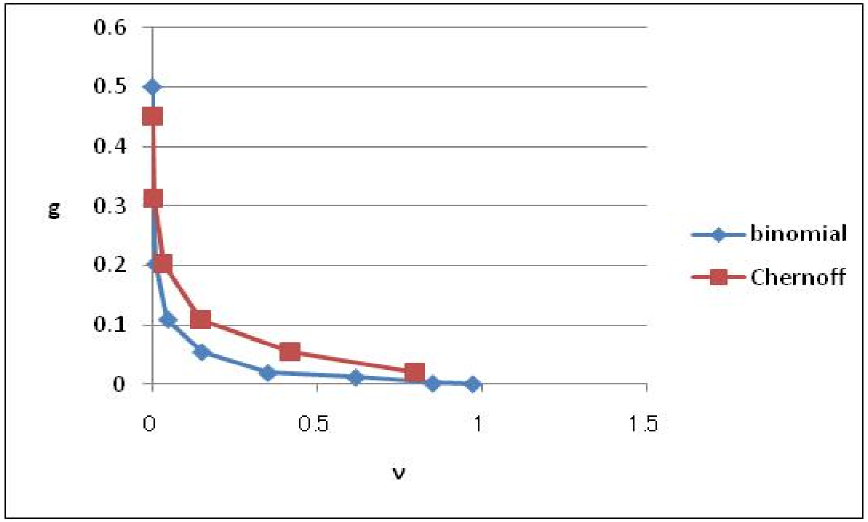


Fig. 8: Binomial distribution and Chernoff bound by codes C using LP upper bound (p=0.3, and n=10)

(Difference of distributions)

V. Numerical Results

B. Chernoff Bound (CB)

(2) Case by UEP codes C_u

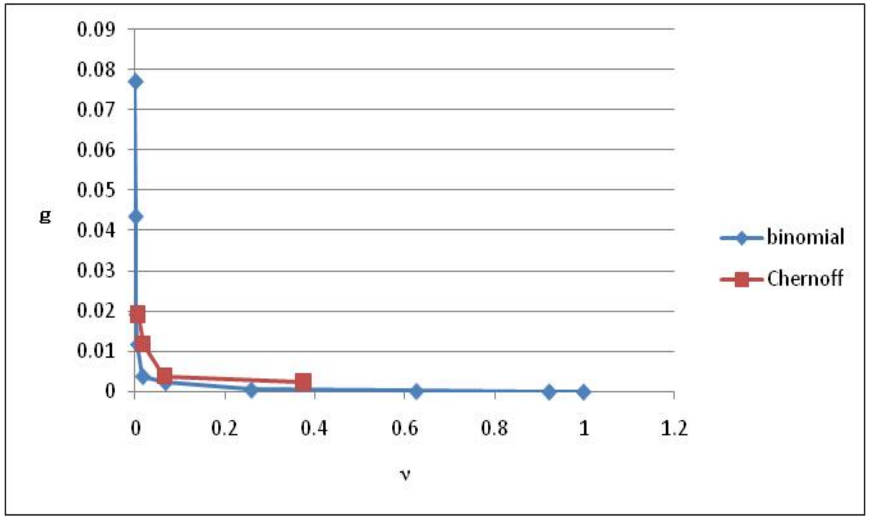


Fig. 9: Binomial distribution and Chernoff bound by codes C_u ((p_1=0.5, p_2=0.25 , and n=0.5)

(Difference of distributions)

V. Numerical Results

C. Discussion

From Fig. 2-9

(1)Elastic: Fig. 2, and 5

Effective Elastic: Fig. 6

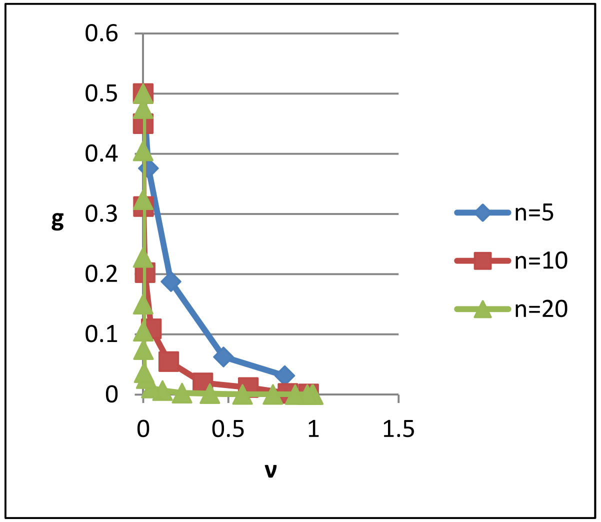


Fig. 2: LP upper bound by Codes C (p=0.5)

