

Redundancy Allocation of Partitioned Linear Block Codes

Yongjune Kim and B. V. K. Vijaya Kumar

Dep. of Electrical & Computer Eng., Data Storage Systems Center (DSSC)

Carnegie Mellon University

Pittsburgh, PA, USA

Email: yongjunekim@cmu.edu, kumar@ece.cmu.edu

Abstract—Most memories suffer from both permanent defects and intermittent random errors. The partitioned linear block codes (PLBC) were proposed by Heegard to efficiently mask stuck-at defects and correct random errors. The PLBC have two separate redundancy parts for defects and random errors. In this paper, we investigate the allocation of redundancy between these two parts. The optimal redundancy allocation will be investigated using simulations and the simulation results show that the PLBC can significantly reduce the probability of decoding failure in memory with defects. In addition, we will derive the upper bound on the probability of decoding failure of PLBC and estimate the optimal redundancy allocation using this upper bound. The estimated redundancy allocation matches the optimal redundancy allocation well.

I. INTRODUCTION

Most memory systems (e.g., flash memory, phase-change memory, etc.) exhibit two types of imperfections that threaten the data reliability. The first type is a defective memory cell, i.e., defect, whose cell value is stuck-at a particular value independent of the input. For example, some of the cells of a binary memory may be stuck-at 0, and when a 1 is attempted to be written into a stuck-at 0 cell, an error results. The second type of imperfection is a noisy cell which can occasionally result in a random error. The distinction between these two types of imperfections is that defects are permanent, whereas random errors caused by noise are intermittent. Often the terms hard and soft errors are used to describe stuck-at errors of defects and noise-induced random errors, respectively [1]–[4].

By carefully testing the memory, it is possible to know the defect information such as locations and stuck-at values, and this information can be exploited in the encoder and/or the decoder for more efficient coding. This problem was first addressed by Kuznetsov and Tsybakov [1]. They assumed that the locations and stuck-at values of the defects are available to the encoder, but not to the decoder [1], [2].

Later, Heegard proposed the partitioned linear block codes (PLBC) that efficiently incorporate the defect information in the encoding process and are capable of correcting both stuck-at errors (due to defects) and random errors [4]. Recently, his work has drawn attention for nonvolatile memories because flash memories and phase change memories (PCM) suffer from defects as well as random errors [5], [6].

The PLBC require two generator matrices. One of them is for correcting stuck-at errors by masking defects. Masking

defects is to find a codeword whose values at the locations of defects match the stuck-at values at those locations [2], [4]. The other generator matrix is for correcting random errors, which is same as a generator matrix of standard error control coding.

Since the PLBC have two generator matrices, we can separate the redundancy for masking defects from the redundancy for correcting random errors [4]. We assume that the number of redundant symbols for masking defects and correcting random errors are l and r , respectively. The total redundancy will be $l + r$, which is same as $n - k$ (where n is the codeword size and k is the message size). Note that the code rate is k/n .

The fact that the redundancy of PLBC can be divided into two parts leads to the problem of redundancy allocation. The objective is to find an optimal redundancy allocation between l and r in order to minimize the probability of decoding failure. Not surprisingly, the optimal redundancy allocation depends on the channel. If a channel exhibits only defects, we should allot all redundancy to masking defects and the optimal redundancy allocation will be $(l^*, r^*) = (n - k, 0)$. Meanwhile, the optimal redundancy allocation for a channel with only random errors will be $(l^*, r^*) = (0, n - k)$.

In this paper, the optimal redundancy allocation for general channels that exhibit both defects and random errors will be investigated. In addition, we will derive an upper bound on the probability of decoding failure. With this upper bound, we can readily obtain an estimate (\hat{l}, \hat{r}) for the optimal redundancy allocation (l^*, r^*) . The estimate based on the upper bound matches the optimal redundancy allocation well.

The rest of the paper is as follows. Section II explains the channel model and PLBC. In section III, we will discuss the optimal redundancy allocation of PLBC. In Section IV, we will derive the upper bound on the probability of decoding failure and compare the optimal redundancy allocation with the estimate based on our upper bound. Section V concludes the paper.

II. PARTITIONED LINEAR BLOCK CODES [4], [7]

A. Channel Model

In [3], [4], the channel model for memories with defects has been introduced. The model assumes both stuck-at defects and additive random errors. In this paper, we will use the notation of [4], [7].

