Channel Coding of Weighted Probabilistic Model

Jielin Wang

Hunan International Economics University

E-mail: 254908447@qq.com

***Abstract*—Let the binary Bernoulli source sequence input into the symmetric discrete memoryless channel (DMC) be X, and linearly substitute the "1" in X with "10" to obtain the sequence Q, then the sequence Q satisfies the condition that each "1" is separate by one or multiple "0". There are lots of ways to convert sequence X into sequence Q, which are collectively referred to as source processing. The transmitting end uses a weighted probability model to linearly encode the sequence Q. The receiving end performs weighted probability model decoding. If more than one "1" is continuously decoded, the data transmission has errors. Then the forward error correction decoding method is constructed based on the characteristics of BSC (ξ) and BEC (ε) channel model. It has been proved that when the code length approaches infinity, the coding method in this paper enables the transmission rate to approximate the channel capacity, and the average decoding error probability approaches 0. Simulated with 1/2 code rate in the Binary Input Additive White Gaussian Noise (BIAWGN) channel, the block error rate (BLER) of this method is lower than that of Polar codes and Low-Density Parity-Check Codes (LDPC).**

***Keywords*—Information entropy, Channel capacity, Channel error correction, Weighted probability**

I: INTRODUCTION

To construct a coding method that approximates the channel capacity, experts and scholars have made unremitting efforts. In 2009, Arikan proposed a coding method based on channel polarization, which was strictly proven to be up to capacity when the code length approached infinity; it was called polar code[1]. LDPC code[2] and Turbo code[3] can also approach Shannon limit. A simple and easy-to-implement linear channel encoding and decoding method based on bits is constructed in this paper. It is different from the existing codes such as Polar codes, LDPC codes and Turbo codes. It is an innovation of basic coding theory. The theoretical proof is given in the article, and it is concluded that this method enables the DMC transmission rate to approximate the channel capacity, and the average decoding error probability approaches 0. According to Shannon noisy channel coding theorem[4], the method in this article belongs to the "good code", which is divided into two parts for the convenience of analysis and proof:

First, construct a linear processing method for binary signal source sequence, referred to as source processing. Source processing makes the binary sequence have error checking conditions. There are many methods of source processing, and because of the different amount of redundant information added, there are different decoding error probabilities. The article analyzes the reason why conditional probability and Markov chain model cannot construct linear channel encoding and decoding methods.

Then a linear coding algorithm with a weighted probability model is constructed, and the transmitting end encodes the sequence in units of bits after source processing through the weighted probability model. The receiving end performs linear decoding through a weighted probability model, and performs error checking during decoding. Do forward error correction or retransmission after an error is found. The source processing method is different, then the code rate of the weighted probability model is different.

II: SOURCE PROCESSING

The bit length agreed by the transmitter and receiver is n. The transmitter generates a binary Bernoulli sequence X of length $n(n = 1,2,\dots)$ from the source. Linearly substitute "1" in X with "101" and "0" with "01" to get the sequence Q. There are many similar methods, such as linearly substituting "1" in X with "10" to get sequence Q or linearly substituting "011" in X with "0110" to get sequence Q. Obviously, sequence Q presents the following morphological characteristics:

"The number of consecutive symbols 0 in the sequence is at most s, and the number of consecutive symbols 1 in the sequence is at most t" (2-1)

Denote $s \in N^*$ and $t \in N^*$ appearing in pairs as (s, t). Condition (2-1) is the basis for judging data error checking. Because (s, t) is different, the amount of redundant information in the sequence is different. Next, analyze the average decoding error probability of $(s \to \infty, t = 1)$ and $(s = 1, t = 2)$, which is denoted as P_{err}. Sequence Q (the length is denoted as l, $l \geq n$) is transmitted through DMC, and Y is the received binary sequence.

2.1 $(\boldsymbol{s \to \infty, t = 1})$

According to $(s \to \infty, t = 1)$, the sequence Q needs to meet the condition:

The number of consecutive symbols 1 is at most 1 (2-2)

Since sequence X is generated by source and it cannot be determined whether the sequence X

satisfies Condition (2-2), any sequence X needs to be processed linearly:

$$\text{Substitute “1” with “10” in sequence } X \tag{2-3}$$

The sequence Q is obtained after processing, and Q must satisfy Condition (2-2). For example: X is 0110111100101, according to (2-3), the sequence Q is 010100101010100010010. From left to right, substitute “10” with “1” in sequence Q to get sequence X. The process of (2-3) is the source processing.

Let event E represent the set of sequences Y satisfying Condition (2-2), and event E has $f(s \to \infty, t = 1, l = i)(i = 1,2, \dots)$ sequences Y.

When $i = 1$, $E = (0,1)$, $f(s \to \infty, t = 1, l = 1) = 2$, and the complementary event is $\bar{E} = \emptyset$.

When $i = 2$, $E = (00,01,10)$, $f(s \to \infty, t = 1, l = 2) = 3$ and $\bar{E} = (11)$.

When $i = 3$, $E = (000,001,010,100,101)$, $f(s \to \infty, t = 1, l = 3) = 5$ and $\bar{E} = (011,110,111)$.

By analogy, when $i \geq 3$:

$$f(s \to \infty, t = 1, l = i) = f(s \to \infty, t = 1, l = i - 1) + f(s \to \infty, t = 1, l = i - 2) \tag{2-4}$$

So the probability of event E is:

$$\mathrm{P}(E) = \frac{f(s \to \infty, t = 1, l = i)}{2^i} \tag{2-5}$$

Let $f(s \to \infty, t = 1, l = i)$ sequences Y in event E obey uniform distribution, then:

$$\begin{aligned} \mathrm{P}(Y = Q) &= \frac{1}{f(s \to \infty, t = 1, l = i)} \\ \mathrm{P}(Y \neq Q) &= \frac{f(s \to \infty, t = 1, l = i) - 1}{f(s \to \infty, t = 1, l = i)} \end{aligned} \tag{2-6}$$

Therefore, the probability of $Y \in E$ and $Y \neq Q$ is:

$$\mathrm{P}(Y \neq Q | Y \in E) = \mathrm{P}(E)\mathrm{P}(Y \neq Q) = \frac{f(s \to \infty, t = 1, l = i) - 1}{2^i} \tag{2-7}$$

$\mathrm{P}(Y \neq Q | Y \in E)$ is the average decoding error probability, denoted as $P(s \to \infty, t = 1, l = i)$, then $P_{err} = P(s \to \infty, t = 1, l = i)$.

Theorem 2.1 Sequence Y satisfies Condition (2-1) and $(s \to \infty, t = 1)$, $\lim_{i \to \infty} P(s \to \infty, t = 1, l = i) = 0$.

Proof Since $i \to \infty$, then $\mathrm{P}(Y \neq Q) \to 1$, so $\mathrm{P}(Y \neq Q | Y \in E) \to \mathrm{P}(E)$. According to the Fibonacci sequence, let $F(0) = 0$, $F(1) = 1$, and when $i \geq 2, i \in N^*$, $F(i) = F(i - 1) + F(i - 2)$. So when $i \geq 1, i \in N^*$, $f(s \to \infty, t = 1, l = i) = F(i) + F(i + 1)$. From the general term of the

Fibonacci sequence, we obtain

$$f(s\to\infty,t=1,l=i)=\frac{1}{\sqrt{5}}\left(\left[\left(\frac{1+\sqrt{5}}{2}\right)^{i}-\left(\frac{1-\sqrt{5}}{2}\right)^{i}\right]+\left[\left(\frac{1+\sqrt{5}}{2}\right)^{i+1}-\left(\frac{1-\sqrt{5}}{2}\right)^{i+1}\right]\right)$$

and

$$\mathrm{P}(E)=\frac{1}{\sqrt{5}}\left(\left[\left(\frac{1+\sqrt{5}}{4}\right)^{i}-\left(\frac{1-\sqrt{5}}{4}\right)^{i}\right]+\left[\left(\frac{1+\sqrt{5}}{4}\right)^{i+1}-\left(\frac{1-\sqrt{5}}{4}\right)^{i+1}\right]\right)$$

Because $\frac{1+\sqrt{5}}{4}<1$ and $\frac{1-\sqrt{5}}{4}<1$, so when $i\to\infty$, $\mathrm{P}(E)\to 0$, that is, $\lim\limits_{i\to\infty}P(s\to\infty,t=1,l=i)=0$.