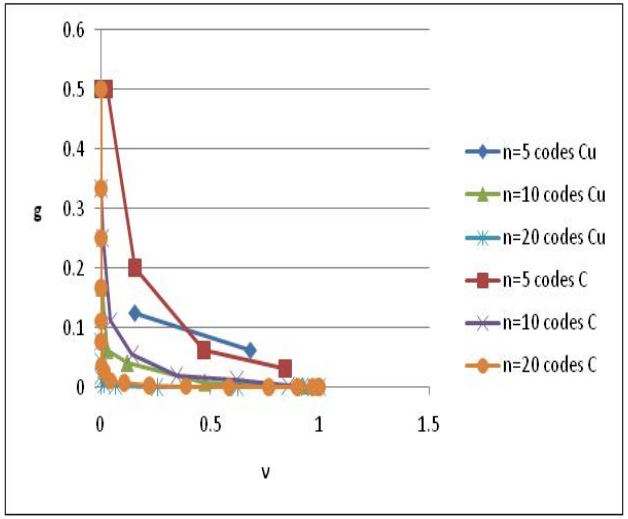


Fig. 5 LP upper bound by codes C and C_u (p_1=0.5 and p_2=0.25)

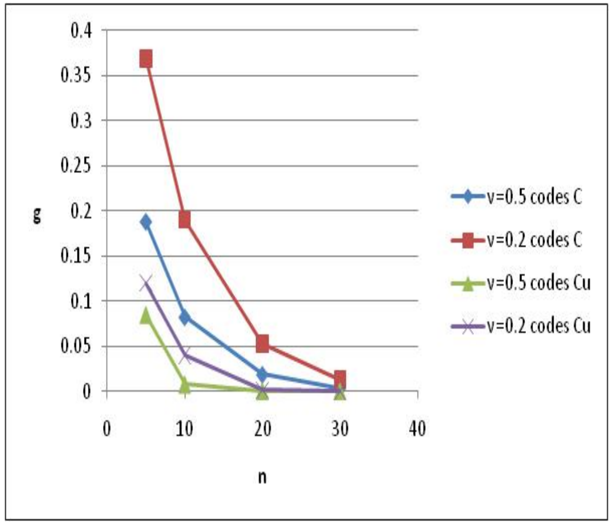


Fig. 6: LP upper bound by codes C (p=0.5, v=0.2, and v=0.5) and codes C_u (p_1=0.5, p_2=0.25 , v=0.2, and v=0.5)

V. Numerical Results

C. Discussion

(2) **Flexible:** Fig. 5

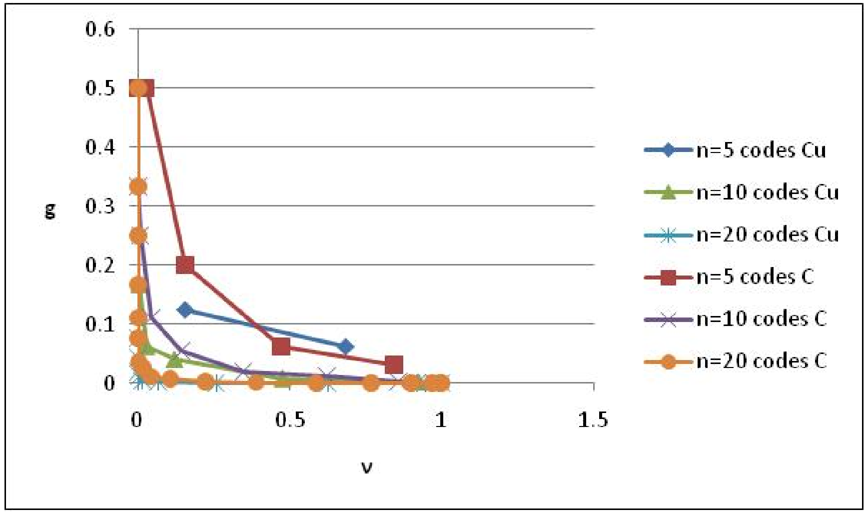


Fig. 5 LP upper bound by codes C and C_u (p_1=0.5 and p_2=0.25)

サイバー大学 Cyber University

C. Discussion

(3) Bounding Techniques

LP upper bound and Gilbert lower bound: Fig. 3

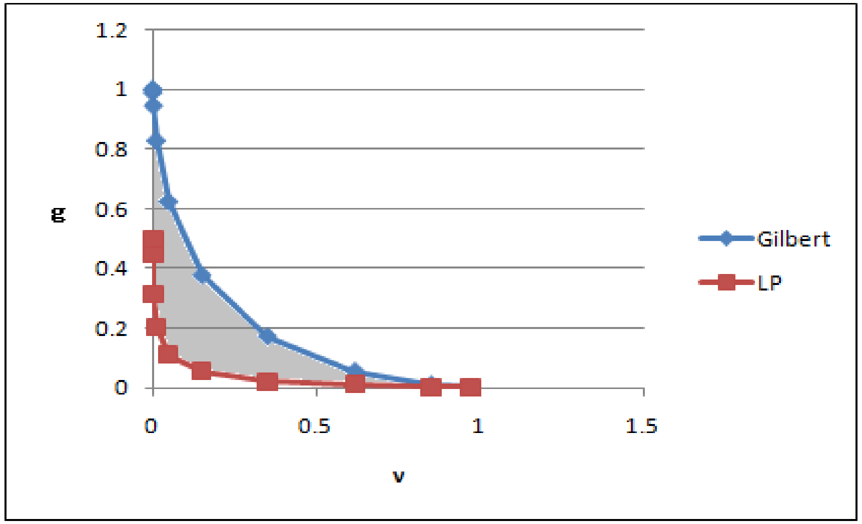


Fig. 3 LP upper bound and Gilbert lower bound by codes C (p=0.3 and n=10)