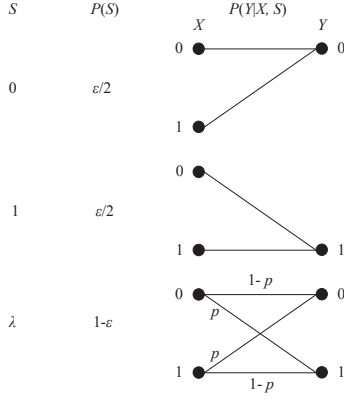


Fig. 1. Channel model for binary memories with defects.

Let q be a power of a prime and F_q be the Galois field with q elements. Let F_q^n denote the set of all n -tuples over F_q . Define an additional variable “ λ ”, $\tilde{F}_q = F_q \cup \lambda$ and define the “ \circ ” operator $\circ : F_q \times \tilde{F}_q \rightarrow F_q$ by

$$x \circ s = \begin{cases} x, & \text{if } s = \lambda; \\ s, & \text{if } s \neq \lambda. \end{cases} \quad (1)$$

An n -cell memory with defects and random errors is modeled by

$$\mathbf{y} = (\mathbf{x} \circ \mathbf{s}) + \mathbf{z} \quad (2)$$

where \mathbf{x} is the vector to be stored, \mathbf{z} is the random error vector and \mathbf{s} is the defect vector. The addition “+” is defined over the field F_q and both + and \circ operate on the vectors component wise.

The number of defects u is equal to the number of non- λ components in \mathbf{s} , and the number of random errors is defined by $t = \|\mathbf{z}\|$ where $\|\cdot\|$ is the Hamming weight of the vector.

A stochastic model for the generation of defects and random errors in a memory cell is obtained by assigning probabilities to defects and random error events. The (ε, p) q -symmetric discrete memoryless memory cell (q -SDMMC) is modeled by $Y = (X \circ S) + Z$, where $X, Y, Z \in F_q, S \in \tilde{F}_q$,

$$P(S = s) = \begin{cases} 1 - \varepsilon, & s = \lambda; \\ \varepsilon/q, & s \neq \lambda, \end{cases} \quad (3)$$

$$P(Z = z | S = \lambda) = \begin{cases} 1 - p, & z = 0; \\ p/(q-1), & z \neq 0. \end{cases}$$

Fig. 1 illustrates the channel model for memories with defects when $q = 2$.

B. Partitioned Linear Block Codes

In [4], Heegard proposed the $[n, k, l]$ PLBC which is a pair of linear subspaces $\mathcal{C}_1 \subset F_q^n$ and $\mathcal{C}_0 \subset F_q^n$ of dimension k and l such that $\mathcal{C}_1 \cap \mathcal{C}_0 = \{\mathbf{0}\}$. Then the direct sum

$$\mathcal{C} \triangleq \mathcal{C}_1 + \mathcal{C}_0 = \{\mathbf{c} = \mathbf{c}_1 + \mathbf{c}_0 | \mathbf{c}_1 \in \mathcal{C}_1, \mathbf{c}_0 \in \mathcal{C}_0\} \quad (4)$$

is an $[n, k+l]$ linear block code (LBC) with a generator matrix $G = [G_1^T \ G_0^T]^T$ where G_1 generates \mathcal{C}_1 and G_0 generates \mathcal{C}_0 (superscript T denotes transpose). The parity check matrix H is $r \times n$ matrix with $k+l+r = n$. A message inverse matrix \tilde{G}_1 is defined as $k \times n$ matrix with $G_1 \tilde{G}_1^T = I_k$ (the k -dimensional identity matrix), and $G_0 \tilde{G}_1^T = \mathbf{0}_{l,k}$ (the $l \times k$ zero matrix) [4].

The encoding and decoding of PLBC are as follows [4].

Encoding: To encode a message $\mathbf{w} \in F_q^k$ into a codeword $\mathbf{c} = \mathbf{w}G_1 + \mathbf{d}G_0$ where $\mathbf{d} \in F_q^l$ is chosen to minimize $\|(\mathbf{c} \circ \mathbf{s}) - \mathbf{c}\|$.

