國立交通大學

電信工程研究所

碩 士 論 文

極碼的可信度傳遞解碼排程

On the Scheduling of Belief Propagation Decoding for Polar Codes

研 究 生： 劉振澧

指導教授： 陸曉峯 教授

中 華 民 國 一 百 年 七 月

極碼的可信度傳遞解碼排程

On the Scheduling of Belief Propagation Decoding for Polar Codes

研 究 生：劉振灃　　　　Student：Chen-Feng Liu

指導教授：陸曉峯　教授　　Advisor：Prof. Hsiao-Feng Lu

國 立 交 通 大 學

電信工程研究所

碩 士 論 文

A Thesis

Submitted to Institute of Communications Engineering

College of Electrical and Computer Engineering

National Chiao Tung University

for the Degree of Master

in

Communications Engineering

July 2011

Hsinchu, Taiwan, Republic of China

中華民國一百年七月

極碼的可信度傳遞解碼排程

學生：劉振澧　　　　　　　　　　　　指導教授:陸曉峯　教授

國立交通大學電信工程研究所碩士班

摘　　要

通道編碼藉由將訊息增加冗餘，讓訊息傳輸更可靠。一直以來有許多專家學者致力於研究通道編碼系統，期望可以建構出一套在傳輸速率不大於通道容量的條件下，可以可靠的傳送訊息的編碼系統。極碼（Polar codes）在使用接續消除解碼（Successive cancellation decoding）的方式下，理論上可以讓傳輸速率達到通道容量。

除了接續消除解碼之外，當我們把碼長增加時，尚有其他解碼方式可以讓錯誤率衰減的更快速。可靠度傳遞解碼（Belief propagation decoding）為其中一種方法。可靠度傳遞解碼已被用來當作極碼的解碼方式，解出的訊息錯誤率相對於接續消除解碼解出的錯誤率，前者較低。

為了方便解碼，接續消除解碼器忽略了某些可利用的訊息，而這些訊息則被可靠度傳遞解碼器善加利用。論文[1]說明了相對於接續消除解碼，將可靠度傳遞解碼作為極碼解碼方式的優越性。

在本篇論文中，我們使用類似接續消除解碼的解碼方式，作為可靠度傳遞解碼的排程。藉由模擬二位元輸入高斯通道，我們可以看出在某些情況下，將可靠度傳遞解碼加入此排程會比單純使用可靠度傳遞解碼有較好的效能。

On the Scheduling of Belief Propagation Decoding for Polar Codes

Student：Chen-Feng Liu Advisors：Prof. Hsiao-Feng Lu

Institute of Communications Engineering
National Chiao Tung University

Abstract

Channel coding is a scheme that adds redundancy to messages for reliable transmission. Many works have been devoting to constructing a coding scheme that can transmit messages reliably at rates set below the channel capacity. Polar code is one such scheme that can achieves channel capacity theoretically under successive cancellation (SC) decoding.

Besides SC decoding, there are other decoding algorithms such that as code length increases, error rates of polar codes decrease more rapidly. For instance, belief propagation (BP) decoding has been used to decode polar codes with error rates better than the SC decoding.

BP decoder uses the information ignored by SC decoder, and its superiority over SC decoder has been established in [1]. In this thesis, we propose a scheduling for the BP decoding of polar codes that resembles the SC decoding. By simulation over binary-input AWGN channel, we see that in some cases, error performance in this schedule is better than that of BP decoding alone.

Acknowledgement

I appreciate my advisor for his teaching to be a good researcher. And many thanks to the mates in Coding science lab, during the past two years, I was glad to do research with all of you.

I am also very thankful to my family and girlfriend. They always encouraged and supported me patiently when I felt upset.

Contents

List of Figures

Chapter 0

Notations and Terminologies

We denote random variables by capital letters, X, and their sample values by the corresponding lower-case letters, x. Matrices are represented by boldface capitals like $\mathbf{G}$, and corresponding boldface lower-case letters with underline and subscript denote row vectors of the matrix, $\underline{\mathbf{g}}_k$ is the kth row vector of matrix $\mathbf{G}$. Vector $(u_1, \cdots, u_N)$ is abbreviated as u_1^N. $\mathcal{A}$ is a subset of integer, $|\mathcal{A}|$ is the size of set $\mathcal{A}$. Let $\underline{u}_{\mathcal{A}}$ denotes $|\mathcal{A}|$-tuple vector, and indices of elements are included in $\mathcal{A}$, i.e., $(u_i : i \in \mathcal{A})$. $\mathbf{1}_{(E)}$ denotes indicator function of event E.

Consider a binary-input discrete memoryless channel (B-DMC) with input alphabet $\mathcal{X}$, output alphabet $\mathcal{Y}$ and transition probability $P(y|x)$, $x \in \mathcal{X}$, $y \in \mathcal{Y}$. The input alphabet $\mathcal{X}$ is GF(2), the output alphabet $\mathcal{Y}$ and transition probability $P(y|x)$ are arbitrary. Bhattacharyya parameter of the B-DMC is

$$Z = \sum_{y \in \mathcal{Y}} \sqrt{P(y|x=0) \cdot P(y|x=1)}$$

Symmetric capacity of B-DMC is mutual information of input and output when input random variable is uniform distributed, i.e.,

$$I(X;Y) = \sum_{y \in \mathcal{Y}} \sum_{x \in \mathcal{X}} \frac{1}{2} P(y|x) \log_2 \frac{P(y|x)}{\frac{1}{2} P(y|x=0) + \frac{1}{2} P(y|x=1)}$$

In this thesis, codes construction will be carried out over GF(2).

Chapter 1

Introduction

Polar codes, introduced by Arikan in [2], are provable to achieve channel capacity for symmetric B-DMCs. As code length grows larger, some of coordinate channels seen by individual bits become more reliable while the others get worse. The effect is called channel polarization. Thus as code length $N \to \infty$, fraction of coordinate channels are noiseless, and we transmit information bits via these reliable coordinate channels. Besides low encoding complexity, the SC decoder, proposed together with polar codes [2], also has low decoding complexity.

1.1 Channel polarization

Channel polarization is an operation that combines N independent copies of B-DMC into a vector channel, and then splits the vector channel into N binary-input coordinate channels. In this thesis, we denote transition probability of vector channel with suffix, i.e., P_N. Besides suffix N, superscript (i) is together with transition probability of the ith coordinate channel, $P_N^{(i)}$.

1.1.1 Channel combination

This operation combines two identical and independent vector channels with input alphabet $\mathcal{X}^{N/2}$ and output alphabet $\mathcal{Y}^{N/2}$ into another vector channel with input alphabet $\mathcal{X}^N$ and output alphabet $\mathcal{Y}^N$ recursively.

The first level of recursion is shown in Fig. 1.1, where u_1 and u_2 are used to produce x_1 and x_2. Then, x_1 and x_2 are transmitted through two independent B-DMCs. We can

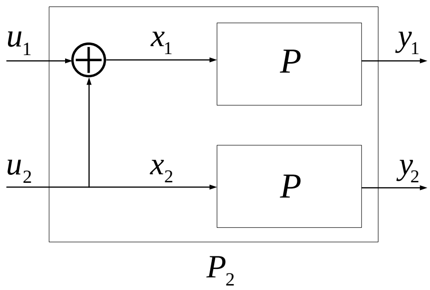


Figure 1.1: Fundamental block of recursive structure.

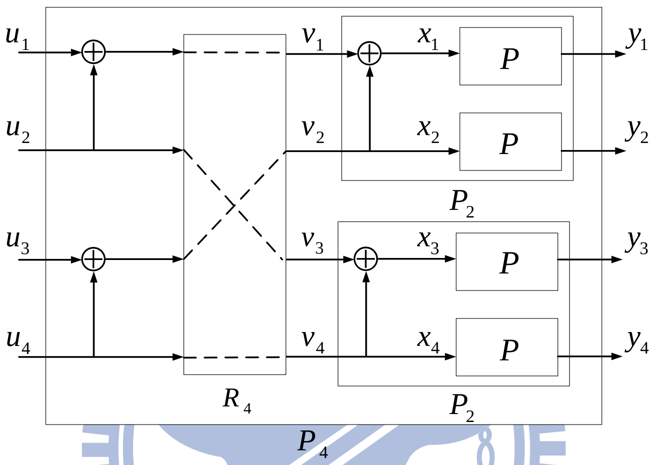


Figure 1.2: Vector channel P_4.

get transition probability of vector channel , $P_2 : \mathcal{X}^2 \to \mathcal{Y}^2$,

$$\begin{aligned}
P_2(y_1, y_2|u_1, u_2) &= P(y_1, y_2|x_1, x_2) \\
&= P(y_1|x_1) \cdot P(y_2|x_2) \\
&= P(y_1|u_1 \oplus u_2) \cdot p(y_2|u_2)
\end{aligned}$$

For $N = 4$, channel combination is shown in Fig. 1.2. At first, the vector (u_1, u_2, u_3, u_4) are calculated over binary field and permuted via R_4; then the permuted vector (v_1, v_2, v_3, v_4) are sent through two identical vector channels P_2. Finally, (x_1, x_2, x_3, x_4) is transmitted through four B-DMCs.

From Fig. 1.1 and Fig. 1.2, the transition probability of vector channel, $P_4 : \mathcal{X}^4 \to \mathcal{Y}^4$, is

$$P_4(y_1, y_2, y_3, y_4|u_1, u_2, u_3, u_4)$$

$$= P_2(y_1, y_2|u_1 \oplus u_2, u_3 \oplus u_4) \cdot P_2(y_3, y_4|u_2, u_4)$$

$$= P(y_1|u_1 \oplus u_2 \oplus u_3 \oplus u_4) \cdot P(y_2|u_3 \oplus u_4) \cdot \ P(y_3|u_2 \oplus u_4) \cdot P(y_4|u_4)$$

To generalize channel combination, we let $N = 2^n$. At first, vector $(u_1, \cdots, u_N)$ is calculated over binary-field and then permuted in block R_N. Permutation function of R_N is

$$v_i = \begin{cases} w_{\frac{k+1}{2}}, & \text{if } k \text{ is odd, } 1 \leq i \leq \frac{N}{2} \\ w_{\frac{k}{2}+\frac{N}{2}}, & \text{if } k \text{ is even, } \frac{N}{2}+1 \leq i \leq N \end{cases}$$

The permuted vector $(v_1, v_2, \cdots, v_N)$ is decomposed into two vectors, $(v_1, v_2, \cdots, v_{\frac{N}{2}})$ and $(v_{\frac{N}{2}+1}, \cdots, v_N)$. The former is sent through the upper vector channel, $P_{N/2}$, and the latter is sent through the lower one. With n levels recursive operations, we transmit the vector $(x_1, \cdots, x_N)$ through N independent B-DMCs, yielding channel output $(y_1, \cdots, y_N)$ at the receiver.

From recursive structure, shown in Fig. 1.3, we know

$$(x_1, \cdots, x_N) = (u_1, \cdots, u_N)\,\mathbf{G}$$

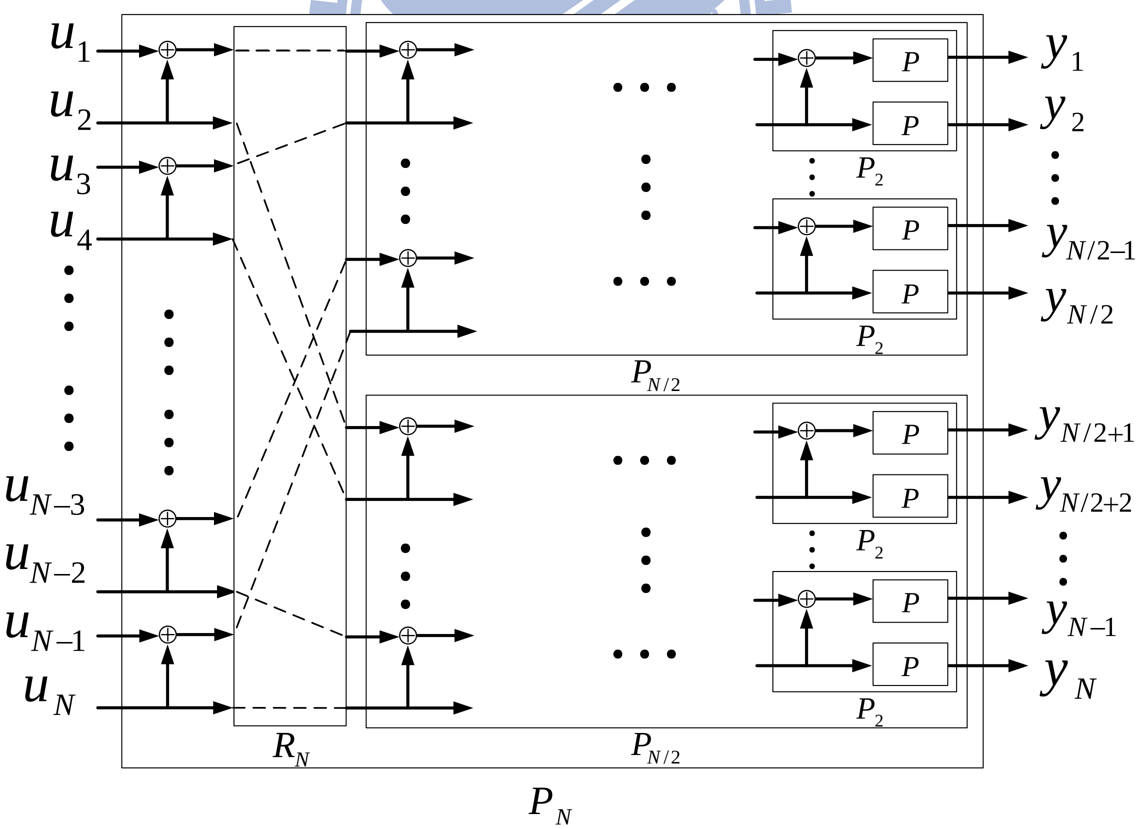


Figure 1.3: Vector channel P_N.