V. Numerical Results

サイバー大学 Cyber University

C. Discussion

(3) Bounding Techniques

Binominal distribution and Chernoff bound: Figs. 8 & 9

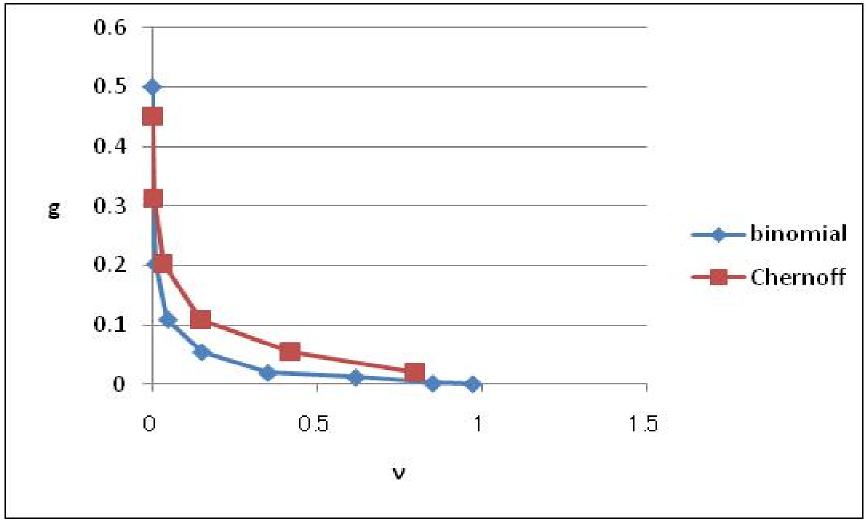


Fig. 8: Binomial distribution and Chernoff bound by codes C using LP upper bound (p=0.3, and n=10)

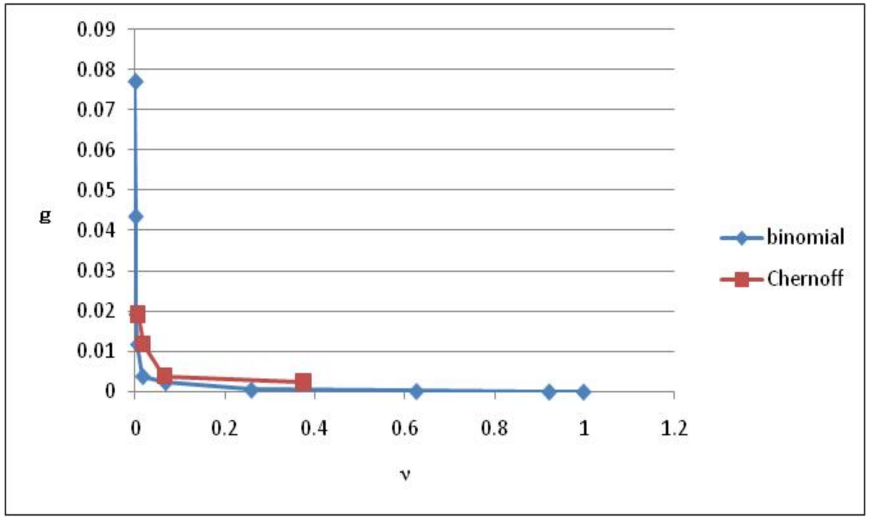


Fig. 9: Binomial distribution and Chernoff bound by codes C_u ((p_1=0.5, p_2=0.25 , and n=0.5)

VI. Concluding Remarks

(1) $g = g(v, n)$ is

(i) an **elastic** function

(ii) an **effective elastic** function

(iii) a **flexible function** for UEP codes compared to EEP codes

The (i) states that tolerating a small access performance degradation in v introduces a drastic saving of the cost g, and

(ii) this property can be effectively enhanced by the number of attribute n becomes large. We can also remark that

(iii) the UEP codes C_u are useful for the disk allocation methods if the probabilities of occurrences of the * are not uniform.

VI. Concluding Remarks

サイバー大学 Cyber University

(2) Generalization:

LP upper bounds: 2-split codes → L-split codes (L>3)

q=2 → $q \geq 3$ (q: power of a prime)

(3) Further research

If we find $g = F(v, n)$ by Chernoff bound, then we can discuss to show analytically whether the system has effective elastic or not.