Decoding: Receive $\mathbf{y} = (\mathbf{c} \circ \mathbf{s}) + \mathbf{z}$. Compute the syndrome $\mathbf{v} = \mathbf{y}H^T$ and choose $\hat{\mathbf{z}} \in F_q^n$ which minimizes $\|\mathbf{z}\|$ subject to $\mathbf{z}H^T = \mathbf{v}$. Then $\hat{\mathbf{w}} = \hat{\mathbf{c}}\tilde{G}_1^T$ where $\hat{\mathbf{c}} = \mathbf{y} - \hat{\mathbf{z}}$.

The encoding of PLBC requires two generator matrices, namely, G_0 and G_1 . First, G_1 encodes a message \mathbf{w} for correcting random errors. Next, G_0 is used to mask defects by $\mathbf{d}G_0$ in order to minimize $\|(\mathbf{c} \circ \mathbf{s}) - \mathbf{c}\|$. The redundancy for masking defects is l and the redundancy for correcting random errors is r . Let (l, r) denote the redundancy of $[n, k, l]$ PLBC. The total redundancy is $l + r = n - k$.

A pair of minimum distances (d_0, d_1) of an $[n, k, l]$ PLBC are given by

$$d_0 = \min_{\substack{\mathbf{c} \neq \mathbf{0} \\ \mathbf{c}G_0^T = \mathbf{0}}} \|\mathbf{c}\|, \quad (5)$$

$$d_1 = \min_{\substack{\mathbf{c}\tilde{G}_1^T \neq \mathbf{0} \\ \mathbf{c}H^T = \mathbf{0}}} \|\mathbf{c}\| \quad (6)$$

where d_1 is greater than or equal to the minimum distance of the $[n, k+l]$ LBC with parity check matrix H , while d_0 is the minimum distance of the $[n, k+r]$ LBC with the parity check matrix G_0 [4]. Note that this $[n, k+r]$ LBC uses G_0 as a parity check matrix instead of a generator matrix.

Theorem 1 [4]: An $[n, k, l]$ PLBC with minimum distances (d_0, d_1) is a u -defect, t -error correcting code if

$$u < d_0 \text{ and } 2t < d_1$$

or

$$u \geq d_0 \text{ and } 2(t + u - (d_0 - 1)) < d_1.$$

If $u < d_0$, all defects will be successfully masked and $\|(\mathbf{c} \circ \mathbf{s}) - \mathbf{c}\| = 0$. Otherwise, it may be that $\|(\mathbf{c} \circ \mathbf{s}) - \mathbf{c}\| \neq 0$ which results in masking failure. When $d_0 - 1$ defects among u defects have been masked, the number of unmasked defects is $u - (d_0 - 1)$. These unmasked defects will be regarded as random errors in the decoder.

Note that the masking succeeds in the encoder if and only if $\mathbf{c} \circ \mathbf{s} = \mathbf{c}$. In addition, $\hat{\mathbf{w}} = \mathbf{w}$ means the decoding success.

C. Two-step Encoding Scheme

The encoding of PLBC includes an implicit optimization problem which can be formulated as follows [4]–[6].

$$\begin{aligned} \mathbf{d}^* &= \underset{\mathbf{d}}{\operatorname{argmin}} \left\| \mathbf{d}G_0^{\Psi_u} + \mathbf{w}G_1^{\Psi_u} - \mathbf{s}^{\Psi_u} \right\| \\ &= \underset{\mathbf{d}}{\operatorname{argmin}} \left\| \mathbf{d}G_0^{\Psi_u} + \mathbf{b}^{\Psi_u} \right\| \end{aligned} \quad (7)$$

TABLE I
TWO-STEP ENCODING SCHEME

Step 1:

- Try to solve (8).
 - If $u < d_0$, a solution \mathbf{d} to (8) always exists and go to end.
 - If $u \geq d_0$, a solution \mathbf{d} to (8) exists so long as (9) holds.
 - * If \mathbf{d} exists, go to end.
 - * Otherwise, go to step 2.

Step 2:

- Choose $d_0 - 1$ locations among u defects and define $\Psi_{d_0-1} = [i_1, \dots, i_{d_0-1}]$.
- Solve $\mathbf{d}G_0^{\Psi_{d_0-1}} = \mathbf{b}^{\Psi_{d_0-1}}$ instead of (8).