$$\mathbf{G} = \mathbf{B}\,\mathbf{F}^{\otimes n} \text{ and } \mathbf{F} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix},$$

$\mathbf{B}$ is called bit-reversed matrix. For an N-tuple vector, $(t_1, t_2, \cdots, t_N)$, we represent the indices of elements in binary sequence. For example, $N = 8$, $(t_1, t_2, t_3, t_4, t_5, t_6, t_7, t_8)$ $=(t_{000}, t_{001}, t_{010}, t_{011}, t_{100}, t_{101}, t_{110}, t_{111})$. After multiplied by bit-reversed matrix,

$$\begin{aligned}
&(t_1, t_2, t_3, t_4, t_5, t_6, t_7, t_8)\,\mathbf{B} \\
&\quad = (t_{000}, t_{001}, t_{010}, t_{011}, t_{100}, t_{101}, t_{110}, t_{111})\,\mathbf{B} \\
&\quad = (t_{000}, t_{100}, t_{010}, t_{110}, t_{001}, t_{101}, t_{011}, t_{111}) \\
&\quad = (t_1, t_5, t_3, t_7, t_2, t_6, t_4, t_8)
\end{aligned}$$

Given the generator matrix $\mathbf{G}$, the transition probability of vector channel, $P_N : \mathcal{X}^N \to \mathcal{Y}^N$ is

$$\begin{aligned}
&P_N(y_1, \cdots, y_N | u_1, \cdots, u_N) \\
&\quad = P(y_1^N | x_1^N = u_1^N \mathbf{G}) \\
&\quad = P(y_1|x_1) \cdot \cdots \cdot P(y_N|x_N)
\end{aligned}$$

1.1.2 Channel splitting

Having combined N independent B-DMCs into a vector channel, the next step is to split it into N binary-input coordinate channels. The transition probability of each coordinate channel is defined as

$$\begin{aligned}
&P_N^{(i)}(y_1, \cdots, y_N, u_1, \cdots, u_{i-1} | u_i) \\
&\quad = \sum_{u_{i+1}^N \in \mathcal{X}^{N-i}} \frac{P(y_1, \cdots, y_N, u_1, \cdots, u_N)}{P(u_i)} \\
&\quad = \sum_{u_{i+1}^N \in \mathcal{X}^{N-i}} \frac{1}{2^{N-1}} P_N(y_1, \cdots, y_N | u_1, \cdots, u_N) \\
&\quad = \sum_{u_{i+1}^N \in \mathcal{X}^{N-i}} \frac{1}{2^{N-1}} P(y_1|x_1) \cdot P(y_2|x_2) \cdot \cdots \cdot P(y_N|x_N),
\end{aligned}$$

$$x_1^N = u_1^N \mathbf{G},\ 1 \le i \le N,\ N = 2^n,\ n \ge 1,$$

where the $(y_1, \cdots, y_N, u_1, \cdots, u_{i-1})$ is considered as channel output and u_i is the input of the ith coordinate channel. Bhattacharyya parameter of the ith coordinate channel is

$$Z(P_N^{(i)}) = \sum_{y_1^N \in \mathcal{Y}^N} \sum_{u_1^{i-1} \in \mathcal{X}^{i-1}} \sqrt{P_N^{(i)}(y_1^N, u_1^{i-1}|u_i = 0) \cdot P_N^{(i)}(y_1^N, u_1^{i-1}|u_i = 1)}$$

It is shown [2] that the transition probabilities of coordinate channels have the following recursive property.

$$\begin{aligned} &P_N^{(2i-1)}(y_1, \cdots y_N, u_1, \cdots, u_{2i-2}|u_{2i-1}) \\ &\quad = \sum_{u_{2i}} \frac{1}{2} \cdot P_{N/2}^{(i)}(y_1, \cdots, y_{N/2}, u_1 \oplus u_2, \cdots, u_{2i-3} \oplus u_{2i-2}|u_{2i-1} \oplus u_{2i}) \\ &\qquad\qquad \cdot P_{N/2}^{(i)}(y_{N/2+1}, \cdots, y_N, u_2, u_4, \cdots, u_{2i-2}|u_{2i}) \end{aligned} \tag{1.1}$$

and

$$\begin{aligned} &P_N^{(2i)}(y_1, \cdots y_N, u_1, \cdots, u_{2i-1}|u_{2i}) \\ &\quad = \frac{1}{2} \cdot P_{N/2}^{(i)}(y_1, \cdots, y_{N/2}, u_1 \oplus u_2, \cdots, u_{2i-3} \oplus u_{2i-2}|u_{2i-1} \oplus u_{2i}) \\ &\qquad \cdot P_{N/2}^{(i)}(y_{N/2+1}, \cdots, y_N, u_2, u_4, \cdots, u_{2i-2}|u_{2i}) \end{aligned} \tag{1.2}$$

1.2 Polar coding

Now, we consider block codes that contain polar codes as a special case. Let $\mathcal{A}$ be an arbitrary subset of $\{1, 2, \cdots, N\}$. We fix $\mathcal{A}$ and $\underline{u}_{\mathcal{A}^c}$, but let $\underline{U}_{\mathcal{A}}$ be free vector; then we get a block code taking the following form

$$(x_1, \cdots, x_N) = \underline{u}_{\mathcal{A}} \begin{bmatrix} \underline{\mathbf{g}}_{j_1} \\ \vdots \\ \underline{\mathbf{g}}_{j_K} \end{bmatrix} + \underline{u}_{\mathcal{A}^c} \begin{bmatrix} \underline{\mathbf{g}}_{l_1} \\ \vdots \\ \underline{\mathbf{g}}_{l_{N-K}} \end{bmatrix}, \; K = |\mathcal{A}|,$$

$$j_i \in \mathcal{A}, \quad \text{and } 1 \leq i \leq K,$$

$$l_m \in \mathcal{A}^c, \text{and } 1 \leq m \leq N - K.$$

In particular, it is a coset code of a linear block code with generator matrix

$$\begin{bmatrix} \underline{\mathbf{g}}_{j_1} \\ \vdots \\ \underline{\mathbf{g}}_{j_K} \end{bmatrix}, \; K = |\mathcal{A}|, \; j_i \in \mathcal{A}, \text{ and } 1 \leq i \leq K,$$

where the coset leader is

$$\underline{u}_{\mathcal{A}^c} \begin{bmatrix} \underline{\mathbf{g}}_{l_1} \\ \vdots \\ \underline{\mathbf{g}}_{l_{N-K}} \end{bmatrix}, \; K = |\mathcal{A}| \;, \; l_m \in \mathcal{A}^c \;, \text{ and } 1 \leq m \leq N - K,$$

We denote this coset code as $(N, \mathcal{A}, \underline{u}_{\mathcal{A}^c})$. We refer to $\mathcal{A}$ as *information set* and $\underline{u}_{\mathcal{A}^c}$ as the *frozen vector*. The code rate of this coset code is $|\mathcal{A}|/N$. Given the coset code $(N, \mathcal{A}, \underline{u}_{\mathcal{A}^c})$, we encode $(u_1, \cdots, u_N)$ into $(x_1, \cdots, x_N)$, and transmit $(x_1, \cdots, x_N)$ through N copies of B-DMC. Upon receiving channel output $(y_1, \cdots, y_N)$, the task is to decode $(\hat{u}_1, \cdots, \hat{u}_N)$.

Polar codes are specified by the rule that minimizes an upper bound of error probability, and they are channel-specific, i.e., polar codes for two different channels may have different information sets.

1.3 Thesis outline

In Chapter 2, we will introduce the successive cancellation (SC) decoder for the polar codes. It is known that polar codes under SC decoding can achieve channel capacity over symmetric B-DMCs [2]. In Chapter 3, we will introduce the factor-graph representation of polar codes and coordinate channels. Another method has been developed to construct polar codes for symmetric B-DMCs, it will also be introduced in Chapter 3. Knowing that the belief propagation (BP) decoder exhibits good error performance on turbo codes and LDPC codes, it will be used to eliminate some drawbacks of SC decoder and will be shown to have good error performance. In the last section of chapter 3, we will compare the performances of SC and BP decoders at moderate code length. In Chapter 4, we will use the idea of SC decoding to modify the scheduling of BP decoder. Simulation results will be provided.

Chapter 2

Successive Cancellation Decoder

2.1 The decoding algorithm

For the coset code $(N, \mathcal{A}, \underline{u}_{\mathcal{A}^c})$, since knowing $\underline{u}_{\mathcal{A}^c}$ in advance, we set $\hat{\underline{u}}_{\mathcal{A}^c} = \underline{u}_{\mathcal{A}^c}$ before decoding. The output of SC decoder is

$$\hat{u}_i = \begin{cases} h(y_1, \cdots, y_N, \hat{u}_1, \cdots, \hat{u}_{i-1}), & \text{if } i \in \mathcal{A} \\ u_i\,, & \text{if } i \in \mathcal{A}^c \end{cases}$$

where the decoding function for information bits is

$$h(y_1, \cdots, y_N, \hat{u}_1, \cdots, \hat{u}_{i-1}) = \begin{cases} 0\,, & \text{if } \frac{P_N^{(i)}(y_1,\cdots,y_N,\hat{u}_1,\cdots,\hat{u}_{i-1}|u_i=0)}{P_N^{(i)}(y_1,\cdots,y_N,\hat{u}_1,\cdots,\hat{u}_{i-1}|u_i=1)} \geq 1 \\ 1\,, & \text{otherwise} \end{cases} \tag{2.1}$$

Decoding error occurs when $\hat{\underline{u}}_{\mathcal{A}} \neq \underline{u}_{\mathcal{A}}$. The block error probability is denoted by $P_e(N, \mathcal{A}, \underline{u}_{\mathcal{A}^c})$.