2.2 $(\boldsymbol{s=1,t=2})$

According to $(s=1,t=2)$, the sequence Q needs to meet the conditions:

The number of consecutive symbols 0 in the sequence is at most 1, and the number of consecutive symbols 1 in the sequence is at most 2 (2-8)

Its source processing method is

Substitute “1” with “101” and “0” with “01” in sequence X (2-9)

The sequence Q is obtained after processing, and Q must satisfy Condition (2-8). For example, X is 0110111100101, and Q is 0110110101101101101101010110101101 according to the process of (2-9). Substitute “01” with “0” and “101” with “1” in sequence Q, so sequence X can be restored.

Let event E represent the set of sequences Y satisfying Condition (2-8), and event E has $f(s=1,t=2,l=i)(i=1,2,...)$ sequences Y.

When $i=1$, $E=(0,1)$, $f(s=1,t=2,l=1)=2$ and the complementary event is $\bar{E}=\emptyset$.

When $i=2$, $E=(01,10,11)$, $f(s=1,t=2,l=2)=3$ and $\bar{E}=(00)$.

When $i=3$, $E=(010,101,011,110)$, $f(s=1,t=2,l=3)=4$ and $\bar{E}=(000,001,100,111)$.

By analogy, when $i\geq 4$:

$$f(s=1,t=2,l=i)=f(s=1,t=2,l=i-2)+f(s=1,t=2,l=i-3) \tag{2-10}$$

Then

$$\mathrm{P}(E)=\frac{f(s=1,t=1,l=i)}{2^{i}} \tag{2-11}$$

$$P_{err}=P(s=1,t=1,l=i)=\frac{f(s=1,t=2,l=i)-1}{2^{i}} \tag{2-12}$$

Theorem 2.2 If sequence Y satisfies Condition (2-1) and $(s=1,t=2)$, $\lim\limits_{i\to\infty}P(s=1,t=$

$1, l = i) = 0$.

Proof $f(s \to \infty, t = 2, l = i)$ and $f(s = 1, t = 2, l = i)$ are monotonically increase, and $f(s = 1, t = 2, l = i) \le f(s \to \infty, t = 2, l = i)$, that is $\frac{f(s=1,t=2,l=i)-1}{2^i} \le \frac{f(s\to\infty,t=1,l=i)-1}{2^i}$. According to Theorem 2.1, when $i \to \infty$, $\frac{f(s\to\infty,t=1,l=i)-1}{2^i} \to 0$, so $\lim_{i\to\infty} P(s = 1, t = 1, l = i) = 0$.

2.3 Analysis of Decoding Error Checking Methods

Supposing that there is an encoding and decoding method with bits as the unit, a binary sequence V transmitted through DMC is obtained after sequence Q is encoded and U is the received binary sequence. The receiving end decodes the binary sequence Y through U. Since the source sequence X is known and its length is n, when (s, t) is determined, the length l of the sequence Q is determined, so the expression of P_{err} is also determined.

i（bits）	$P(s \to \infty, t = 1, l = i)$	$P(s = 1, t = 1, l = i)$
32	0.001327807	3.16184E-06
64	1.50584E-06	5.9561E-12
112	5.75104E-11	1.53974E-20
256	3.20367E-24	2.66011E-46

Table 1 Set the length of sequence Y and calculate the average decoding error probability (Scientific notation)

Therefore, the encoding and decoding methods have different average decoding error probabilities through setting n. According to Theorems 2.1 and 2.2, n and (s, t) are known, and $\mathrm{P}(E)$ can be obtained by (2-5) and (2-11). Because the receiving end can only decode sequence Y through sequence U, the decoding error checking method is:

(1) $Y \in \bar{E} \to U \ne V, Y \ne Q$;

(2) $Y \in E \to \lim_{l\to\infty} \mathrm{P}(Y \ne Q | Y \in E) = 0, U = V, Y = Q$.

Obviously, when the code length is long enough, $Y \in \bar{E}$ then U is wrong; when $Y \in E$, the decoding is correct. From Table 1, when the code length is the same, $P(s = 1, t = 1, l = i) < P(s \to \infty, t = 1, l = i)$, and $(s = 1, t = 1)$ has a lower probability of decoding errors. Different (s, t) or n can be used in different channel environments.

III: THE ENCODING AND DECODING METHOD OF WEIGHTED PROBABILITY MODEL

Since the probability of the symbols in the sequence Q is known, and the current symbol state is related to the state of a finite number of adjacent symbols, can a Markov chain or conditional probability model be used to construct an encoding and decoding method?

Take $(s \to \infty, t = 2)$ as an example, let the sequence Q be 011001000011010, and the sequence Q is composed of “0”, “10” and “110”. Based on the analyses of Markov chain or conditional probability, there are three probability mass functions for symbol “0”, which are $p(0|0)$, $p(0|1)$ and $p(0|1,1)$. Symbol 1 has two probability mass functions, namely $p(1|0)$ and $p(1|1)$. When the transmitter performs linearly-encoding, because the sequence Q is known, the probability mass function used by each symbol can be accurately selected. However, the probability quality function cannot be accurately selected during linear decoding at the receiving end. If “0” has been decoded, since there are three probability quality functions for symbol “0”, it is impossible to correctly select the probability quality function to decode the next symbol. Similarly, the symbol “1” cannot select the probability mass function correctly. When “011” has been decoded, since the symbol “0” must follow “011”, there is the only choice $p(0|1,1)$. Because the probability quality function is not unique, it is not feasible to use Markov chain or conditional probability to construct channel encoding and decoding methods, and source coding can be constructed.

Taking $(s \to \infty, t = 1)$ as an example, let the sequence Q be 0101001010101000010010. The traditional source coding method is: from left to right, substitute “10” with “1” in sequence Q to get sequence X as 0110111100101, and then encode sequence X to approximate $H(X)$ (namely information entropy[4]). However, the traditional source coding method cannot perform error checking during decoding. If the sequence Q is encoded, since redundant information is added, $H(Q) > H(X)$, which the traditional source coding method cannot be approximated to $H(X)$.

Supposing that there is function $\varphi(p(x), r) = rp(x)$. $p(x)$ is the probability of symbol x. r characterizes the morphological characteristics of sequence Q and is called weight coefficient. $\varphi(p(x), r)$ is called the weighted probability mass function, and the encoding and decoding method is constructed based on $\varphi(p(x), r)$ below.

3.1 Weighted probability model coding

Definition 3.1 We let the discrete random variable be X, where $X \in \{0,1\}$. Furthermore, $\mathrm{P}\{X = a\} = p(a) (a \in \{0,1\})$, and the weighted probability mass function is $\varphi(a) = r\mathrm{P}\{X = a\} = rp(a)$. $p(a)$ is a probability mass function, $0 \leq p(a) \leq 1$, r is a weight coefficient, and

$$F(a)=\sum_{i\le a}p(i) \tag{3-1}$$

If $F(a,r)$ satisfies $F(a,r)=rF(a)$, then $F(a,r)$ is called a weighted cumulative distribution function, or a weighted distribution function for short. The sum of the weighted probabilities of all the symbols is $\sum_{a=0}^{k}\varphi(a)=r$.

According to (3-1), $F(X_i-1)=F(X_i)-p(X_i)$. when $X_i=0$, $F(X_i-1)=0$; when $X_i=1$, $F(X_i-1)=\varphi(0)$. Denote the weighted distribution function of sequence Q as $F(Q,r)$:

When $l=1$, $F(Q,r)=rF(X_1-1)+rp(X_1)F(X,r)=rF(X_1-1)+rp(X_1)$.

When $l=2$, $F(Q,r)=rF(X_1-1)+r^2F(X_2-1)p(X_1)+r^2p(X_1)p(X_2)$.