End

where $\Psi_u = [i_1, \dots, i_u]$ indicates the locations of u defects and $\mathbf{b}^{\Psi_u} = \mathbf{w}G_1^{\Psi_u} - \mathbf{s}^{\Psi_u}$. We use the notation of $s^{\Psi_u} = [s_{i_1}, \dots, s_{i_u}]$, $G_0^{\Psi_u} = [\mathbf{g}_{0,i_1}, \dots, \mathbf{g}_{0,i_u}]$, and $G_1^{\Psi_u} = [\mathbf{g}_{1,i_1}, \dots, \mathbf{g}_{1,i_u}]$ where $\mathbf{g}_{0,i}$ and $\mathbf{g}_{1,i}$ are the i -th columns of G_0 and G_1 respectively. Then, $\|\mathbf{d}G_0^{\Psi_u} + \mathbf{b}^{\Psi_u}\|$ is the number of unmasked defects.

In order to mask all defects, we should have a solution \mathbf{d} satisfying

$$\mathbf{d}G_0^{\Psi_u} = \mathbf{b}^{\Psi_u}. \quad (8)$$

It is true that (8) has at least one solution if and only if

$$\text{rank} \left(\left(G_0^{\Psi_u} \right)^T \right) = \text{rank} \left(\left(G_0^{\Psi_u} \right)^T \middle| \left(\mathbf{b}^{\Psi_u} \right)^T \right) \quad (9)$$

where $((G_0^{\Psi_u})^T | (\mathbf{b}^{\Psi_u})^T)$ is the augmented matrix [8].

If $u < d_0$, $\text{rank} \left(G_0^{\Psi_u} \right)$ is always u by (5). Therefore, (9) holds and a solution \mathbf{d} satisfying (8) exists. Gaussian elimination or some other solution methods for linear equations can be used to solve (8).

However, if $u \geq d_0$, the optimal solution of (7) may fail to mask all defects. In addition, the computational complexity for solving the optimization problem is exponential, which is impractical as l increases [6].

In [4], a modified formulation of (7) was described, which chooses only $\min(d_0 - 1, u)$ locations among u defects instead of solving the optimization problem. Then, a solution of the modified formulation exists. This scheme achieves the u -defect, t -error correcting code of Theorem 1 [4]. We call it *one-step encoding scheme*.

In [7], the *two-step encoding scheme* has been proposed, which can mask more defects than one-step encoding scheme. The computational complexity of the two-step encoding scheme is comparable to that of the one-step encoding scheme. The two-step encoding scheme is summarized in Table I. We will use this two-step encoding scheme for encoding of PLBC.

III. OPTIMAL REDUNDANCY ALLOCATION

In order to minimize the probability of decoding failure of PLBC, we have to find the optimal redundancy allocation

TABLE II
CHANNELS WITH THE SAME C_{\min}

Channel	p	ε	C_{\min}	C_{\max}
1	4.0×10^{-3}	0	0.9624	0.9624
2	3.0×10^{-3}	2.0×10^{-3}	0.9624	0.9686
3	2.5×10^{-3}	3.0×10^{-3}	0.9624	0.9719
4	2.0×10^{-3}	4.0×10^{-3}	0.9624	0.9753
5	1.0×10^{-3}	6.0×10^{-3}	0.9624	0.9827
6	5.0×10^{-4}	7.0×10^{-3}	0.9624	0.9868
7	0	8.0×10^{-3}	0.9624	0.9920

TABLE III
ALL POSSIBLE REDUNDANCY ALLOCATION CANDIDATES OF
[$n = 1023, k = 923, l$] PBCH CODES

Code	l	r	d_0	d_1	Notes
0	0	100	0	21	only correcting random errors
1	10	90	3	19	
2	20	80	5	17	
3	30	70	7	15	
4	40	60	9	13	
5	50	50	11	11	
6	60	40	13	9	
7	70	30	15	7	
8	80	20	17	5	
9	90	10	19	3	
10	100	0	21	0	only masking defects

(l^*, r^*) . This problem can be formulated as follows.

$$(l^*, r^*) = \underset{(l, r)}{\text{argmin}} \quad P(\text{decoding failure}) \quad (10)$$

subject to $l + r = n - k$

(l^*, r^*) depends on the parameters of the given channel such as ε and p . Without an expression for $P(\text{decoding failure})$ as a function of (l, r) , this optimization problem cannot be solved. Unfortunately, it is difficult to obtain the exact mathematical expression for $P(\text{decoding failure})$. Therefore, we obtained (l^*, r^*) via simulation of the given channel and the given PLBC.