2.2 Block error of SC decoding

For a fixed information set $\mathcal{A}$,

$$P(u_1, \cdots, u_N, y_1, \cdots, y_N) = \frac{P_N(y_1, \cdots, y_N|u_1, \cdots, u_N)}{2^N}$$
$$\text{and } (u_1, \cdots, u_N) \in \mathcal{X}^N,\ (y_1, \cdots, y_N) \in \mathcal{Y}^N,$$

Define $\mathcal{E}$ as the error event of SC decoding, i.e.,

$$\mathcal{E} = \{(u_1, \cdots, u_N, y_1, \cdots, y_N)\ : \hat{\underline{U}}_{\mathcal{A}}(y_1, \cdots, y_N, u_1, \cdots, u_N) \neq \underline{u}_{\mathcal{A}}\}$$

$$\hat{U}_i(y_1^N, u_1^N) = \begin{cases} h(y_1^N, \hat{u}_1^{i-1}), & \text{if } i \in \mathcal{A} \\ u_i\,, & \text{if } i \in \mathcal{A}^c \end{cases}$$

Given a fixed $\underline{u}_{\mathcal{A}^c}$, the error probability $\Pr(\mathcal{E}|_{U_{\mathcal{A}^c}=u_{\mathcal{A}^c}}) = P_e(N, \mathcal{A}, \underline{u}_{\mathcal{A}^c})$. We denote by $\mathcal{B}_i$ the first SC decoding error occurs at the ith bit, i.e.,

$$\begin{aligned}
\mathcal{B}_i &= \{(u_1^N, y_1^N)\ : \hat{U}_1(y_1^N) = u_1, \cdots, \hat{U}_{i-1}(y_1^N, \hat{u}_1^{i-2}) = u_{i-1}, \\
&\qquad \hat{U}_i(y_1^N, \hat{u}_1^{i-1}) \neq u_i\} \\
&= \{(u_1^N, y_1^N)\ : \hat{U}_1(y_1^N) = u_1, \cdots, \hat{U}_{i-1}(y_1^N, u_1^{i-2}) = u_{i-1}, \\
&\qquad \hat{U}_i(y_1^N, u_1^{i-1}) \neq u_i\} \\
&\subseteq \{(u_1^N, y_1^N)\ : \hat{U}_i(y_1^N, u_1^{i-1}) \neq u_i\} \\
&= \{(u_1^N, y_1^N)\ : P_N^{(i)}(y_1, \cdots, y_N, u_1, \cdots, u_{i-1}|u_i) \\
&\qquad \leq P_N^{(i)}(y_1, \cdots, y_N, u_1, \cdots, u_{i-1}|u_i \oplus 1)\} \\
&:= \mathcal{E}_i
\end{aligned}$$

Note that $\mathcal{E} = \cup_{i\in\mathcal{A}}\mathcal{B}_i$, and $\{\mathcal{B}_i,\ i \in \mathcal{A}\}$ are disjoint. Then

$$\begin{aligned}
\Pr(\mathcal{E}) &= \sum_{u_{\mathcal{A}^c}\in\mathcal{X}^{N-K}} \frac{1}{2^{N-K}} \cdot P(\mathcal{E}|_{U_{\mathcal{A}^c}=u_{\mathcal{A}^c}}) \\
&= \sum_{u_{\mathcal{A}^c}\in\mathcal{X}^{N-K}} \frac{1}{2^{N-K}} \cdot \sum_{u_{\mathcal{A}}\in\mathcal{X}^K, y_1^N\in\mathcal{Y}^N} \frac{P_N(y_1^N|u_1^N)}{2^K} \cdot \mathbf{1}_{(\cup_{i\in\mathcal{A}}\mathcal{B}_i)} \\
&= \sum_{u_{\mathcal{A}^c}\in\mathcal{X}^{N-K}} \frac{1}{2^{N-K}} \cdot \sum_{u_{\mathcal{A}}\in\mathcal{X}^K, y_1^N\in\mathcal{Y}^N} \frac{P_N(y_1^N|u_1^N)}{2^K} \cdot \sum_{i\in\mathcal{A}} \mathbf{1}_{(\mathcal{B}_i)} \\
&\leq \sum_{(u_1^N, y_1^N)\in\mathcal{X}^N\times\mathcal{Y}^N} \frac{P_N(y_1^N|u_1^N)}{2^N} \cdot \sum_{i\in\mathcal{A}} \mathbf{1}_{(\mathcal{E}_i)} \\
&= \sum_{i\in\mathcal{A}} \sum_{(u_1^N, y_1^N)\in\mathcal{X}^N\times\mathcal{Y}^N} \frac{P_N(y_1^N|u_1^N)}{2^N} \cdot \mathbf{1}_{(\mathcal{E}_i)} = \sum_{i\in\mathcal{A}} \Pr(\mathcal{E}_i) \qquad (2.2) \\
&\leq \sum_{i\in\mathcal{A}} \sum_{(u_1^N, y_1^N)\in\mathcal{X}^N\times\mathcal{Y}^N} \frac{P_N(y_1^N|u_1^N)}{2^N} \cdot \sqrt{\frac{P_N^{(i)}(y_1^N, u_1^{i-1}|u_i \oplus 1)}{p_N^{(i)}(y_1^N, u_1^{i-1}|u_i)}}
\end{aligned}$$

$$= \sum_{i\in\mathcal{A}} \sum_{u_i\in\mathcal{X}} \sum_{(y_1^N, u_1^{i-1})\in\mathcal{Y}^N\times\mathcal{X}^{i-1}} \sum_{u_{i+1}^N\in\mathcal{X}^{N-i}} \frac{P_N(y_1^N|u_1^N)}{2^N} \cdot \sqrt{\frac{P_N^{(i)}(y_1^N, u_1^{i-1}|u_i\oplus 1)}{P_N^{(i)}(y_1^N, u_1^{i-1}|u_i)}}$$

$$= \sum_{i\in\mathcal{A}} \sum_{u_i\in\mathcal{X}} \sum_{(y_1^N, u_1^{i-1})\in\mathcal{Y}^N\times\mathcal{X}^{i-1}} \frac{P_N^{(i)}(y_1^N, u_1^{i-1}|u_i)}{2} \cdot \sqrt{\frac{P_N^{(i)}(y_1^N, u_1^{i-1}|u_i\oplus 1)}{P_N^{(i)}(y_1^N, u_1^{i-1}|u_i)}}$$

$$= \sum_{i\in\mathcal{A}} \sum_{u_i\in\mathcal{X}} \frac{1}{2}\cdot Z(P_N^{(i)}) = \sum_{i\in\mathcal{A}} Z(P_N^{(i)}) \tag{2.3}$$

Following from (2.2) and (2.3), we arrive at the following inequality

$$\Pr(\mathcal{E}) \leq \sum_{i\in\mathcal{A}} \Pr(\mathcal{E}_i) \leq \sum_{i\in\mathcal{A}} Z(P_N^{(i)}).$$

From the above, the notion of *polar code* can be defined.

Definition 2.1. For a given code rate K/N, the coset code $(N, \mathcal{A}, \underline{u}_{\mathcal{A}^c})$ will be called a *polar code* if the subset $\mathcal{A}$ with $|\mathcal{A}| = K$ is chosen such that $\sum_{i\in\mathcal{A}} Z(P_N^{(i)})$ is minimized. □

Remark 2.2. The frozen vector $\underline{u}_{\mathcal{A}^c}$ can be chosen at will, because for additive-noise channel the choice is insensitive to the code performance. For symmetric channels, choosing 0 or 1 for frozen bits would not make any difference on error performance [2]. For simplicity, we choose 0 as frozen bits over this thesis for simulation. □

In [2], the functions $Z(P_N^{(i)})$ are shown to possess the following recursive property

$$\begin{cases} Z(P_{N/2}^{(i)}) \leq Z(P_N^{(2i-1)}) \leq 2Z(P_{N/2}^{(i)}) - Z(P_{N/2}^{(i)})^2 \\ \\ Z(P_N^{(2i)}) = Z(P_{N/2}^{(i)})^2 \end{cases}$$

$$1 \leq i \leq N/2,\ 0 \leq Z(P) \leq 1,$$

In particular, the equality holds for BEC, making the task of determining the information set much easier.

Remark 2.3. For a given B-DMC, Bhattachayya parameter of the channel approximates to 0, while symmetric capacity of the channel is much closed to 1, and vice versa [2]. □

On the other hand, the information theoretic view suggests

$$\begin{aligned} I(U_1^N;Y_1^N) &= \sum_{i=1}^{N} I(U_i\,;Y_1^N|U_1^{i-1}) \\ &= \sum_{i=1}^{N} \left\{I(U_i\,;Y_1^N,U_1^{i-1}) - I(U_i\,;U_1^{i-1})\right\} \\ &= \sum_{i=1}^{N} I(U_i\,;Y_1^N,U_1^{i-1}) \end{aligned} \tag{2.4}$$

where the second equality follows from the independence among $U_1,\cdots,U_N$; hence we get $I(U_i\,;U_1^{i-1})=0$. Furthermore, we remark that $I(U_i\,;Y_1^N,U_1^{i-1})$ is mutual information of the ith coordinate channel. On the other hand, note

$$I(U_1^N;Y_1^N) = I(X_1^N;Y_1^N) = NI(X;Y) \tag{2.5}$$

where the first equality is due to that the generator matrix $\mathbf{G}$ is one-to-one and onto mapping.

Since $U_1,\cdots,U_N$ are i.i.d. and uniform distributed, $I(U_i\,;Y_1^N,U_1^{i-1})$ and $I(X;Y)$ are symmetric capacity of the ith coordinate channel and transmitting channel respectively.

Lemma 2.4 ([2]). *As $N\to\infty$, $I(U_i\,;Y_1^N,U_1^{i-1})$ converge to 0 or 1 almost surely.*

From (2.4) and (2.5), we can get $\sum_{i=1}^{N} I(U_i\,;Y_1^N,U_1^{i-1})/N = I(X;Y)$. It follows that the fraction of noiseless coordinate channels is symmetric capacity of transmitting channel as $N\to\infty$. Furthermore, as $N\to\infty$, fraction of coordinate channels whose Bhattacharyya parameters approach 0 is symmetric capacity of transmitting channel. To summarize, the above altogether shows that we can make code rate equals symmetric capacity.

As $N\to\infty$, we can encode information bits with $R\le$*symmetric capacity*, with SC decoding error tending to 0. We know that symmetric capacity is channel capacity for symmetric B-DMCs. So, polar codes can achieve the capacity for symmetric B-DMCs [2].

2.3 Recursive property of SC decoding

In this section we provide the recursive property for the log-likelihood ratios (LLRs) that will be used for the decoding function h (c.f. (2.8)) of SC decoding for polar codes.

From (1.1) and (1.2), we get

$$
\begin{aligned}
& L_N^{(2i-1)}(y_1, \cdots, y_N, \hat{u}_1, \cdots, \hat{u}_{2i-2}) \\
& \quad = \ln \frac{P_N^{(2i-1)}(y_1, \cdots, y_N, \hat{u}_1, \cdots, \hat{u}_{2i-2} | u_{2i-1} = 0)}{P_N^{(2i-1)}(y_1, \cdots, y_N, \hat{u}_1, \cdots, \hat{u}_{2i-2} | u_{2i-1} = 1)} \\
& \quad = 2 \tanh^{-1}\{\tanh[L_{N/2}^{(i)}(y_1, \cdots, y_{N/2}, \hat{u}_1 \oplus \hat{u}_2, \cdots, \hat{u}_{2i-3} \oplus \hat{u}_{2i-2})/2] \\
& \qquad\qquad \times \tanh[L_{N/2}^{(i)}(y_{N/2+1}, \cdots, y_N, \hat{u}_2, \hat{u}_4, \cdots, \hat{u}_{2i-2})/2]\} \qquad (2.6)
\end{aligned}
$$

and

$$
\begin{aligned}
& L_N^{(2i)}(y_1, \cdots, y_N, \hat{u}_1, \cdots, \hat{u}_{2i-1}) \\
& \quad = L_{N/2}^{(i)}(y_{N/2+1}, \cdots, y_N, \hat{u}_2, \hat{u}_4, \cdots, \hat{u}_{2i-2}) \\
& \qquad + (-1)^{\hat{u}_{2i-1}} \cdot L_{N/2}^{(i)}(y_1, \cdots, y_{N/2}, \hat{u}_1 \oplus \hat{u}_2, \cdots, \hat{u}_{2i-3} \oplus \hat{u}_{2i-2}) \qquad (2.7)
\end{aligned}
$$

Each value in the pair

$$\left(L_N^{(2i-1)}(y_1, \cdots, y_N, \hat{u}_1, \cdots, \hat{u}_{2i-2}), L_N^{(2i)}(y_1, \cdots, y_N, \hat{u}_1, \cdots, \hat{u}_{2i-1})\right)$$

can be assembled from the same pair

$$\left(L_{N/2}^{(i)}(y_1^{N/2}, \hat{u}_1 \oplus \hat{u}_2, \cdots, \hat{u}_{2i-3} \oplus \hat{u}_{2i-2}), L_{N/2}^{(i)}(y_{N/2+1}^{N}, \hat{u}_2, \hat{u}_4, \cdots, \hat{u}_{2i-2})\right).$$

The N values $L_N^{(i)}$, $1 \leq i \leq N$, can be calculated from N values $L_{N/2}^{(j)}$, $1 \leq j \leq N/2$. In the following, if (a, b) is assembled from (c, d), we denote it by $(a, b) \longleftrightarrow (c, d)$. This, for example in the case of N=4,

$$
\begin{aligned}
\left(L_4^{(1)}(y_1^4), L_4^{(2)}(y_1^4, \hat{u}_1)\right) &\longleftrightarrow \left(L_2^{(1)}(y_1, y_2), L_2^{(1)}(y_3, y_4)\right) \\
\left(L_4^{(3)}(y_1^4, \hat{u}_1, \hat{u}_2), L_4^{(4)}(y_1^4, \hat{u}_1, \hat{u}_2, \hat{u}_3)\right) &\longleftrightarrow \left(L_2^{(2)}(y_1, y_2, \hat{u}_1 \oplus \hat{u}_2), L_2^{(2)}(y_3, y_4, \hat{u}_2)\right)
\end{aligned}
$$

$$
\begin{aligned}
\left(L_2^{(1)}(y_1, y_2), L_2^{(2)}(y_1, y_2, \hat{u}_1 \oplus \hat{u}_2\right) &\longleftrightarrow \left(L_1^{(1)}(y_1), L_1^{(1)}(y_2)\right) \\
\left(L_2^{(1)}(y_3, y_4), L_2^{(2)}(y_3, y_4, \hat{u}_2)\right) &\longleftrightarrow \left(L_1^{(1)}(y_3), L_1^{(1)}(y_4)\right)
\end{aligned}
$$

The above relation is shown in Fig. 2.1.