When $l=3$, $F(Q,r)=rF(X_1-1)+r^2F(X_2-1)p(X_1)+r^3F(X_3-1)p(X_1)p(X_2)+r^3p(X_1)p(X_2)p(X_3)$.

Let $\prod_{j=1}^{0}p(X_j)=1$, when $l\ge 1$,

$$F(Q,r)=\sum_{i=1}^{l}r^iF(X_i-1)\prod_{j=1}^{i-1}p(X_j)+r^l\prod_{i=1}^{l}p(X_i) \tag{3-2}$$

A set of weighted distribution functions satisfying Equation (3-2) is defined as a binary weighted model, which is henceforth referred to as a weighted model, and is denoted by $\{F(Q,r)\}$. Let

$$H_l=F(Q,r) \tag{3-3}$$

$$R_l=r^l\prod_{i=1}^{l}p(X_i) \tag{3-4}$$

$$L_l=H_l-R_l=\sum_{i=1}^{l}r^iF(X_i-1)\prod_{j=1}^{i-1}p(X_j) \tag{3-5}$$

where $X_i\in\{0,1\}, l=1,2,\dots..$ When $r=1$,

$$F(Q,1)=\sum_{i=1}^{l}F(X_i-1)\prod_{j=1}^{i-1}p(X_j)+\prod_{i=1}^{l}p(X_i) \tag{3-6}$$

Letting $H_l=F(Q,1)$, $R_l=\prod_{i=1}^{l}p(X_i)$ and $L_l=H_l-R_l$, we know that the arithmetic coding (region coding) [5],[6] is a lossless coding method based on the weighted distribution function of $r=1$. The weighted model can be extended to the case of $X_i\in\{0,1,2,\dots\}$, which is not discussed in this article.

Since X_i must take a value in A, $p(X_i)\ge 0$. Equations (3-3), (3-4), and (3-5) are interval columns, and L_i, H_i is the corresponding interval superscript and subscript of variable X_i of the source sequence X at time $i(i = 0,1,2,\dots,l)$. $R_i = H_i - L_i$ is the length of the interval. According to Equations (3-3), (3-4), and (3-5), the iterative expressions of weighted probability model linear coding are:

$$R_i = R_{i-1}\varphi(X_i), \qquad L_i = L_{i-1} + R_{i-1}F(X_i - 1, r), \qquad H_i = L_i + R_i \qquad \text{(3-7)}$$

Taking $(s \to \infty, t = 1)$ as an example, we let $r > 1$, and the three symbols of the sequence Q starting from the position of $i + 1$ are $0, 1, 0$. According to Equation (3-7), the coding process of the binary weighted model is shown in Fig. 2.

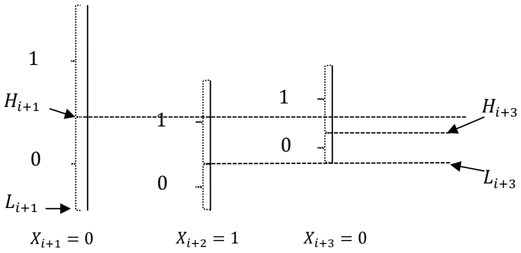

Figure 2: Process of coding 010 of binary weighted model

According to Fig. 2, if $H_{i+3} > H_{i+1}$, because interval $[H_{i+1}, H_{i+3}) \in [H_{i+1}, H_{i+1} + R_{i+1})$ and $[H_{i+1}, H_i + R_i)$ corresponds to symbol "1," the $(i + 1)$th symbol "0" may be erroneously decoded as symbol "1." If $H_{i+3} \le H_{i+1}$, $[L_{i+3}, H_{i+3}) \in [L_{i+1}, H_{i+1})$. As shown in Fig. 2, $[L_{i+1}, H_{i+1})$ uniquely corresponds to symbol "0," so symbol "0" at $i + 1$ is correctly decoded by L_{i+3}, and symbols "1" and "0" at $i + 2$ and $i + 3$, respectively, can also be decoded correctly. When $0 < r \le 1$, $[L_{i+1}, H_{i+1}) \in [L_i, H_i)$ can be decoded without loss at any time. Since $F(0 - 1) = 0$ and $F(0) = p(0)$, we can obtain the following from Equations (3-7):

$$\begin{gathered} H_{i+3} = L_i + R_i\varphi(0)^2 + R_i\varphi(0)^2\varphi(1) = L_i + R_i r^2 p(0)^2 + R_i r^3 p(0)^2 p(1) \\ H_{i+1} = L_i + R_i\varphi(0) = L_i + R_i r p(0) \end{gathered} \qquad \text{(3-8)}$$

Since $H_{i+3} \le H_{i+1}$,

$$\varphi(0)\varphi(1) + \varphi(0) = r^2 p(1)p(0) + rp(0) \le 1 \qquad \text{(3-9)}$$

We let $ar^2 + br + c = 0$, where $a = p(1)p(0)$, $b = p(0)$, $c = -1$, and $r > 0$. Thus, the positive real number that satisfies this quadratic equation is $\frac{\sqrt{p(0)^2 + 4p(1)p(0)} - p(0)}{2p(1)p(0)}$. Since $p(1) = 1 - p(0)$, when $p(1) = 0$ and $r \le 1$,

$$r \le \frac{\sqrt{4p(0) - 3p(0)^2} - p(0)}{2p(0) - 2p(0)^2} = \frac{\sqrt{4/p(0) - 3} - 1}{2 - 2p(0)} \qquad \text{(3-10)}$$

We let $r_{max} = \frac{\sqrt{4/p(0) - 3} - 1}{2 - 2p(0)}$, where r_{max} is the maximum value of r. r_{max} can be fully decoded by L_i only if the source sequence Q satisfies Condition (2-1).

We let there be $j+2(j=1,2,3,\ldots)$ symbols starting from the $(i+1)$-th symbol in the source sequence Q, that is, $0,1,\ldots,1,0$, where the number of consecutive symbols "1" is j. According to Condition (2-1), $j \le t$. Since $H_{i+j+2} \le H_{i+1}$, according to Equations (3-7), we have

$$\varphi(0)+\varphi(0)\varphi(1)+\varphi(0)\varphi(1)^2+\cdots+\varphi(0)\varphi(1)^j \le 1 \tag{3-11}$$

Then:

$$\varphi(0)+\varphi(0)\varphi(1)+\varphi(0)\varphi(1)^2+\cdots+\varphi(0)\varphi(1)^{j-1} \le \frac{1-\varphi(0)}{\varphi(1)} \tag{3-12}$$

Subtract (3-12) from (3-11) to simplify and obtain

$$r-r^{j+2}p(1)^{j+1}+r^{j+2}p(1)^{j+2} \ge 1 \tag{3-13}$$

$p(1)$ is known. If Inequality (3-13) is equal, we can obtain r_{max}. When $p(1)=1$ or $p(0)=0$, $r_{max}=1$; when $0<p(0)<1$ and $j\to\infty$, ${r_{max}}^{j+2}p(1)^{j+1}\to 0$, ${r_{max}}^{j+2}p(1)^{j+2}\to 0$, then $r_{max}\to 1$. When $j<t$ or $r<r_{max}$, $rp(0)+r^2p(0)p(1)+r^3p(0)p(1)^2+\cdots+r^{j+1}p(0)p(1)^j<1$.

3.2 Feasibility of lossless decoding

Theorem 3.2 The weighted model satisfies:

(1) $L_l<H_l \wedge L_l<H_{l-1} \wedge\ldots\wedge L_l<H_1$. The source sequence Q can be completely restored by L_l.

(2) $\lim_{l\to\infty}(H_l-L_l)=0$, that is astringency.

(3) $\lim_{l\to\infty}H_l=L_l$, that is uniqueness.