We will consider the channels of Table II. All channels of Table II are chosen to have the same C_{\min} , which is the channel capacity when neither the encoder nor the decoder knows the defect information [3]. C_{\min} is given by

$$C_{\min} = 1 - h \left((1 - \varepsilon)p + \frac{\varepsilon}{2} \right) \quad (11)$$

where $h(x) = -x \log_2 x - (1 - x) \log_2 (1 - x)$. Note that (11) equals the capacity of a binary symmetric channel (BSC) with parameter $\tilde{p} = (1 - \varepsilon)p + \frac{\varepsilon}{2}$. All channels of Table II have the same $\tilde{p} \cong 4.0 \times 10^{-3}$. If either the encoder or the decoder knows the information of defects, the maximum capacity can be achieved [3]. The capacity is given by

$$C_{\max} = (1 - \varepsilon)(1 - h(p)). \quad (12)$$

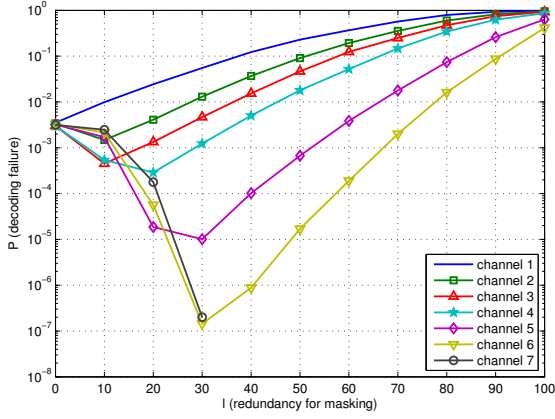


Fig. 2. $P(\text{decoding failure})$ of channels in Table II.

TABLE IV
OPTIMAL REDUNDANCY ALLOCATION l^* AND ITS ESTIMATE \hat{l} BY UPPER
BOUND FOR $[n = 1023, k = 923, l]$ PBCH CODES

Channel	l^*	\hat{l}	$ l^* - \hat{l} $
1	0	0	0
2	10	10	0
3	10	20	10
4	20	20	0
5	30	30	0
6	30	30	0
7	100	100	0

Each channel of Table II has different C_{\max} . As ε increases, we can obtain more information about the channel, which results in the increase of C_{\max} .

For these channels of Table II, we apply several kinds of $[n = 1023, k = 923, l]$ PLBC. With the fixed code rate of $\frac{k}{n} = \frac{923}{1023}$, the redundancy allocation between (l, r) will be varied. All possible redundancy allocation candidates of the partitioned Bose-Chaudhuri-Hocquenghem (PBCH) code are presented in Table III. The PBCH code is a special class of PLBC and its generator matrices and minimum distances can be designed by a similar method of standard BCH codes [4].

Fig. 2 shows the simulation results for the channels of Table II. The optimal redundancy allocation for channel 1 (only random errors) will be $(l^* = 0, r^* = n - k = 100)$, i.e., we have to allot all redundancy for correcting random errors, which is equivalent to the standard BCH code. The more defects a channel has, the larger l is expected to be for the optimal redundancy allocation. Eventually, the optimal redundancy for channel 7 will be $(l^* = n - k = 100, r^* = 0)$ though the simulation result of channel 7 is incomplete because of its impractical computational complexity. The optimal l^* for all channels of Table II are presented in the second column of Table IV. The optimal r^* can be obtained by $r^* = n - k - l^*$.

In addition, it is worth mentioning that $P(\text{decoding failure})$ for (l^*, r^*) improves as ε increases in Fig. 2. The reason is that the PLBC can exploit more information of defects,

which has been indicated by C_{\max} of Table II. Note that $P(\text{decoding failure})$ for all channels will be same if we use the redundancy allocation $(l = 0, r = n - k)$ when the encoder does not use the information of defects.

To find the optimal redundancy allocation (l^*, r^*) by simulation requires significant computations. Therefore, we will also investigate using the upper bound on $P(\text{decoding failure})$ instead of the simulation for estimating the optimal redundancy allocation.