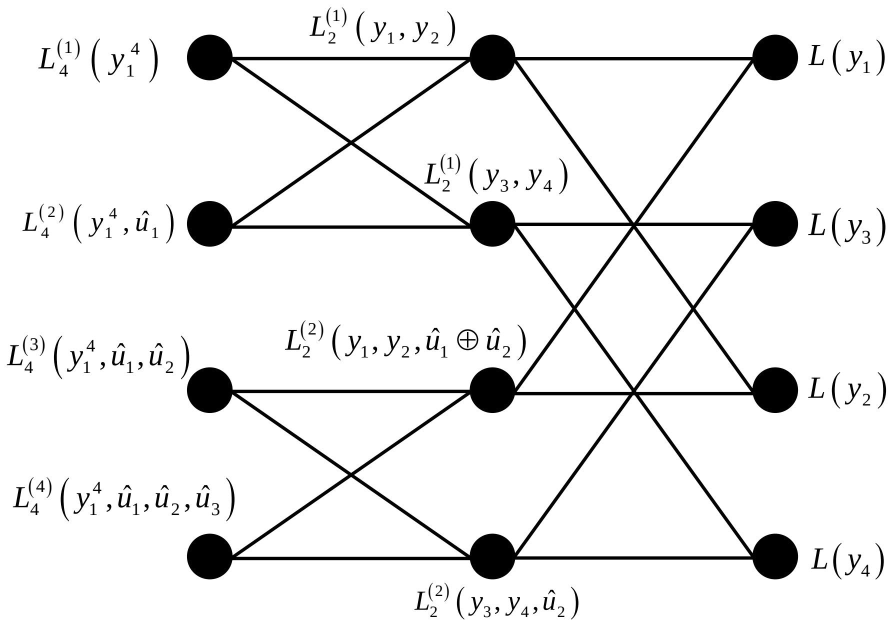


Figure 2.1: Recursive property of SC decoding function for N=4. The left pair in the butterfly pattern can be assembled from the right pair.

The decoding function can be rewritten as

$$h(y_1, \cdots, y_N, \hat{u}_1, \cdots, \hat{u}_{i-1}) = \begin{cases} 0, & \text{if } L_N^{(i)}(y_1, \cdots, y_N, \hat{u}_1, \cdots, \hat{u}_{i-1}) \geq 0 \\ 1, & \text{otherwise} \end{cases} \tag{2.8}$$

Thus, knowing LLR values of every channel output, we can successively decode information bits by calculating N LLR values, $L_N^{(i)}$, which are functions of channel LLR values and previously decoded bits.

Chapter 3

Belief Propagation Decoder

3.1 Factor graph

In [3], Forney showed normalized factor graph of RM codes. Since polar codes are sub-codes of full RM(n, n) codes, we know polar codes can also be represented by the same factor graph, rendering the BP decoder to the decoding of polar codes. There are $n = \log_2 N$ sections in the factor graph of polar codes, and each section consists of $N/2$ Z-shaped sub-graphs. Fig. 3.1 is the factor graph of polar codes for $N = 8$. In addition to BP decoding, we remark that the SC decoding can also be illustrated by factor graph.

From (1.1) and (1.2), we can get factor graph of the ith coordinate channel from factor graph of polar codes [4]. The factor graph of the ith coordinate channel can be obtained through the following steps.

1. In the left-most section, eliminate the degree-3 check nodes that are not connected to the ith variables nodes and the edges incident to the check nodes.
2. Eliminate the rest degree-0 and 1 variable nodes except the ith variables nodes, the edges incident to these eliminated variable nodes should also be eliminated.
3. Eliminate the rest degree-1 check nodes and the incident edges.
4. In the second left-most section, eliminate the degree-3 check nodes that are not connected to the left-most ith variable node via left path.
5. Do the above steps iteratively until elimination in the right-most section is finished.

Then, we can get the factor graphs of coordinate channels of all N bits, u_1^N. They all are trees. Fig. 3.2 is the factor graph of the 4th coordinate channel for $N = 8$.

3.2 Codes construction on symmetric B-DMCs

Let $a_N^{(i)}$ denotes the probability density function (pdf) of $L_N^{(i)}(y_1^N, u_1^{i-1})$ conditioned on $u_i = 0$. In [4], it's shown that for the symmetric B-DMCs,

$$\Pr(\mathcal{E}_i) = \frac{1}{2}\int_{-\infty}^{\infty} a_N^{(i)}(x) e^{-\left(\frac{|x|}{2}+\frac{x}{2}\right)} dx, \quad 1 \le i \le N \tag{3.1}$$

For symmetric B-DMCs, the Bhattacharyya parameter of the channel is

$$Z = \int_{-\infty}^{\infty} a(x) e^{-\frac{x}{2}} dx$$

$a(x)$ is pdf of LLR of the symmetric B-DMC output conditioned on input bit is zero [5].

For symmetric B-DMCs, we know that Bhattacharyya parameters of all N coordinate channels are

$$Z(P_N^{(i)}) = \int_{-\infty}^{\infty} a_N^{(i)}(x) e^{-\frac{x}{2}} dx, \quad 1 \le i \le N$$

We can choose $\mathcal{A}$ to minimize $\sum_{i\in\mathcal{A}} \Pr(\mathcal{E}_i)$ or $\sum_{i\in\mathcal{A}} Z(P_N^{(i)})$.

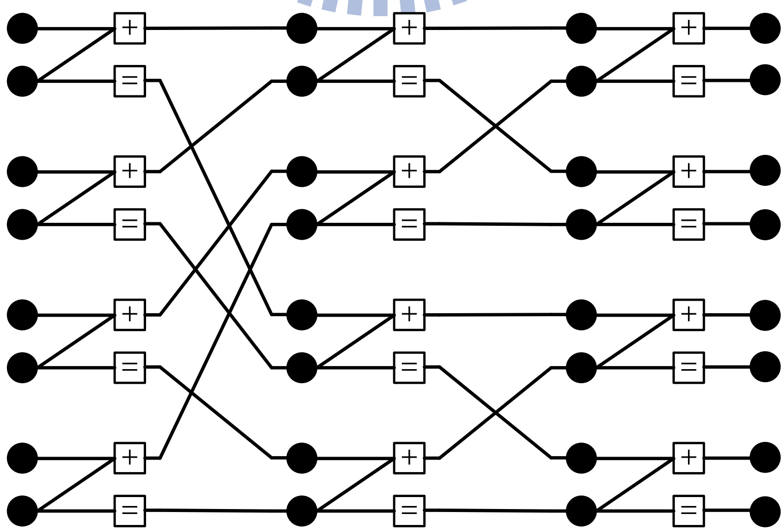


Figure 3.1: Factor graph of polar codes for N=8.

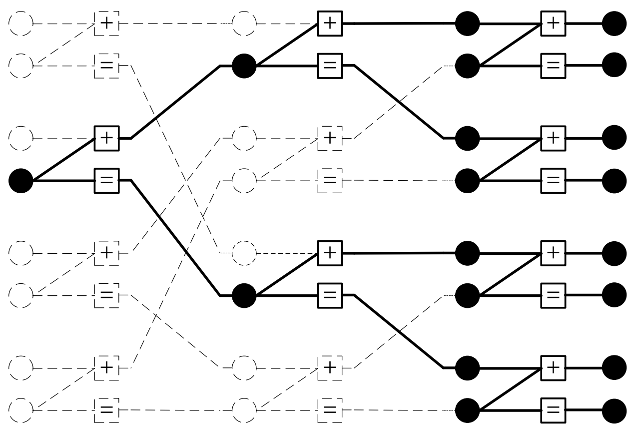


Figure 3.2: Factor graph of the 4th coordinate channel for N=8, $P_8^{(4)}$, dashed lines and nodes are eliminated edges and nodes.

3.3 SC decoding in factor graph

The factor graphs of coordinate channels, introduced in the former section, can be used to illustrate SC decoding. Since $u_{i+1}, \cdots, u_N$ are not characteristic of the ith coordinate channel $P_N^{(i)}$, and $\hat{u}_1, \cdots, \hat{u}_{i-1}$ have been decoded, so these $N-1$ variable nodes in the factor graph of polar codes can be eliminated. We do the same eliminating procedure like the former section to get the factor graph of the ith coordinate channel for SC decoding. On the leaf nodes, the channel LLR values are effected by the former decoded bits. It's illustrated below and in Fig. 3.3 for $N = 4$ and $i = 3$.

The first leaf node:

$$\text{Since } x_1 = u_1 \oplus u_2 \oplus u_3 \oplus u_4, \text{ and } P(y_1|x_1) = P(y_1|u_1 \oplus u_2 \oplus u_3 \oplus u_4)$$

$$\begin{aligned}\text{Input} &= \ln \frac{P(y_1|\hat{u}_1 \oplus \hat{u}_2 \oplus u_3 \oplus u_4 = 0)}{P(y_1|\hat{u}_1 \oplus \hat{u}_2 \oplus u_3 \oplus u_4 = 1)} \\ &= \ln \frac{P(y_1|u_3 \oplus u_4 = 0 \oplus \hat{u}_1 \oplus \hat{u}_2)}{P(y_1|u_3 \oplus u_4 = 1 \oplus \hat{u}_1 \oplus \hat{u}_2)} \\ &= L(y_1)[-2(\hat{u}_1 \oplus \hat{u}_2) + 1]\end{aligned}$$

The second leaf node:

$$\text{Since } x_2 = u_3 \oplus u_4, \text{ and } P(y_2|x_2) = P(y_2|u_3 \oplus u_4)$$

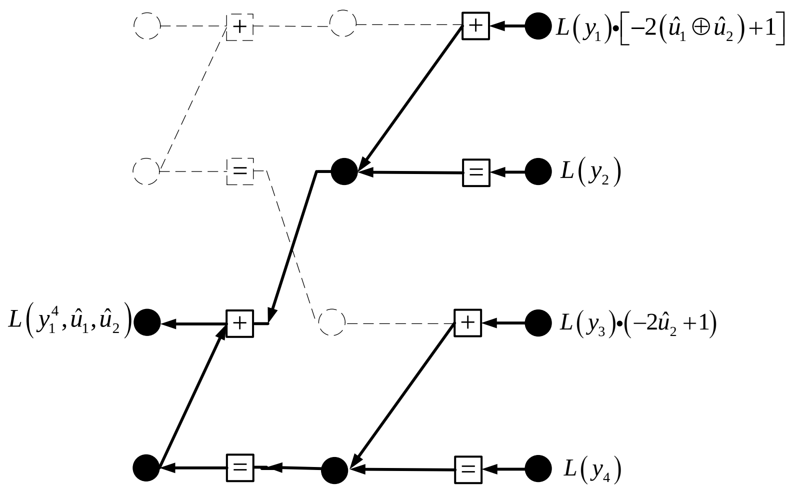


Figure 3.3: SC decoding diagram using factor graph, arrows represent message-passing directions in decoding process.

$$\text{Input} = \ln \frac{P(y_2|u_3 \oplus u_4 = 0)}{P(y_2|u_3 \oplus u_4 = 1)} = L(y_2)$$

The third leaf node:

$$\text{Since } x_3 = u_2 \oplus u_4, \text{ and } P(y_3|x_3) = P(y_3|u_2 \oplus u_4)$$