Proofs

(1) According to Inequality (3-11), $j>t$ or $r>r_{max}$, then we have $H_{i+j+2}>H_{i+1}$. Since $[H_{i+j+2},H_{i+1})$ corresponds to symbol "1", then the $(i+1)$-th symbol cannot be accurately decoded as symbol "0", which does not accord with the lossless decoding requirements, so the following must be satisfied at the same time: $0\le j\le t$ and $0<r\le r_{max}$. Since $F(0-1,r)=0$, $L_{i-1}\ge 0$ and $R_{i-1}\ge 0$, so L_l is a monotonic nondecreasing function. If and only if $L_l\in[L_l,H_l)\wedge L_l\in[L_{l-1},H_{l-1})\wedge\ldots\wedge L_l\in[L_1,H_1)$, because $[L_i,H_i)(i=1,2,\ldots,l)$ is uniquely mapped to the variable X_i, so when $L_l\in[L_i,H_i)(i=1,2,\ldots,l)$, the unique symbol X_i is obtained, so that the source sequence X can be completely obtained, so $L_l<H_l\wedge L_l<H_{l-1}\wedge\ldots\wedge L_l<H_1$.

(2) Since $j\le t$ and $r\le r_{max}$, there is $\varphi(0)+\varphi(0)\varphi(1)+\varphi(0)\varphi(1)^2+\cdots+\varphi(0)\varphi(1)^j\le$

1, so $H_{i+j+2} \leq H_{i+1}$. If and only if $j = t$ and $r = r_{max}$, $H_{i+j+2} = H_{i+1}$.

Let $R_{j+1} = \varphi(0)\prod_{i=1}^{j}\varphi(1)$, $R_j = \varphi(0)\prod_{i=1}^{j-1}\varphi(1)$,…, $R_2 = \varphi(0)\varphi(1)$ and $R_1 = \varphi(0)$, so $R_l = \prod R_{j+1}\prod R_j \dots \prod R_2 \prod R_1$. When $j < t$ and $r < r_{max}$, we can obtain $R_1 = \varphi(0) < 1$, $R_2 = \varphi(0)\varphi(1) < 1$, $R_3 = \varphi(0)\varphi(1)^2 < 1$, $\cdots$, $R_{j+1} = \varphi(0)\varphi(1)^j < 1$ from Inequality (3-11). Therefore, when $l \to \infty$, $R_l \to 0$, then $\lim_{l\to\infty}(H_l - L_l) = \lim_{l\to\infty} R_l = 0$, that is the weighted probability model is convergent.

(3) $\{L_l\}$ is a strictly monotonic nondecreasing sequence with an upper bound. Based on the monotone bounded theorem, let $\lim_{l\to\infty} L_l = \xi$ and $\xi \geq L_l$. Since $\lim_{l\to\infty}(H_l - L_l) = 0$, $\lim_{l\to\infty} L_l = \lim_{l\to\infty} H_l = \xi$. Thus, $\xi = L_l$, $\lim_{l\to\infty} H_l = \xi = L_l$, and L_l is unique.

Inference 3.3 Let $\varphi(1) = 1$. When $\varphi(0) \leq \frac{1}{t+1}$, the weighted model can completely restore the sequence Q through L_l.

Proof According to Inequality (3-11), when $\varphi(1) = 1$, $(t+1)\varphi(0) \leq 1$, then $\varphi(0) \leq \frac{1}{t+1}$.

According to Inference 3.3, since $r = \varphi(0) + \varphi(1)$, then $r \leq \frac{t+2}{t+1}$. But $r_{max} = \frac{t+2}{t+1}$ cannot be obtained. Taking $t = 1$ as an example, $r_{max} = \frac{3}{2}$, which is substituted into Inequality (3-9) to solve. When $p(0) \leq \frac{1}{3}$, Inequality (3-9) is tenable, and the weighted model satisfies Theorem 3.2(1). When $t = 1$, $p(0) \geq \frac{1}{2}$ in sequence Q, so $r_{max} \neq \frac{t+2}{t+1}$. Thus $r_{max} - {r_{max}}^{j+2}p(1)^{j+1} + {r_{max}}^{j+2}p(1)^{j+2} = 1 (j \leq t)$ is the necessary and sufficient conditions for model lossless encoding and decoding of weighted model.

3.3 Information entropy of weighted model

When $r = 1$, $\varphi(X_i) = p(X_i)$. The entropy of Q is

$$H(Q) = -p(0)\log_2 p(0) - p(1)\log_2 p(1) \tag{3-14}$$

When $r \neq 1$, the self-information of the random variable X_i with weighted probability $\varphi(X_i)$ is

$$I(X_i, r) = -\log\varphi(X_i) = -\log r - \log p(X_i) \tag{3-15}$$

Let the number of a in $\{X_i = a\}(i = 1,2,\dots,l, a \in A)$ be c_a. When the value of r is determined, the total amount of information of source sequence Q is

$$-c_0 \log \varphi(0) - c_1 \log \varphi(1)$$

Thus, the average amount of information per symbol is

$$-\frac{c_0}{l} \log \varphi(0) - \frac{c_1}{l} \log \varphi(1) = -p(0) \log \varphi(0) - p(1) \log \varphi(1) = -\log r + H(Q)$$

where $p(0) = \frac{c_0}{l}$ and $p(1) = \frac{c_1}{l}$ are the probability mass functions of symbols “0” and “1” in sequence Q. According to Section 3.1 and 3.2, $r \le r_{max}$. Since $r_{max} > 1$, $-\log r + H(Q) < H(Q)$. Because the weighted model cannot restore the sequence Q when $r > r_{max}$, $I(X_i, r)$ is the smallest when $r = r_{max}$. Then the information entropy of the weighted model is

$$\begin{aligned} H(Q, r_{max}) &= -\sum_{X_i=0}^{k} p(X_i) \log \varphi(X_i) \\ &= -\log r_{max} - \sum_{X_i=0}^{k} p(X_i) \log p(X_i) \\ &= -\log r_{max} + H(Q) \end{aligned} \tag{3-16}$$

3.4 The coding rate of weighted model

According to Section 2.3, the encoding and decoding of the weighted model satisfies:

(1) Symbol “0” and symbol “1” have unique probability mass functions $p(0)$ and $p(1)$ while encoding and decoding.

(2) Since (s, t) is known, there is a unique r_{max} corresponding to (s, t) and $r_{max} > 1$, so $-\log r_{max} < 0$, thus $H(Q, r_{max}) < H(Q)$. Therefore, the weighted probability model coding is closer to $H(X)$.

(3) $Y = Q$ and $Y \in E$ after U is decoded without error.

(4) When $n \to \infty$, $l \to \infty$ and $Y \in \bar{E}$, U is wrong; when $Y \in E$, U is correct, so $Y = Q$.

Therefore, the sequence Q is encoded as a sequence V by a weighted model, and V is transmitted through DMC, and U is the received binary sequence. The receiving end decodes the binary sequence Y with the weighted model through U.

According to (3-16), the average amount of information carried by each bit in the sequence Q is $H(Q, r_{max})(\text{bit/bit})$, and the total amount of information is $lH(Q, r_{max})(\text{bit})$. The total information of the source sequence X is $nH(X)(\text{bit})$, and the code rate of the weighted model can be obtained:

$$R = \frac{nH(X)}{lH(Q, r_{max})} \tag{3-17}$$

According to the BSC (ξ) channel model, sequence X is directly transmitted through BSC (ξ), and ξ is known. When the symbols in sequence X have equal probability ($p = \frac{1}{2}$), the transmission rate of BSC (ξ) is the largest. Then the BSC (ξ) channel capacity is $C_{BSC} = 1 - H(\xi)$ and $H(X) = 1$. According to the BEC (ε) channel model, the channel capacity is $C_{BEC} = 1 - \varepsilon$. From (3-17), the channel capacity of BSC (ξ) and BEC (ε) per unit time is:

$$C_{BSC} = R\big(1 - H(\xi)\big) \tag{3-18}$$

$$C_{BEC} = R(1 - \varepsilon) \tag{3-19}$$

Since sequence Q satisfies Condition (2-1), r_{max} and $r_{max} \geq 1$ can be obtained according to (3-13). The arithmetic coding limit of the sequence that satisfies $Y \in B$ and $Y = Q$ is $H(Q)$, that is, arithmetic coding is adopted in bit units for the sequence Q. Because $r_{max} \geq 1$ in the weighted model, so $-\log r_{max} \leq 0$, then $H(Q, r_{max}) \leq H(Q)$. Obviously, the weighted model coding can make the transmission rate of the channel higher. For channels with low transmission rate requirements, arithmetic coding (or interval coding) can be used.