IV. UPPER BOUND ON $P(\text{DECODING FAILURE})$

In this section, the upper bound on $P(\text{decoding failure})$ will be derived. For convenience, we will define three random variables of D , M , and U .

$$D = \begin{cases} 0, & \hat{\mathbf{w}} \neq \mathbf{w} \text{ (decoding failure);} \\ 1, & \hat{\mathbf{w}} = \mathbf{w} \text{ (decoding success),} \end{cases} \quad (13)$$

$$M = \begin{cases} 0, & \mathbf{c} \circ \mathbf{s} \neq \mathbf{c} \text{ (masking failure);} \\ 1, & \mathbf{c} \circ \mathbf{s} = \mathbf{c} \text{ (masking success)} \end{cases}$$

In addition, U represents the number of defects per codeword of PLBC. The probability of decoding failure $P(D = 0)$ will be given by

$$P(D = 0) = P(M = 0, D = 0) + P(M = 1, D = 0). \quad (14)$$

First, we will derive the upper bound on $P(M = 0, D = 0)$. By the chain rule, $P(M = 0, D = 0)$ is given by

$$P(M = 0, D = 0) = \sum_{u=1}^n \{P(U = u) \cdot P(M = 0|U = u) \cdot P(D = 0|M = 0, U = u)\}. \quad (15)$$

Note that we do not need to consider $u = 0$ since the masking always succeeds for $u = 0$.

By (3), we can assume that U is a binomial random variable. Therefore, $P(U = u)$ is given by

$$P(U = u) = \binom{n}{u} \varepsilon^u (1 - \varepsilon)^{n-u}, \quad 0 \leq u \leq n. \quad (16)$$

In [7], the following upper bound on $P(M = 0|U = u)$ was derived.

$$P(M = 0|U = u) \leq \frac{\sum_{w=d_0}^u A_w \binom{n-w}{u-w}}{\binom{n}{u}} \quad (17)$$

where A_w is the number of codewords of weight w in the LBC with the parity check matrix G_0 . Note that $A_w = 0$ for $0 < w < d_0$ by (5).

Also, $P(D = 0|M = 0, U = u)$ is given by

$$P(D = 0|M = 0, U = u) = P(\{u - (d_0 - 1)\} + t > t_1) \\ = P(t \geq t_1 + d_0 - u) \quad (18)$$

where $u - (d_0 - 1)$ represents the number of unmasked defects and t is the number of random errors. In addition, $t_1 = \lfloor \frac{d_1 - 1}{2} \rfloor$ (where $\lfloor x \rfloor$ is the largest integer not greater than x) is the error correcting capability of \mathcal{C}_1 . Since the number of random

errors can be modeled by the binomial random variable by (3), $P(D = 0|M = 0, U = u)$ is given by

$$P(D = 0|M = 0, U = u) = \sum_{t=t_1+d_0-u}^{n-u} \binom{n-u}{t} p^t (1-p)^{n-u-t}. \quad (19)$$

By substituting (16), (17) and (19) into (15), the upper bound on $P(M = 0, D = 0)$ is given by

$$P(M = 0, D = 0) \leq \sum_{u=d_0}^n \left\{ \binom{n}{u} \varepsilon^u (1-\varepsilon)^{n-u} \cdot \frac{\sum_{w=d_0}^u A_w \binom{n-w}{u-w}}{\binom{n}{u}} \cdot \sum_{t=t_1+d_0-u}^{n-u} \binom{n-u}{t} p^t (1-p)^{n-u-t} \right\}. \quad (20)$$

Now, we will derive the upper bound on $P(M = 1, D = 0)$ in (14). By the chain rule,

$$P(M = 1, D = 0) = \sum_{u=0}^n \{P(U = u) \cdot P(M = 1|U = u) \cdot P(D = 0|M = 1, U = u)\} \quad (21)$$

where $P(U = u)$ was given by (16) and $P(M = 1|U = u) \leq 1$. Also, $P(D = 0|M = 1, U = u)$ is given by

$$P(D = 0|M = 1, U = u) = P(t > t_1) = \sum_{t=t_1+1}^{n-u} \binom{n-u}{t} p^t (1-p)^{n-u-t}. \quad (22)$$

By substituting (16), (22), and $P(M = 1|U = u) \leq 1$ into (21), the upper bound on $P(M = 1, D = 0)$ is given by

$$P(M = 1, D = 0) \leq \sum_{u=0}^n \left\{ \binom{n}{u} \varepsilon^u (1-\varepsilon)^{n-u} \cdot \sum_{t=t_1+1}^{n-u} \binom{n-u}{t} p^t (1-p)^{n-u-t} \right\}. \quad (23)$$