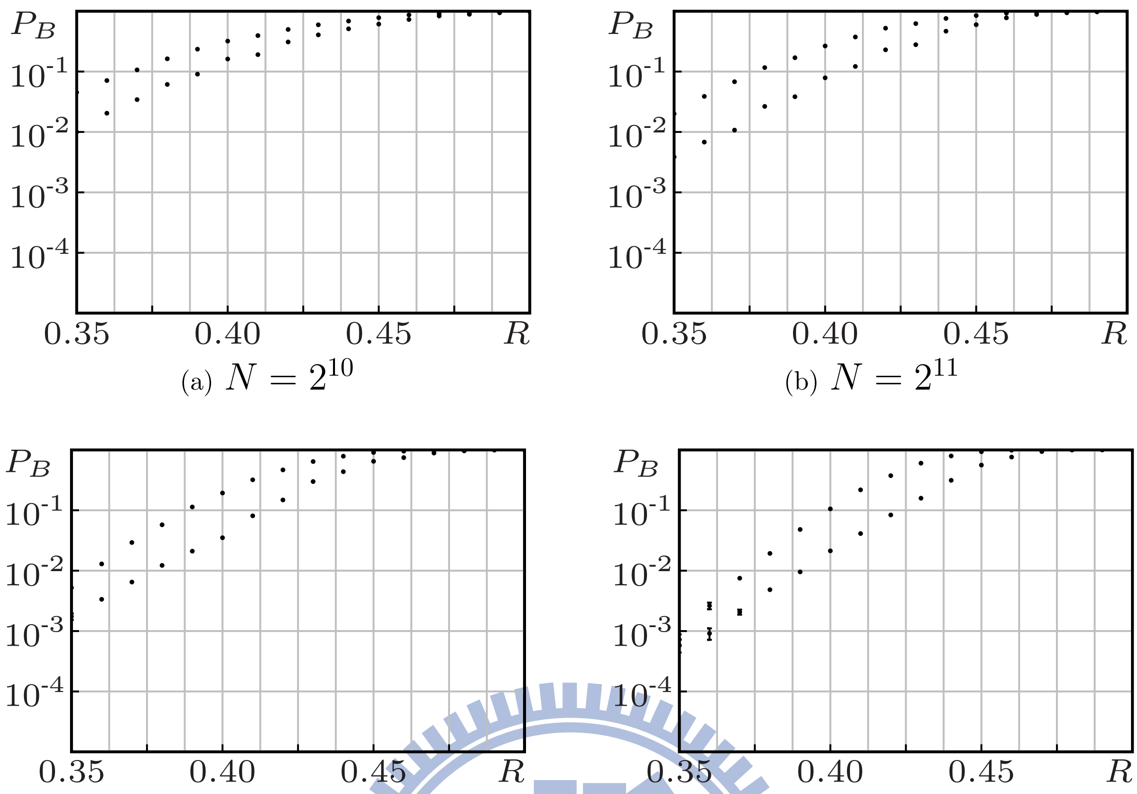
$$\text{Input} = \ln \frac{P(y_3|\hat{u}_2 \oplus u_4 = 0)}{P(y_3|\hat{u}_2 \oplus u_4 = 1)} = \ln \frac{P(y_3|u_4 = 0 \oplus \hat{u}_2)}{P(y_3|u_4 = 1 \oplus \hat{u}_2)} = L(y_3)(-2\hat{u}_2 + 1)$$

The forth leaf node:

$$\text{Since } x_4 = u_4, \text{ and } P(y_4|x_4) = P(y_4|u_4)$$

$$\text{Input} = \ln \frac{P(y_4|u_4 = 0)}{P(y_4|u_4 = 1)} = L(y_4)$$

Under SC decoding, we decode information bits successively, and use former $i-1$ decoded bits, i.e., $\hat{u}_1, \cdots, \hat{u}_{i-1}$, as information for decoding u_i. Since decoding errors maybe occur while decoding the former information bits, the errors will propagate. In order to get decoding convenience, i.e., recursive formulae (2.6) and (2.7), we assume all the frozen bits are free variables. The assumption violates the actual message probability distribution, so decoding errors will happen more frequently.



(a) $N = 2^{10}$ (b) $N = 2^{11}$ (c) $N = 2^{12}$ (d) $N = 2^{13}$

Figure 3.4: Comparison of SC decoding (upper curve) and BP decoding (lower curve) in terms of word error rate, when transmission takes place over binary-input AWGN channel (σ=0.97865) [1].

3.4 Belief propagation decoding

Polar codes with SC decoding are asymptotically capacity achieving, but not record breaking for moderate length. Under BP decoding, it uses information provided by frozen bits, and without using the hard-decision bits decoded previously, it can avoid error propagation. We can conclude BP decoding is better than SC decoding.

BP decoding was used to compare performance of polar codes constructed on Bhattacharyya parameters and RM codes in [6]. Performance of polar codes under SC decoding and BP decoding was already shown in [1], we can see that BP decoding is obviously better than SC decoding with various code rates and block lengths.

Chapter 4

Scheduling of BP Decoding

Although BP decoding is better than SC decoding for polar codes, there still exists a gap between BP decoding and MAP decoding for the error rate curve [1]. Some methods have been proposed to improve error performance over BP decoding [1][7].

4.1 Incremental updating schedule

From Section 2.2, we know that if $\hat{u}_1, \cdots, \hat{u}_{i-1}$ are correctly decoded, then u_i almost can be decoded correctly for long block length. But it's not easy to correctly decode information bit successively for moderate code length. Since SC decoding can be regarded as a case of BP decoding, the successive decoding notion can be used to modify BP decoding schedule. The proposed decoding schedule is called *incremental updating schedule*.

We partition the N bits including frozen bits and information bits into many equal-sized blocks according to index-order. Each block has a factor graph that consists of factor graphs associated with coordinate channels described in Section 3.1. For example, for $N = 8$ we partition N bits into 4 blocks. The first block is (u_1, u_2), second is (u_3, u_4) and so on. Fig. 4.1 shows factor graph of the first block, and factor graph for the second one is shown in Fig. 4.2.

For code length is N, block size is 2^l and $l \geq 1$, there are $N/2^l$ iteration times in incremental updating decoding schedule. The steps of decoding process are showing below, some figures are also shown for illustration:

1. We first set the initial condition using information provided by information bits and frozen bits, and calculate LLR of every path in the factor graph of polar codes from

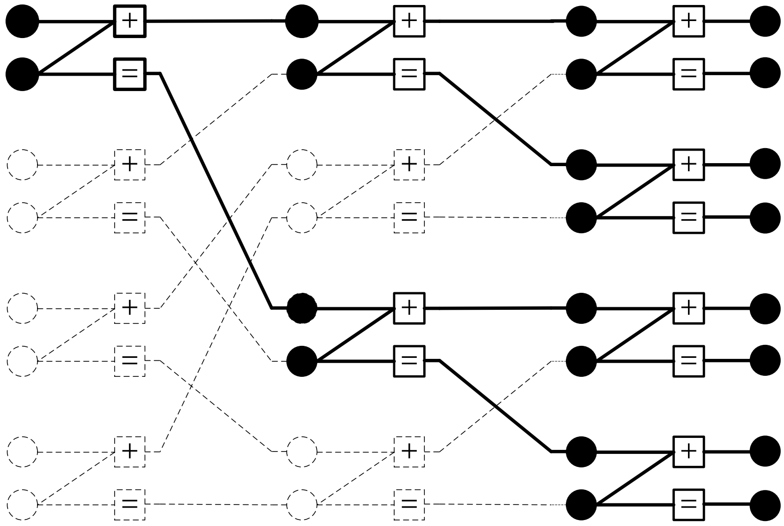


Figure 4.1: Factor graph of the first block for N=8.

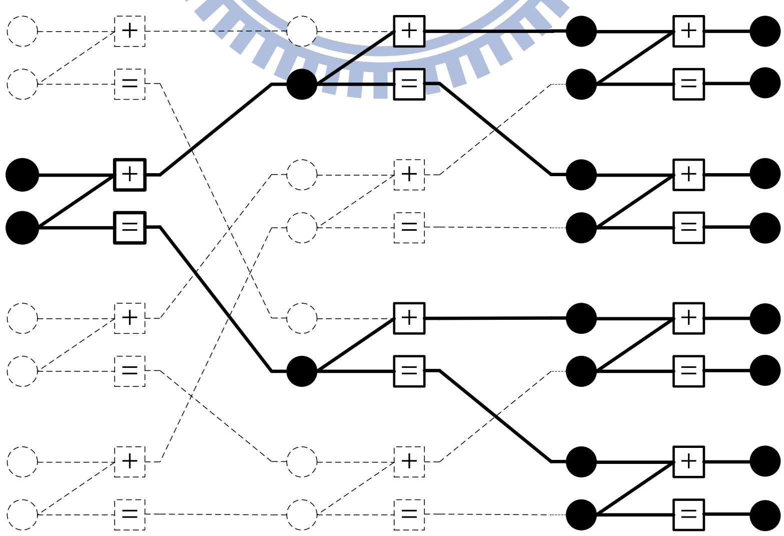


Figure 4.2: Factor graph of the second block for N=8.

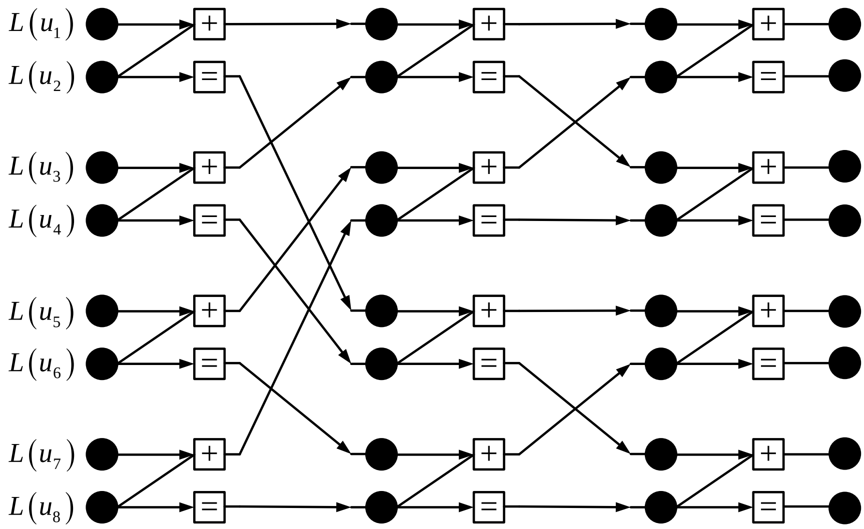


Figure 4.3: Calculate initial LLR values of polar codes factor graph.

left-most level to right-most level. It's shown in Fig. 4.3.

2. At the first iteration, we consider the factor graph of the first block, and calculate LLR of every path in the factor graph from right-most level to left-most level. It's shown in Fig. 4.4.

3. Update the LLR of every path in the factor graph from left-most level to the right-most level as shown in Fig. 4.5.

4. Consider the factor graph union of block 1 and block 2, revise message passing through every path of the factor graph from right-most level to the left-most level and update back. Updating from right side to left side is shown in Fig. 4.6.

5. During the ith iteration, consider the first i blocks and the associated factor graph, repeat the updating procedure described above.

6. At the last iteration, consider all block, and the associated factor graph is the factor graph of polar codes. We only update LLR from right to left, and add the information provided by frozen bits.

7. Quantize the LLR values associated with every information bit and frozen bit. Finally, decide whether they are 0 or 1.

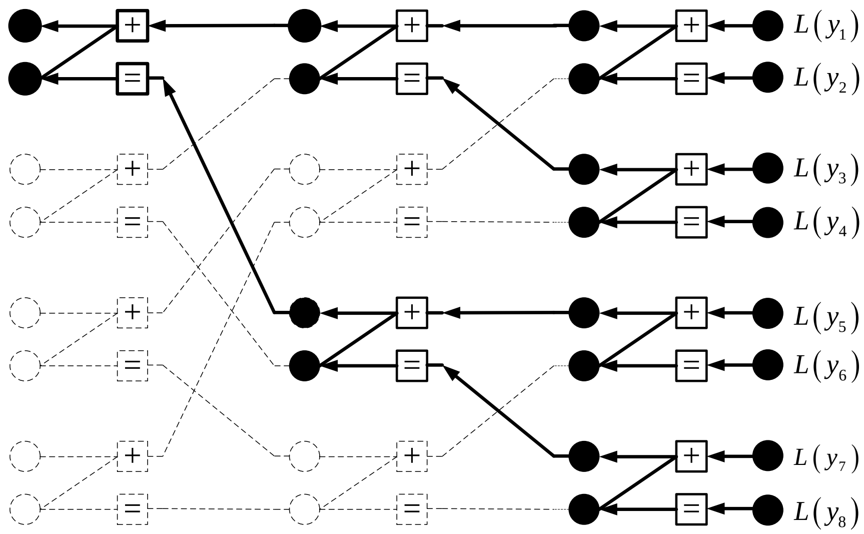


Figure 4.4: Update LLR from right to left in iteration 1.