IV: ERROR CHECK AND FORWARD ERROR CORRECTION ENCODING AND DECODING

$(s \to \infty, t = 1)$ is taken as an example to construct forward error correction encoding and decoding in this chapter. Let the probability of symbol “0” in a binary Bernoulli source sequence X of length n be $p (0 \leq p \leq 1)$, then $H(X) = -p \log_2 p - (1-p) \log_2(1-p)$. In the processing of (2-3) the sequence Q is obtained, and the length of sequence Q is $l = (2-p)n$, then $\frac{n}{l} = \frac{1}{2-p}$.

Theorem 4.1 $(s \to \infty, t = 1)$. When $n \to \infty$, $p = \varphi(0) = \frac{1}{2}$ and $\varphi(1) = 1$, the transmission rate of BSC(ξ) and BEC(ε) reaches the channel capacity, and the average decoding error probability approaches zero.

Proof When $p = \frac{1}{2}$, $nH(X) = n$. According to Inference 3.3, $H\left(Q, \varphi(0) = \frac{1}{2}, \varphi(1) = 1\right) = -\frac{1}{2-p} \log_2 \frac{1}{2} = \frac{1}{2-p}$, then $lH\left(Q, \varphi(0) = \frac{1}{2}, \varphi(1) = 1\right) = \frac{(2-p)n}{2-p} = n$. From (3-17) we obtain:

$$R = \frac{nH(X)}{lH\left(Q, \varphi(0) = \frac{1}{2}, \varphi(1) = 1\right)} = 1$$

Substitute it into (3-18) and (3-19) to obtain $C_{BSC}(t) = 1 - H(\xi)$ and $C_{BEC}(t) = 1 - \varepsilon$, so the transmission rates of BSC (ξ) and BEC (ε) reach the channel capacity. According to Theorem 2.1 and Theorem 2.2, when $n \to \infty$, $l \to \infty$ and $\lim_{l\to\infty} P(s \to \infty, t = 1, l = i) = 0$, that is, the average decoding error probability tends to be close to 0.

The probability mass functions of symbol “0” and symbol “1” in sequence Q are $p(0) = \frac{1}{2-p}$, $p(1) = \frac{1-p}{2-p}$, and $\frac{1}{2} \le p(0) \le 1$. According to (3-10), when $p(0) = 1$, $r_{max} = 1$ and $p = 1$; when $\frac{1}{2} \le p(0) < 1$, $r_{max} = \frac{(2-p)(\sqrt{5-4p}-1)}{2-2p}$.

$$\begin{aligned} lH(Q, r_{max}) &= -n\log_2\frac{\sqrt{5-4p}-1}{2-2p} - (1-p)n\log_2\frac{\sqrt{5-4p}-1}{2} \\ &= (2-p)n\log_2\frac{2}{\sqrt{5-4p}-1} + n\log_2(1-p) \end{aligned} \tag{4-1}$$

Theorem 4.2 $(s \to \infty, t = 1)$. When $n \to \infty$, the transmission rate of BSC(ξ) and BEC(ε) reaches the channel capacity, and the average decoding error probability approaches zero.

Proof According to (4-1),

$$\begin{aligned} lH(Q, r_{max}) - nH(X) &= (2-p)n\log_2\frac{2}{\sqrt{5-4p}-1} + (2-p)n\log_2(1-p) + pn\log_2 p \\ &= (2-p)n\log_2\frac{2-2p}{\sqrt{5-4p}-1} + pn\log_2 p = n\log_2\left(\left(\frac{2-2p}{\sqrt{5-4p}-1}\right)^{2-p} * p^p\right) \end{aligned}$$

Since $0 \le p \le 1$, $4(1-p)^2 \ge 0$, then $4 - 8p + 4p^2 \ge 0$. Since $4 - 8p + 4p^2 = (3 - 2p)^2 - (5 - 4p) \ge 0$, $2 - 2p \ge \sqrt{5-4p} - 1$. Since $(2-2p)^2 \ge \left(\sqrt{5-4p}-1\right)^2$, we can obtain $2p - 2p^2 \le \sqrt{5-4p} - 1$, then $\frac{\sqrt{5-4p}-1}{2-2p} \ge p$. Since $\left(\frac{2-2p}{\sqrt{5-4p}-1}\right)^{2-p} * p^p = \left(\frac{\sqrt{5-4p}-1}{2-2p}\right)^{p-2} * p^p \ge p^{2p-2} = \left(\frac{1}{p}\right)^{2-2p}$, $2 - 2p \ge 0$ and $\frac{1}{p} \ge 1$, $\left(\frac{1}{p}\right)^{2-2p} \ge 1$, that is $lH(Q, r_{max}) - nH(X) \ge 0$, then we can obtain $R = \frac{nH(X)}{lH(Q, r_{max})} \le 1$. It can be concluded that the transmission rate of BSC (ξ) and BEC (ε) can reach the channel capacity from (3-18) and (3-19).

Since $n \to \infty$ when $l \to \infty$, $\lim_{l\to\infty} P(s \to \infty, t = 1, l = i) = 0$, so the average decoding error probability approaches zero.

4.1 Encoding

According to theorem 4.1 and 4.2, a forward error correction encoding and decoding algorithm can be constructed. When $p \geq 0.652$, $H(Q, r_{max}) \leq \frac{1}{2-p} = \frac{n}{l}$ through calculation, that is, $lH(Q, r_{max}) \leq n$, so the weighted model coding has the effect of channel lossless compression. Since $\varphi(0) = \frac{1}{2}$ and $\varphi(1) = 1$ cannot adapt to changes of p, this section uses $\varphi(0) = r_{max}p(0)$ and $\varphi(1) = r_{max}p(1)$ for weighted model coding. So the basic operation variables are: $p(0) = \frac{1}{2-p}$, $p(1) = \frac{1-p}{2-p}$, $\varphi(0) = \frac{r_{max}}{2-p}$, $\varphi(1) = \frac{r_{max}(1-p)}{2-p}$, $r_{max} = \frac{(2-p)(\sqrt{5-4p}-1)}{2-2p}$.

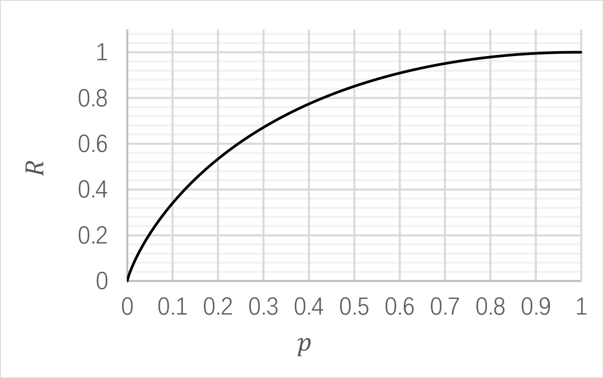

Figure 3 The relationship between the weighted model coding rate R and the probability p of symbol “0” in the sequence X

Fig. 3 shows that when $p < \frac{1}{2}$, the symbols in the sequence X are exchanged.