By substituting (20) and (23) into (14), $P(D = 0)$ is given by

$$P(D = 0) \leq \sum_{u=d_0}^n \left\{ \binom{n}{u} \varepsilon^u (1-\varepsilon)^{n-u} \cdot \frac{\sum_{w=d_0}^u A_w \binom{n-w}{u-w}}{\binom{n}{u}} \cdot \sum_{t=t_1+d_0-u}^{n-u} \binom{n-u}{t} p^t (1-p)^{n-u-t} \right\} + \sum_{u=0}^n \left\{ \binom{n}{u} \varepsilon^u (1-\varepsilon)^{n-u} \cdot \sum_{t=t_1+1}^{n-u} \binom{n-u}{t} p^t (1-p)^{n-u-t} \right\}. \quad (24)$$

For $\varepsilon = 0$, we can claim that $P(M = 0, D = 0) = 0$ since the upper bound on $P(M = 0, D = 0)$ becomes zero by (20).

In addition, the terms of (23) also become zero for $u \geq 1$. Therefore, (24) will be changed into

$$P(D = 0) \leq \sum_{t=t_1+1}^n \binom{n}{t} p^t (1-p)^{n-t}. \quad (25)$$

For $l = 0$, the PLBC is same as the standard error control coding which does not know the information of defects. Therefore, $P(D = 0)$ will be as follows instead of (24).

$$P(D = 0) \leq \sum_{t=t_1+1}^n \binom{n}{t} \tilde{p}^t (1-\tilde{p})^{n-t} \quad (26)$$

By (24), (25) and (26), we can readily obtain the upper bound on the probability of decoding failure for the given channel and all possible redundancy allocation candidates such as Table III. Then, we can choose the redundancy allocation minimizing the upper bound instead of the probability of decoding failure. The redundancy allocation that minimizes the upper bound is the estimate of the optimal redundancy allocation.

The estimates \hat{l} for all channels of Table II are presented in the third column of Table IV. Note that $\hat{r} = n - k - \hat{l}$. The estimates by the upper bound match the optimal redundancy allocation well. The only exception occurs in channel 3, and the difference of l is only 10 bits, which is the smallest difference among all possible redundancy allocation candidates of Table III.

V. CONCLUSION

The redundancy allocation of PLBC was discussed and the optimal redundancy allocation was investigated via simulation. In addition, we derived an upper bound on decoding failure probability of PLBC for general channels, and used it for estimating the optimal redundancy allocation. The estimated redundancy allocation is very similar to the optimal redundancy allocation whereas it requires much less computation than simulations required for determining the optimal redundancy allocation.

REFERENCES

- [1] A. V. Kuznetsov and B. S. Tsybakov, "Coding in a memory with defective cells," *Problemy Peredachi Informatsii*, vol. 10, no. 2, pp. 52–60, Apr.–Jun. 1974.
- [2] A. V. Kuznetsov, T. Kasami, and S. Yamamura, "An error correcting scheme for defective memory," *IEEE Trans. Inf. Theory*, vol. 24, no. 6, pp. 712–718, Nov. 1978.
- [3] C. Heegard and A. El Gamal, "On the capacity of computer memory with defects," *IEEE Trans. Inf. Theory*, vol. 29, no. 5, pp. 731–739, Sep. 1983.
- [4] C. Heegard, "Partitioned linear block codes for computer memory with 'stuck-at' defects," *IEEE Trans. Inf. Theory*, vol. 29, no. 6, pp. 831–842, Nov. 1983.
- [5] L. A. Lastras-Montano, A. Jagmohan, and M. M. Franceschini, "Algorithms for memories with stuck cells," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jun. 2010, pp. 968–972.
- [6] E. Hwang, B. Narayanaswamy, R. Negi, and B. V. K. Vijaya Kumar, "Iterative cross-entropy encoding for memory systems with stuck-at errors," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2011, pp. 1–5.
- [7] Y. Kim and B. V. K. Vijaya Kumar, "Coding for memory with stuck-at defects," accepted to *IEEE Int. Conf. Commun. (ICC)*, Jun. 2013.
- [8] S. H. Friedberg, A. J. Insel, and L. E. Spence, *Linear Algebra*, 3rd ed. Upper Saddle River, NJ: Prentice Hall, 1997.