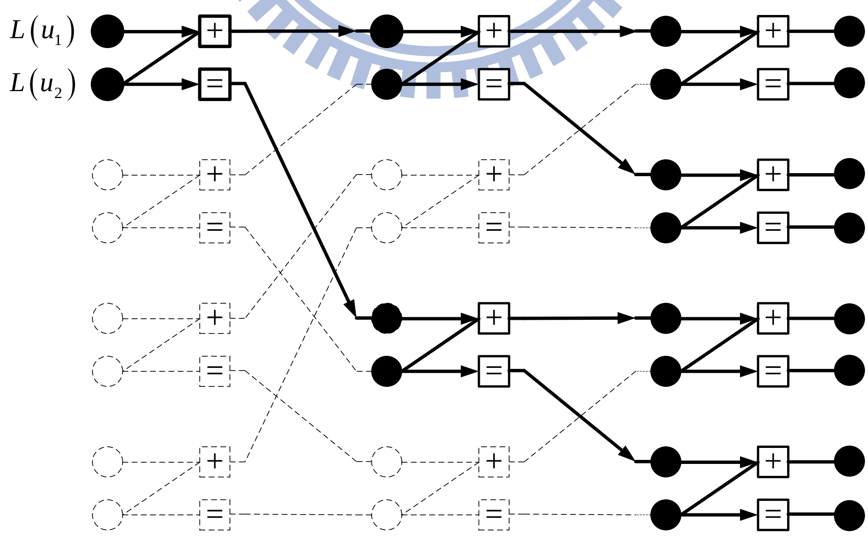


Figure 4.5: Update LLR from left to right in Iteration 1.

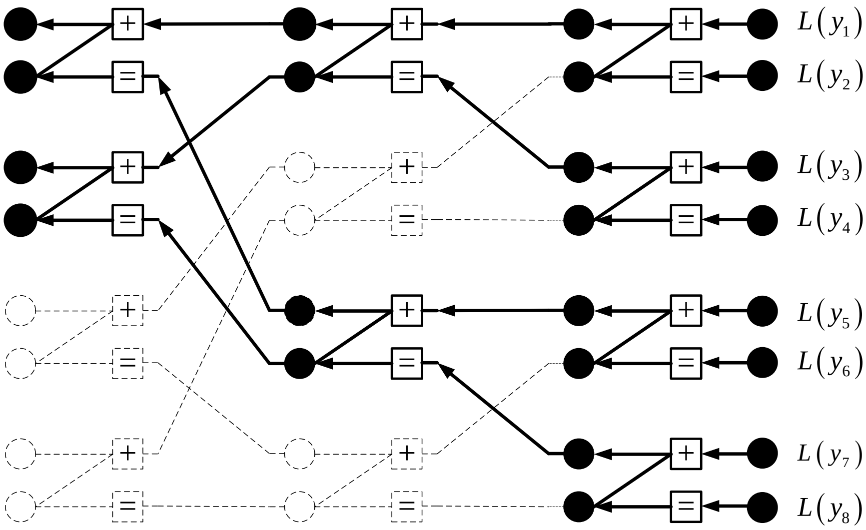


Figure 4.6: Update LLR from right to left in iteration 2.

4.2 Simulation results

In this thesis, we simulate polar codes with code rates 1/3, 1/2 or 5/6 and code length ranging form 2^{11} to 2^{13} over binary-input AWGN channel.

The task of finding $\mathcal{A}$ is according to (2.2) and (3.1), i.e.,

$$\Pr(\mathcal{E}) \le \sum_{i\in\mathcal{A}} \Pr(\mathcal{E}_i)$$

$$\Pr(\mathcal{E}_i) = \frac{1}{2}\int_{-\infty}^{\infty} a_N^{(i)}(x)e^{-\left(\frac{|x|}{2}+\frac{x}{2}\right)}dx, \quad 1 \le i \le N$$

Specifically, we simulate 10000 trials to find $\mathcal{A}$. In each trial, we first calculate $L_N^{(i)}(y_1, \cdots, y_N, u_1, \cdots, u_{i-1})$ while the sent bit $u_i = 0$. After the 10000 trials, we find the expected values

$$\frac{1}{2}e^{-\left(\frac{|L_N^{(i)}|}{2}+\frac{L_N^{(i)}}{2}\right)}, \text{conditioned on } u_i = 0, \text{ for } 1 \le i \le N$$

We note that the above expected values would converge to $\Pr(\mathcal{E}_i)$ as the number of trials increases. After obtaining the value associated with each i, for any specific code rate, we choose the subset of $\{1, \cdots, N\}$ so as to minimize $\sum_{i\in\mathcal{A}} \Pr(\mathcal{E}_i)$. Having fixed the code rate and code length, there are two parts in the following simulations.

The first part: For a specific rate and code length, we simulate incremental updating schedule with three partition types, i.e., the decoded bits in the three types are partitioned into 2^8, 2^9 or 2^{10} blocks respectively, and show the results of increasing number of blocks in incremental updating schedule.

The second part: We partition decoded bits into 2^{10} in incremental updating schedule. The iteration times of BP decoding without scheduling is set to 640 so that the two decoding algorithms almost have the same complexity, i.e., the average iteration times of degree-3 nodes are the same. Besides error performance comparison of SC decoding, BP decoding without scheduling and BP decoding in incremental updating schedule, we also show the results of increasing iteration times under BP decoding without scheduling.

In both simulations we observe that the scheduling based on incremental updating performs better than the one without.

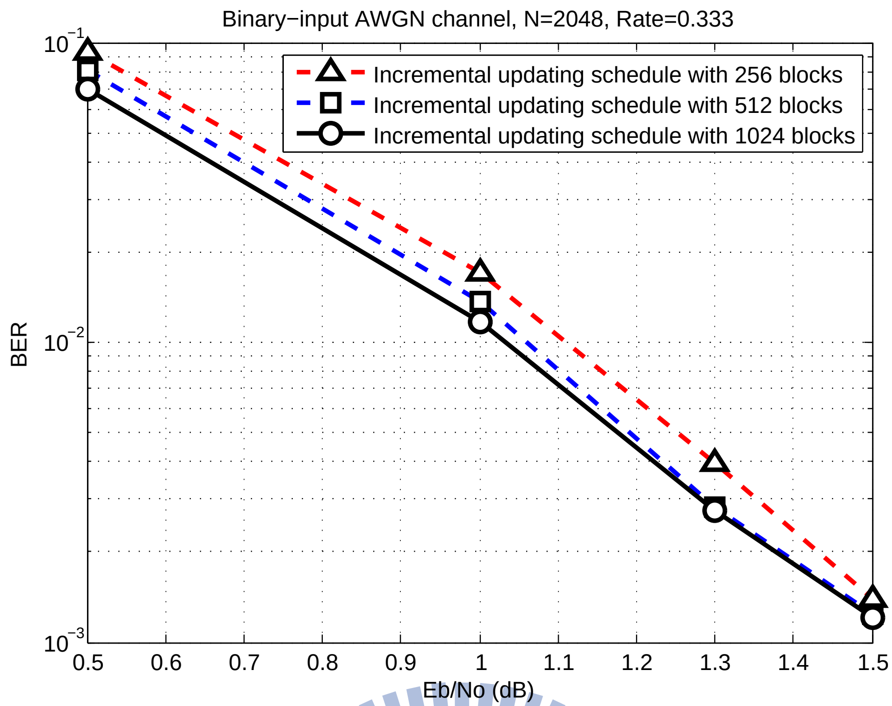


Figure 4.7: Error performance comparison of increasing block number in incremental updating schedule. Code rate is 1/3 and block length is 2048.

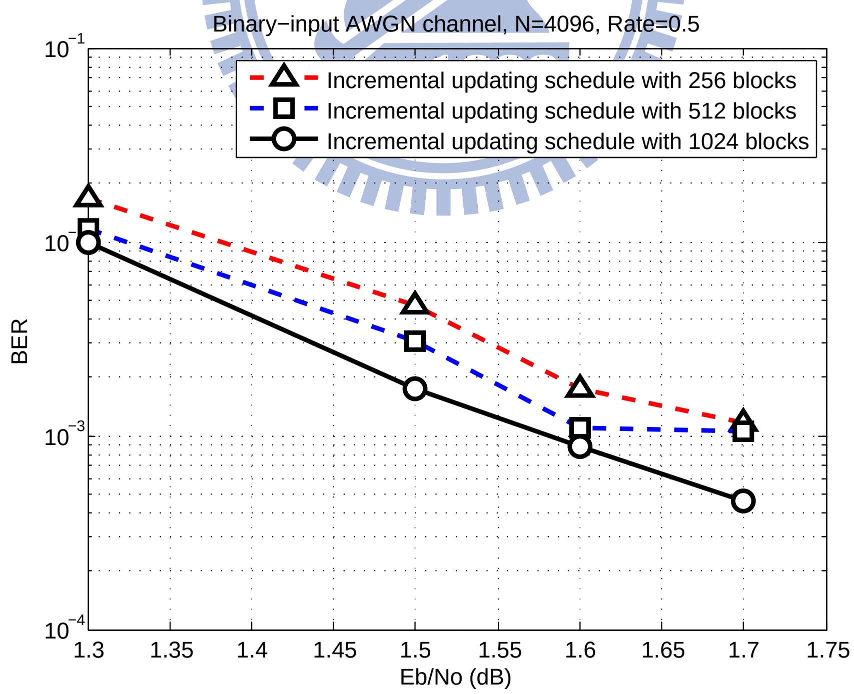


Figure 4.8: Error performance comparison of increasing block number in incremental updating schedule. Code rate is 1/2 and block length is 4096.

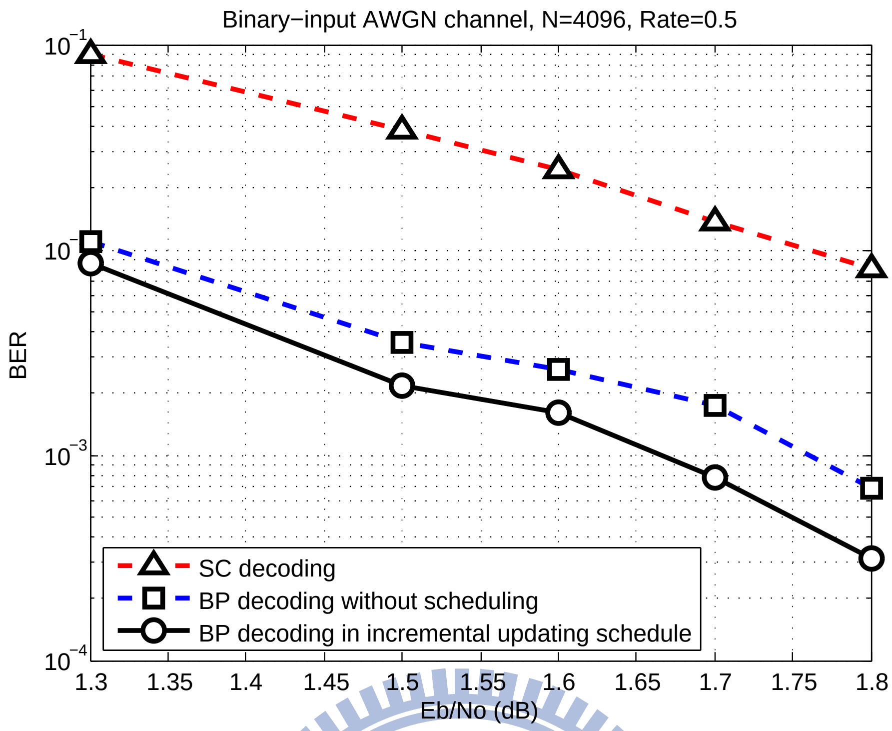


Figure 4.9: Error performance comparison for code rate is 1/2 and block length is 4096.

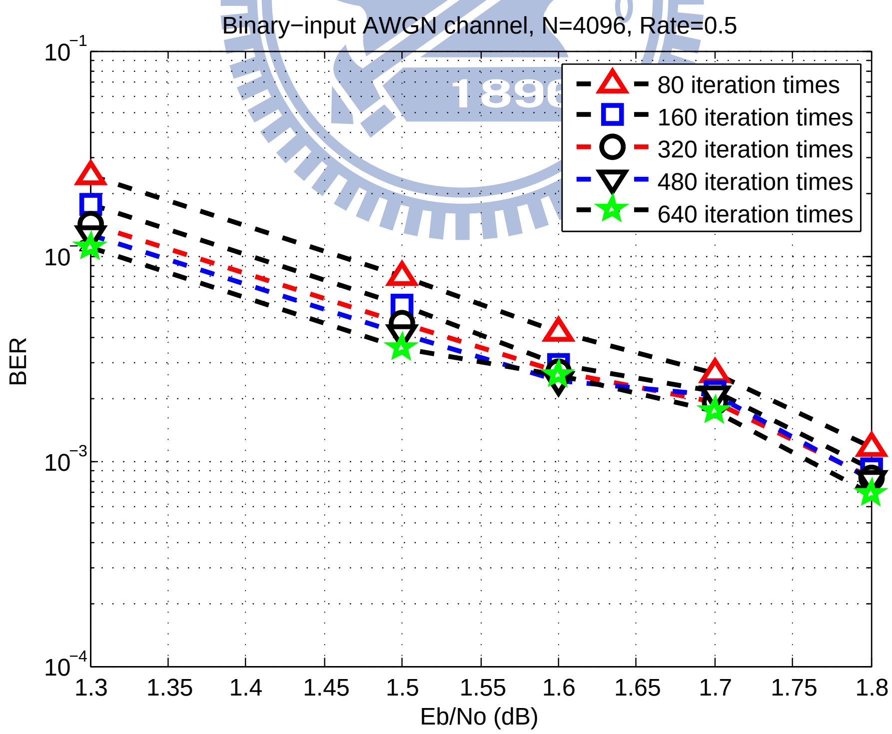


Figure 4.10: Error performance comparison of increasing iteration times under BP decoding for code rate is 1/2 and block length is 4096.