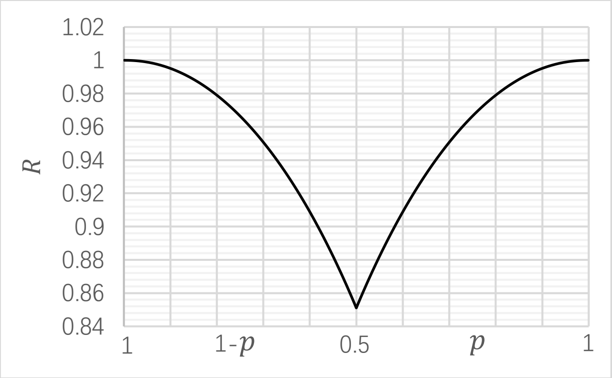

Figure 4 The relationship between R and $1-p$ when $p < \frac{1}{2}$, the relationship between R and p when $p \geq \frac{1}{2}$

It can be seen from Fig. 4 that the weighted model coding rate is the smallest when $p = \frac{1}{2}$, and $\min R = 0.85108$. According to (3-7), the weighting model is based on bit-based linear coding, combining the source processing of sequence X in the coding step. There are two situations when coding:

(1) When $p \geq \frac{1}{2}$, $R_i = R_{i-1}\varphi(0)$ and $L_i = L_{i-1}$ while encoding the symbol “0” in the sequence X; because of $(s \to \infty, t = 1)$ the actual encoded symbols are “10”, then $R_i = R_{i-1}\varphi(1)\varphi(0)$ and $L_i = L_{i-1} + R_{i-1}\varphi(0)$ while encoding the symbol “1” in the sequence X. The encoding steps are as follow:

1): Initialization parameters, so $R_0 = 1$, $L_0 = 0$, $p = 0$ and $i = 1$;

2): Record the number of symbols “0” in the statistical sequence X as c, and obtain $p = \frac{c}{n}$;

3): Calculate the weight coefficient and $r_{max} = \frac{(2-p)(\sqrt{5-4p}-1)}{2-2p}$;

4): Calculate the weighted probability, so $\varphi(0) = \frac{r_{max}}{2-p}$ and $\varphi(1) = \frac{r_{max}(1-p)}{2-p}$;

5): Obtain the i-th symbol x_i in the sequence X;

6): If $x_i = 0$, $R_i = R_{i-1}\varphi(0)$, or $R_i = R_{i-1}\varphi(1)\varphi(0)$ and $L_i = L_{i-1} + R_{i-1}\varphi(0)$;

7): $i = i + 1$. If $i < n$, repeat step 5 to step 7 to get L_n;

8): End encoding, return to L_n and c.

The above steps can be illustrated in Figure 5.

(2) When $p < \frac{1}{2}$, since the actual encoded symbols are “10” while encoding the symbol “0” in the sequence X, then $R_i = R_{i-1}\varphi(1)\varphi(0)$ and $L_i = L_{i-1} + R_{i-1}\varphi(0)$; $R_i = R_{i-1}\varphi(0)$ and $L_i = L_{i-1}$ while encoding the symbol “1” in the sequence X. The sequence X can be encoded by exchanging the relevant parameters of the above steps.

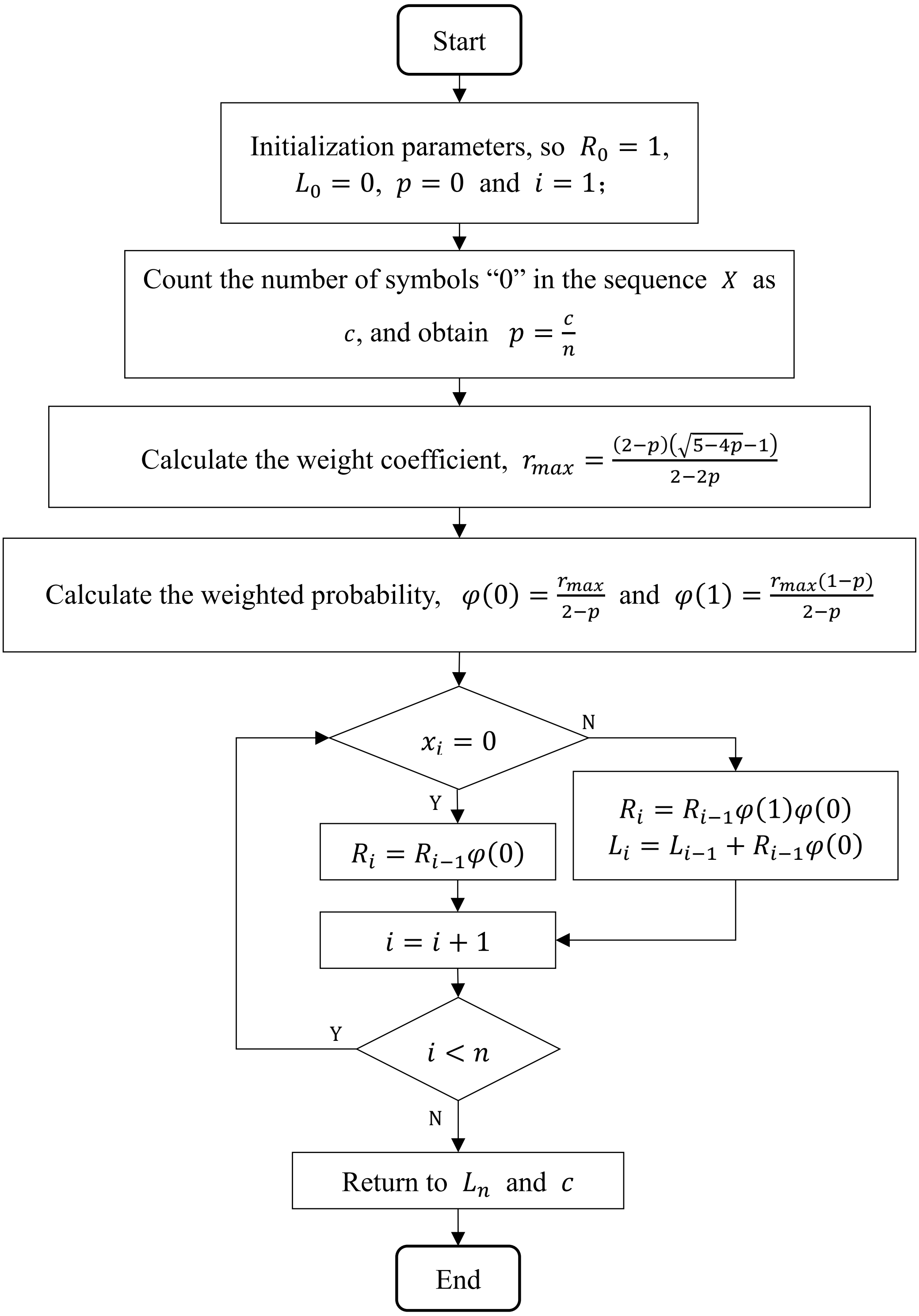
Start
Initialization parameters, so $R_0 = 1$, $L_0 = 0$, $p = 0$ and $i = 1$;
Count the number of symbols “0” in the sequence X as c, and obtain $p = \frac{c}{n}$
Calculate the weight coefficient, $r_{max} = \frac{(2-p)(\sqrt{5-4p}-1)}{2-2p}$
Calculate the weighted probability, $\varphi(0) = \frac{r_{max}}{2-p}$ and $\varphi(1) = \frac{r_{max}(1-p)}{2-p}$
$x_i = 0$
N
Y
$R_i = R_{i-1}\varphi(0)$
$R_i = R_{i-1}\varphi(1)\varphi(0)$
$L_i = L_{i-1} + R_{i-1}\varphi(0)$
$i = i + 1$
Y
$i < n$
N
Return to L_n and c
End

Figure 5 Schematic diagram of weighted model coding when $p \geq \frac{1}{2}$

According to Fig. 5, the weighted model coding algorithm in the case of $p \geq \frac{1}{2}$ and $p < \frac{1}{2}$ is as follows:

Algorithm(1)：Weighted model coding based on $(s \to \infty, t = 1)$

Input：array $XBitArray$ of sequence X with length n

Output：the array of bits $VBitArray$ and c

```
$R_0 \leftarrow 1; L_0 \leftarrow 0$
for $i \leftarrow 0$ to $n$
    if $XBitArray[i] = 0$ then
        $c \leftarrow c + 1;$
    end if
end for
$p \leftarrow \frac{c}{n}; p(0) \leftarrow \frac{1}{2-p};$
$r_{max} \leftarrow \frac{(2-p)\left(\sqrt{5-4p}-1\right)}{2-2p}; \varphi(0) \leftarrow r_{max}p(0); \varphi(1) \leftarrow r_{max}\left(1-p(0)\right);$
for $i \leftarrow 1$ to $n$
    if $XBitArray[i-1] = 0$ then
        if $p < 0.5$ then
            $R_i \leftarrow R_{i-1}\varphi(1)\varphi(0);$
            $L_i \leftarrow L_{i-1} + R_{i-1}\varphi(0);$
        else
            $R_i \leftarrow R_{i-1}\varphi(0);$
        end if
    else
        if $p < 0.5$ then
            $R_i \leftarrow R_{i-1}\varphi(0);$
        else
            $R_i \leftarrow R_{i-1}\varphi(1)\varphi(0);$
            $L_i \leftarrow L_{i-1} + R_{i-1}\varphi(0);$
        end if
    end if
end for
$VBitArray \leftarrow L_n;$
return $VBitArray, c;$
```

The pseudo-code in this article aims at realizing logic, where V, R_i, and L_i are defined as real numbers with infinite precision, and c can be transmitted without error. In practical applications, only need to substitute $\varphi(0)$ and $\varphi(1)$ into arithmetic coding (interval coding) to achieve weighted model encoding and decoding. Considering that c needs to be verified at the receiving end, it is agreed that the transmitter and the receiver use d bits to record the integer c. In Algorithm (1), $n = n + d$, c is stored in the first d bits of the array $XBitArray[n + d]$, and X is stored in the last n bits, then the weighted model encoding is performed. The receiving end can achieve the verification of c.

4.2 Decoding and error checking

The decoding process of data error checking is presented in this section. Because of

$(s \to \infty, t = 1)$, it can be determined that $U \neq V$ when two or more symbols “1” are decoded continuously. The receiving end obtains the binary sequence U and c, and denotes the length of the sequence U as m, so $m = (2n - c)H(Q, r_{max})$, where $p < \frac{1}{2}$. The error checking and decoding steps are as follows when $p \geq \frac{1}{2}$.

1): Initialization parameters, $R_0 = 1$, $L_0 = 0$, $i = 1, H = 0$, $s = 0$ and $p = \frac{c}{n}$;

2): Calculate the weight coefficient, and $r_{max} = \frac{(2-p)(\sqrt{5-4p}-1)}{2-2p}$;

3): Calculate the weighted probability, $\varphi(0) = \frac{r_{max}}{2-p}$, $\varphi(1) = \frac{r_{max}(1-p)}{2-p}$;

5): Calculate H_i when the i-th symbol $y_i = 0$, and $H_i = H = L_{i-1} + R_{i-1}\varphi(0)$;

6): When $U \geq H$ and $s = 1$, U is incorrect, then return null, and end decoding;

7): When $U \geq H$ and $s = 0$, $y_i = 1$.Uupdate s, R_i and L_i so that $s = 1$, $R_i = R_{i-1}\varphi(1)$ and $L_i = L_{i-1} + R_{i-1}\varphi(0)$;

8): When $U < H$ and $s = 1$, update s, R_i and L_i so that $s = 0$, $R_i = R_{i-1}\varphi(0)$ and $L_i = L_{i-1}$;

9): When $U < H$ and $s = 0$, $y_i = 0$. Update s, R_i and L_i so that $s = 0$, $R_i = R_{i-1}\varphi(0)$ and $L_i = L_{i-1}$;

10): $i = i + 1$. If $i < n$, repeat steps 5 to 10 to decode the sequence Y of n bits;

11): End encoding, and return to sequence $Y = (y_1, y_2, \dots, y_n)$.

The above steps can be illustrated in Fig. 6.

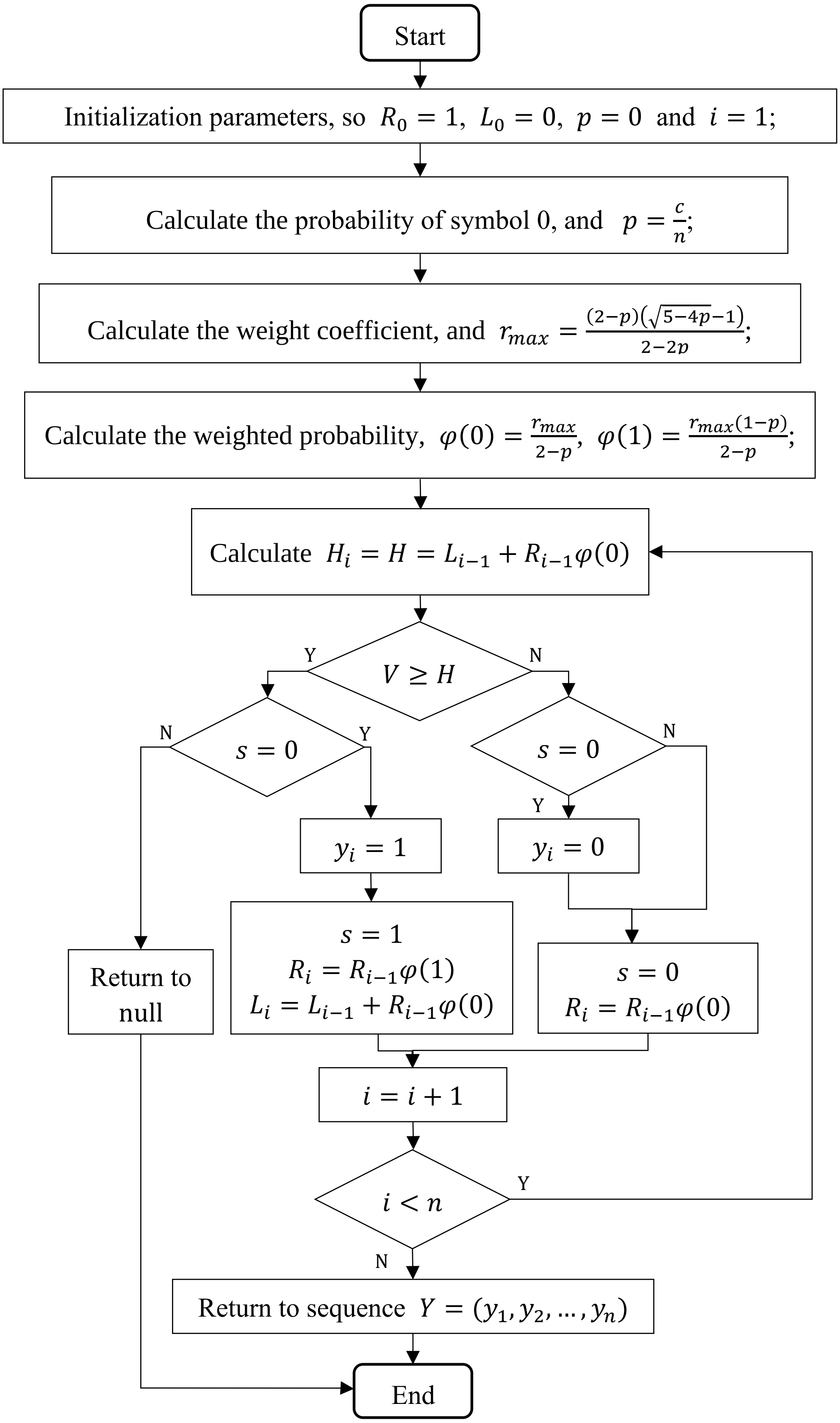
Start
Initialization parameters, so $R_0 = 1$, $L_0 = 0$, $p = 0$ and $i = 1$;
Calculate the probability of symbol 0, and $p = \frac{c}{n}$;
Calculate the weight coefficient, and $r_{max} = \frac{(2-p)(\sqrt{5-4p}-1)}{2-2p}$;
Calculate the weighted probability, $\varphi(0) = \frac{r_{max}}{2-p}$, $\varphi(1) = \frac{r_{max}(1-p)}{2-p}$;
Calculate $H_i = H = L_{i-1} + R_{i-1}\varphi(0)$
$V \geq H$
Y
N
$s = 0$
N
Y
$s = 0$
N
Y
$y_i = 1$
$y_i = 0$
Return to null
$s = 1$
$R_i = R_{i-1}\varphi(1)$
$L_i = L_{i-1} + R_{i-1}\varphi(0)$
$s = 0$
$R_i = R_{i-1}\varphi(0)$
$i = i + 1$
$i < n$
Y
N
Return to sequence $Y = (y_1, y_2, \dots, y_n)$
End

Figure 6 Schematic diagram of weighted model error checking and decoding when $p \geq \frac{1}{2}$

According to Fig. 6, the weighted model error check and decoding algorithm in the case of $p \geq \frac{1}{2}$ and $p < \frac{1}{2}$ is as follows.