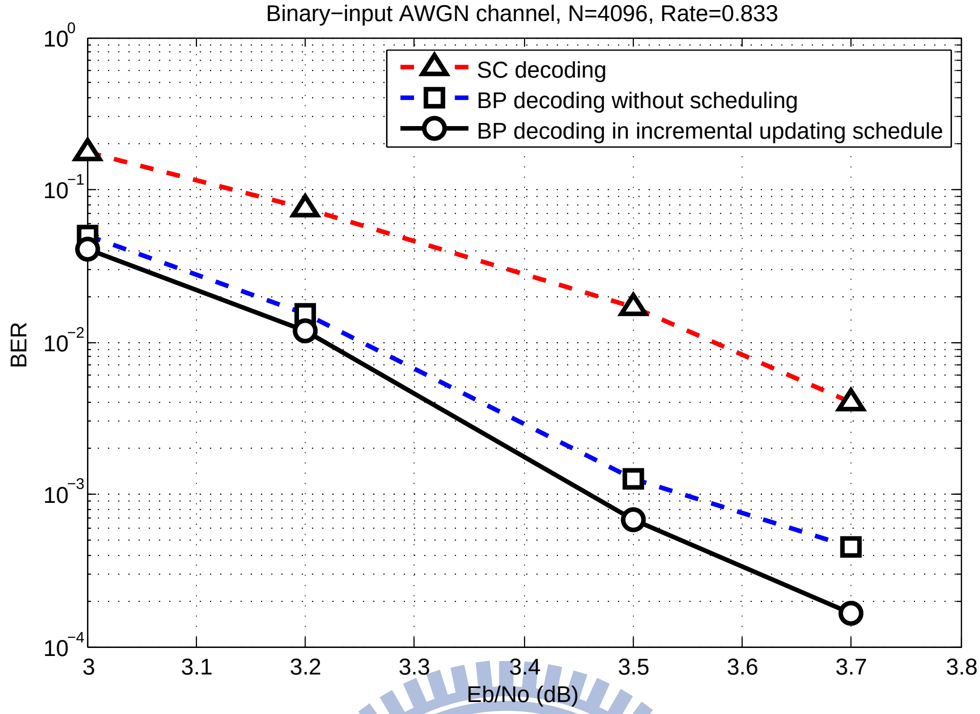


Figure 4.11: Error performance comparison for code rate is 5/6 and block length is 4096.

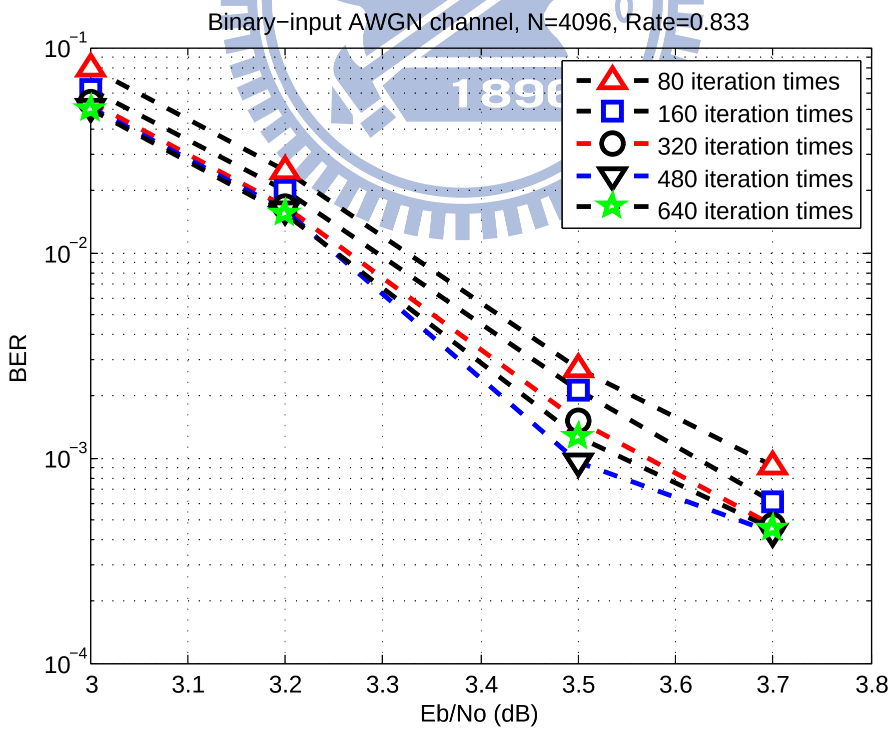


Figure 4.12: Error performance comparison of increasing iteration times under BP decoding for code rate is 5/6 and block length is 4096.

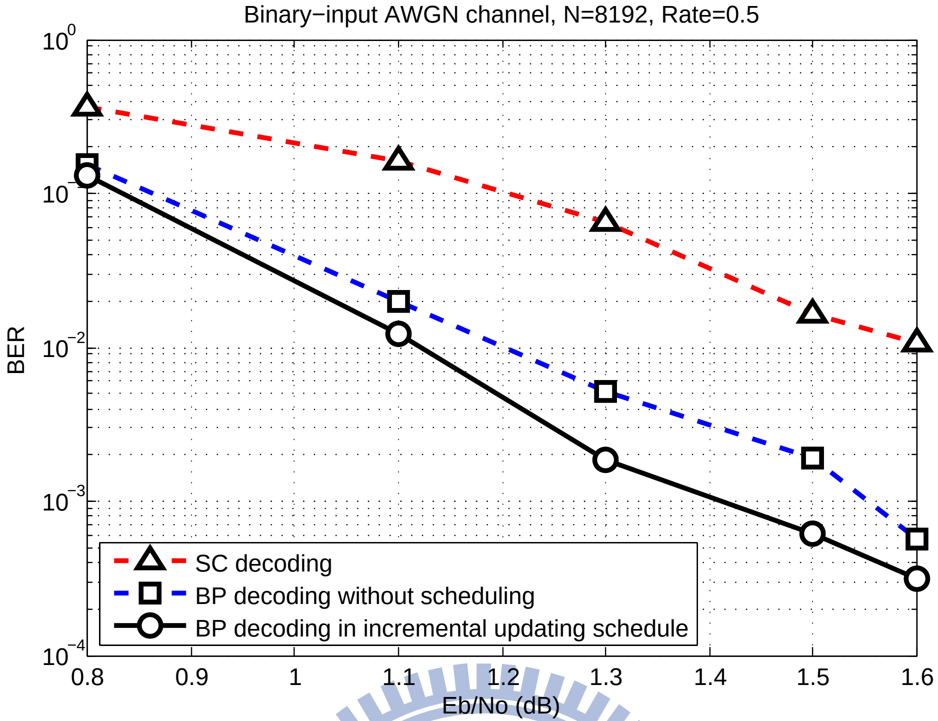


Figure 4.13: Error performance comparison for code rate is 1/2 and block length is 8192.

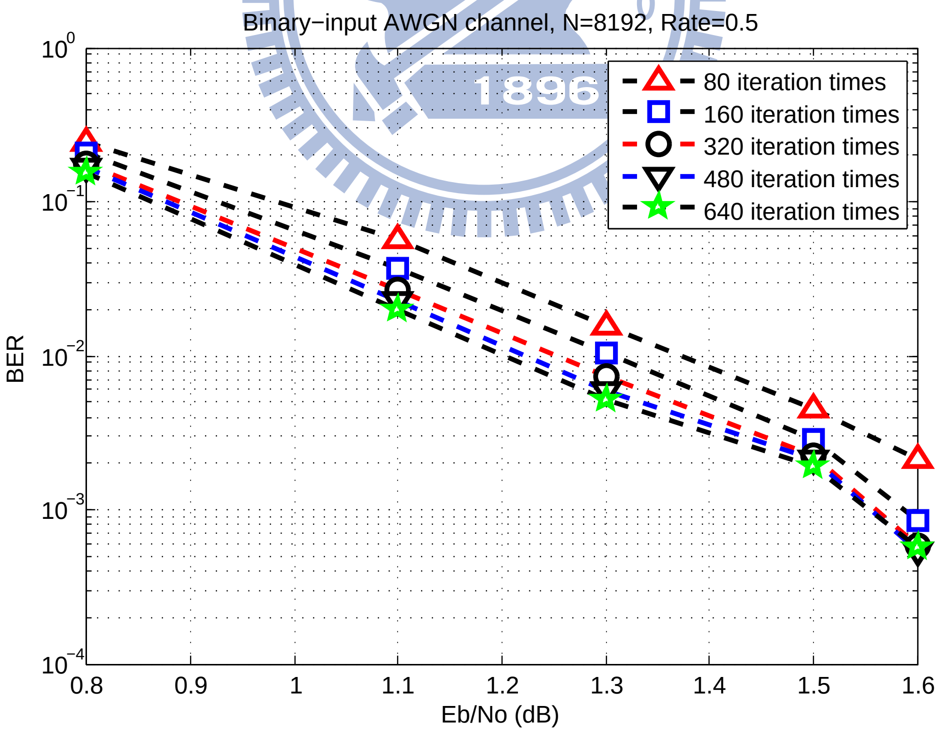


Figure 4.14: Error performance comparison of increasing iteration times under BP decoding for code rate is 1/2 and block length is 8192.

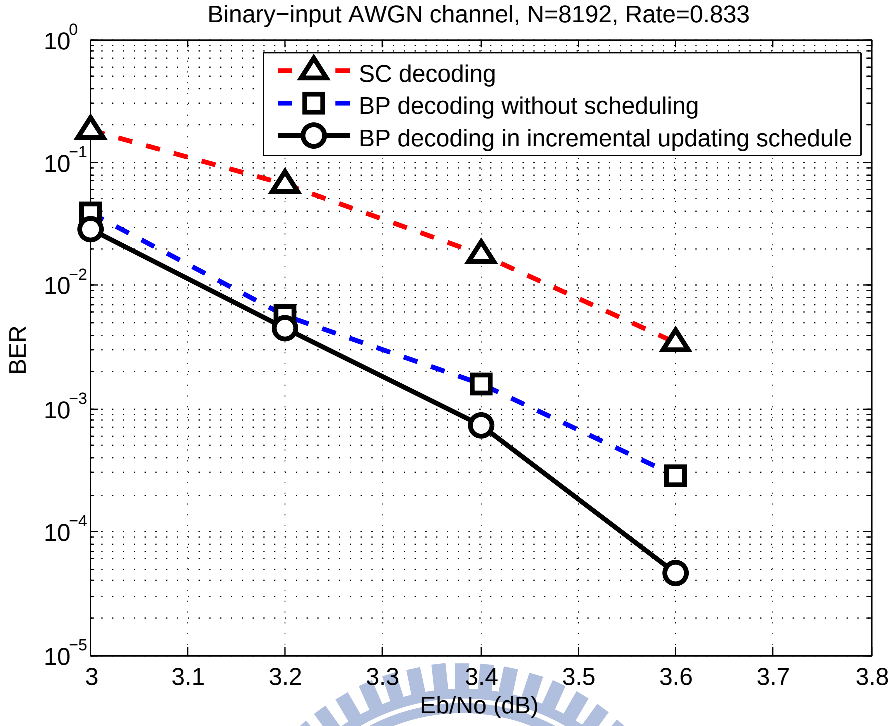


Figure 4.15: Error performance comparison for code rate is 5/6 and block length is 8192.

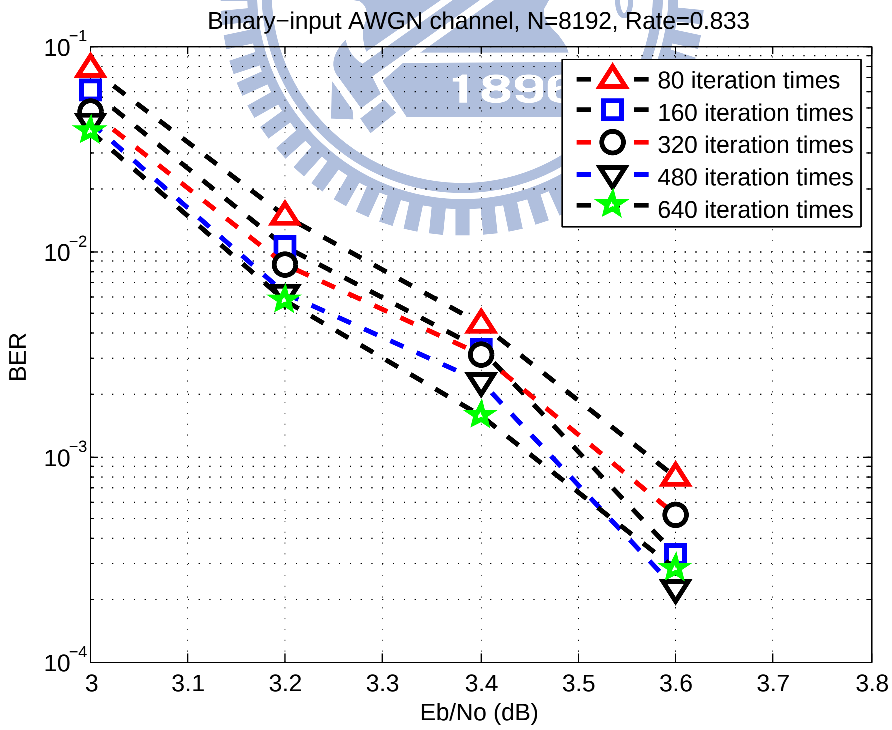


Figure 4.16: Error performance comparison of increasing iteration times under BP decoding for code rate is 5/6 and block length is 8192.

Chapter 5

Conclusion

In the incremental updating schedule, if we partition N bits into more blocks, the error performance will get better. In Fig. 4.7 and Fig. 4.8 we can see that the type with 2^{10} blocks is better than the other two types. The type with 2^8 blocks is the worst one among all three. For BP decoding, error performance will get better as number of iterations grows. For incremental updating schedule, if we partition N bits into more blocks, error rates will get better, too. For the cases, in Fig. 4.9, Fig. 4.11, Fig. 4.13 and Fig. 4.15, error performance in incremental updating schedule is better than error performance under BP decoding.

However, in some cases like Fig. 5.1 and Fig. 5.3, decoding error rate in incremental updating schedule is not better than that under BP decoding. In Fig. 5.1, we even have partitioned with minimum block size. For a specific rate or block length, the block number we should partition into so that decoding in incremental updating schedule will be better than BP decoding is not easy to choose and calls for future works.

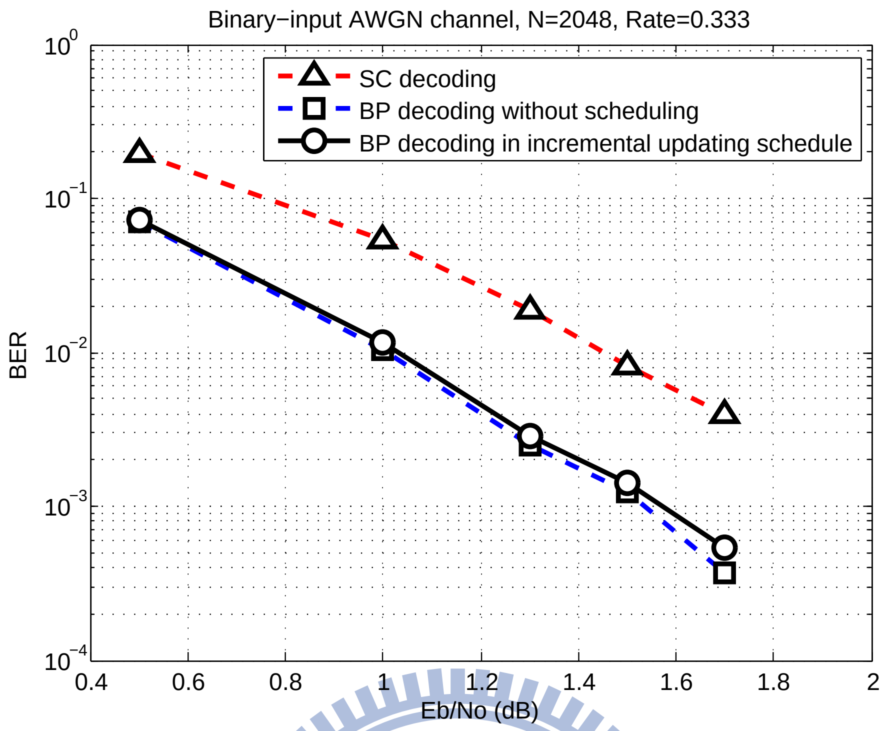


Figure 5.1: Error performance comparison for code rate is 1/3 and block length is 2048.

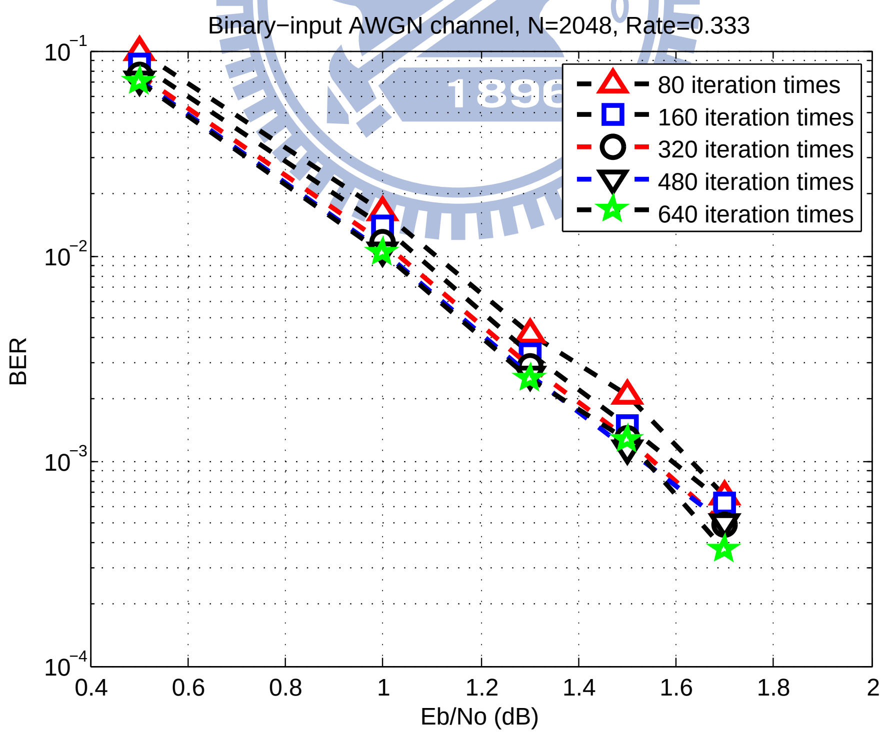


Figure 5.2: Error performance comparison of increasing iteration times under BP decoding for code rate is 1/3 and block length is 2048.

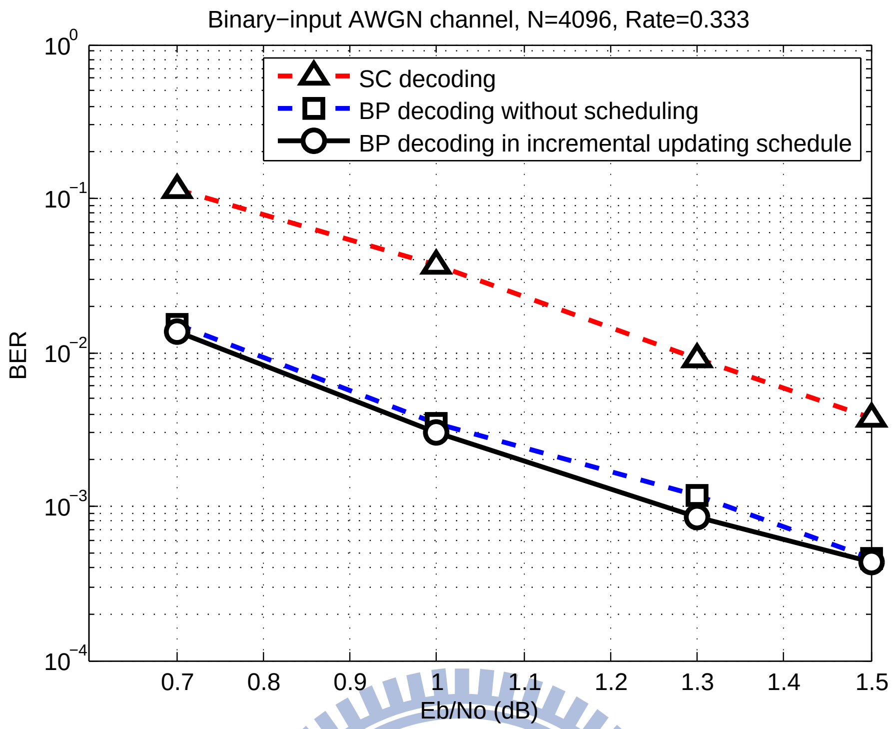


Figure 5.3: Error performance comparison for code rate is 1/3 and block length is 4096.

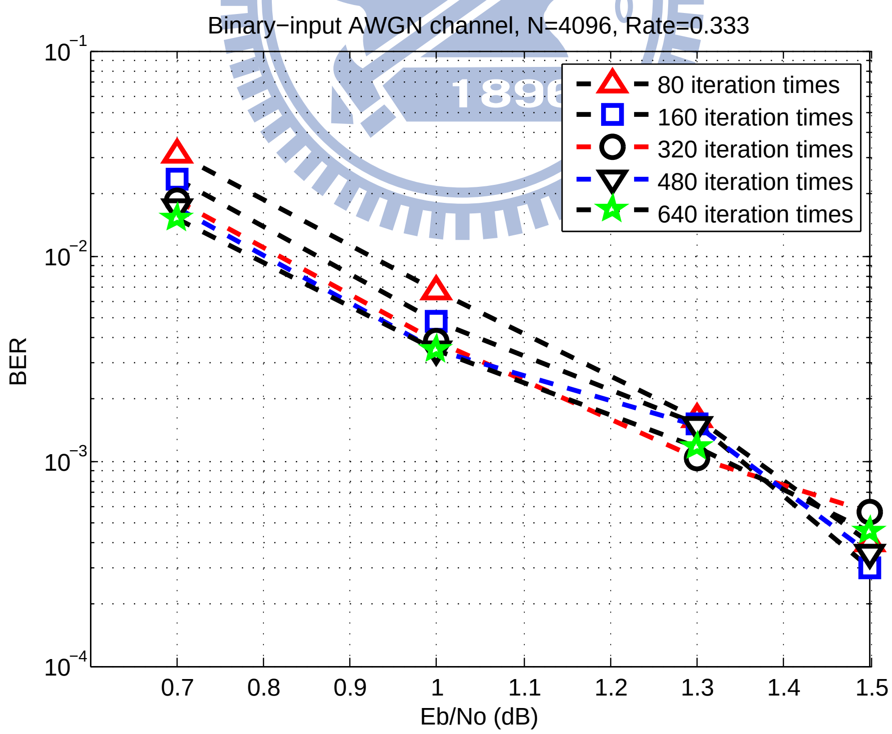


Figure 5.4: Error performance comparison of increasing iteration times under BP decoding for code rate is 1/3 and block length is 4096.

Reference

[1] N. Hussami, S. B. Korada and R. Urbanke, "Perfoamance of polar codes for channel and source coding," *IEEE International Sympousion on Information Theory,* 2009.

[2] E. Arikan, "Channel polarization: a method for constructing capacity-achieving codes for symmetric binary-input memoryless channels,"*IEEE Transactions on Information Theory,* vol. 55, pp. 3051-3073, Jul. 2009.

[3] G. D. Forney Jr., "Codes on graphs: normal realizations," *IEEE Transactions on Information Theory,* vol. IT-47, pp. 520-548, Feb. 2001.

[4] R. Mori and T. Tanaka, "Perfoamance of polar codes with the construction using density evolution," *IEEE Communications Letters,* vol. 13, no. 7, pp. 519-521, Jul. 2009.

[5] T. Richardson and R. Urbanke, *Modern Coding Theory.* Cambridge University, 2008.

[6] E. Arikan, "A perfoamance comparison of polar codes and reed-muller codes," *IEEE Communications Letters,* vol. 12, no. 6, pp. 447-449, Jun. 2008.

[7] A. Eslami and H. Pishro-Nik, "On bit error rate performance of polar codes in finite regime," 48*th Annual Allerton Conference on Communication, Control and Computing,* 2010.

Biography

I received my BS degree from Department of Engineering and System Science, National Tsing Hua University in 2009.

I received my MS degree from Institute of Communications Engineering, National Chiao Tung University in 2011.

Prof. Hsiao-Feng Lu was my advisor. During the master study, my research field was error-correcting coding.