Algorithm(2)：Weighted model error checking decoding based on $(s \to \infty, t = 1)$

Input：the array of bits $UBitArray$ and c

Output：the array of bits $YBitArray$ or null

```
$R_0 \leftarrow 1; L_0 \leftarrow 0; i \leftarrow 1; H \leftarrow 0; s \leftarrow 0; p \leftarrow \frac{c}{n}; p(0) \leftarrow \frac{1}{2-p};$
$U \leftarrow UBitArray;$
$r_{max} \leftarrow \frac{(2-p)(\sqrt{5-4p}-1)}{2-2p}; \varphi(0) \leftarrow r_{max}p(0); \varphi(1) \leftarrow r_{max}(1-p(0));$
while $i < n$
    $H \leftarrow L_{i-1} + \varphi(0)R_{i-1};$
    if $U \geq H$ then
        if $s = 1$ then
            return null;
        else
            if $p < 0.5$ then
                $YBitArray[i-1] \leftarrow 0;$
            else
                $YBitArray[i-1] \leftarrow 1;$
            end if
            $R_i \leftarrow R_{i-1}\varphi(1);$
            $L_i \leftarrow L_{i-1} + R_{i-1}\varphi(0);$
            $s \leftarrow 1;$
        end if
    else
        if $s = 0$ then
            $R_i \leftarrow R_{i-1}\varphi(0);$
        else
            $R_i \leftarrow R_{i-1}\varphi(0);$
            if $p < 0.5$ then
                $YBitArray[i-1] \leftarrow 1;$
            else
                $YBitArray[i-1] \leftarrow 0;$
            end if
        end if
        $s \leftarrow 0;$
    end if
    $i \leftarrow i + 1;$
end while
return $YBitArray;$
```

Firstly, decode d bits through Algorithm (2), and check whether c is correct. If c is correct, then continue to decode. When Algorithm (2) returns to null, it means that U is wrong.

4.3 Forward error correction decoding of BSC (ξ)

According to the BSC (ξ) channel model, there is a probability ξ so that symbol “0” is received as symbol “1”, or symbol “1” is received as symbol “0”. Based on the weighted model, when Algorithm (2) returns null, some symbols “0” or “1” in the binary sequence U are incorrect. Suppose there are e bit errors in the sequence U:

When $e = 1$, the forward error correction decoding steps are as follows.

1): Initialization parameters, $i = m - 1$;

2): The i-th bit of sequence U is negated so that $u_i = \overline{u_i}$;

3): Input U into Algorithm (2) for data verification and decoding;

4): When $i > 0$ and Algorithm (2) returns to null, $u_i = \overline{u_i}$ and $i = i - 1$, then repeat steps 2 to 4; when $i > 0$ and Algorithm (2) returns to sequence Y, end error correction coding; when $i = 0$ and Algorithm (2) returns to null, then $e > 1$.

The above steps can be illustrated in Figure 7.

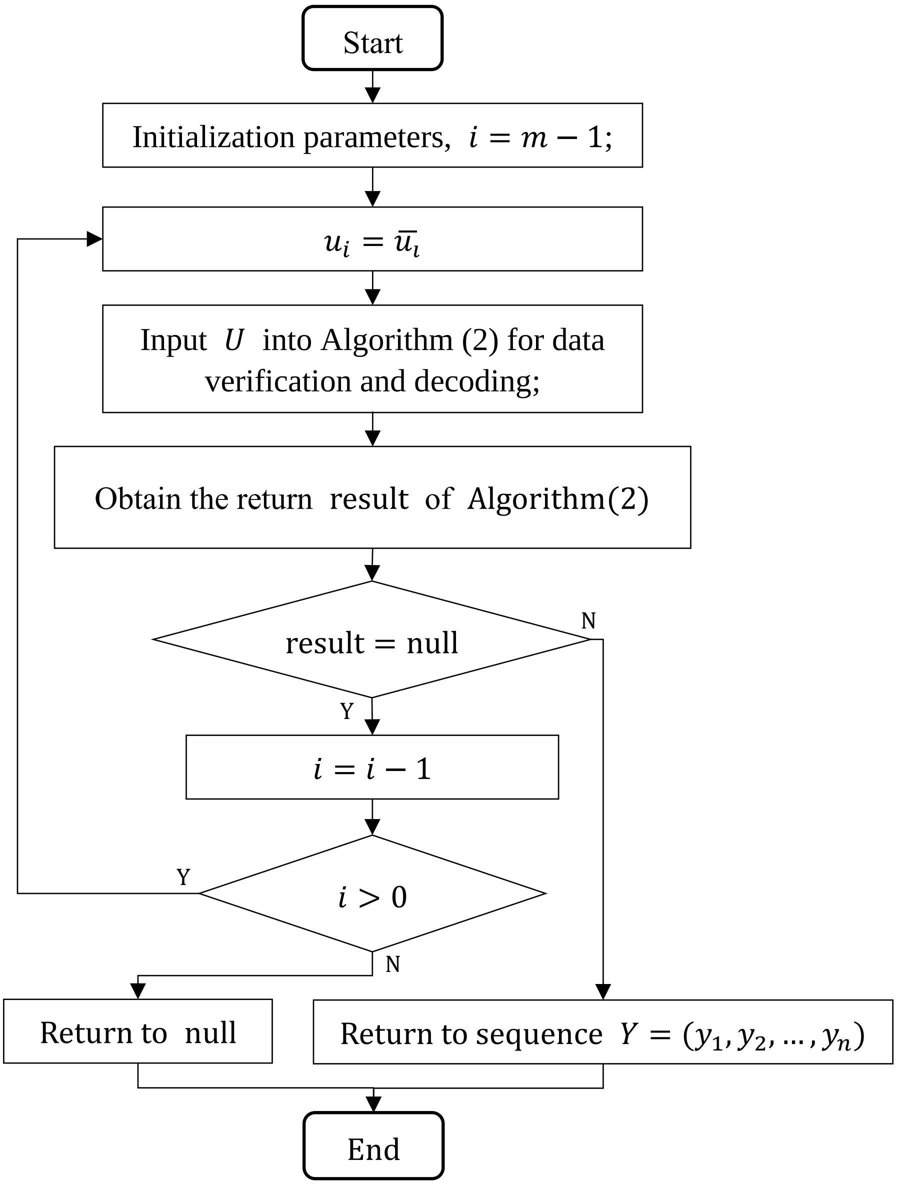
Start
Initialization parameters, $i = m - 1$;
$u_i = \overline{u_\imath}$
Input U into Algorithm (2) for data verification and decoding;
Obtain the return result of Algorithm(2)
result = null
N
Y
$i = i - 1$
$i > 0$
Y
N
Return to null
Return to sequence $Y = (y_1, y_2, \dots, y_n)$
End

Figure 7 Forward error correction decoding when $e = 1$

According to Fig. 7, the error correction decoding algorithm when $e = 1$ is as follows.

Algorithm(3)：Perform forward error correction decoding when $e = 1$ based on $(s \to \infty, t = 1)$ and BSC (ξ)

Input：the array of bits $UBitArray$

Output：the array of bits $YBitArray$ or null

```
for $i \leftarrow m - 1$ to 0
    $UBitArray[i] \leftarrow$ not $UBitArray[i]$;
    $YBitArray \leftarrow$ Algorithm(2) $\leftarrow UBitArray$;
    if $YBitArray =$ null then
        $UBitArray[i] \leftarrow$ not $UBitArray[i]$;
    else
        break;
    end if
end for
```

```
**return** $YBitArray$;
```

Algorithm (3) can traverse $C_m^1 = m$ times at most. If it traverses m times and returns null, it means that the sequence U has more than a bit error. Then perform the forward error correction algorithm of $e = 2$:

Algorithm(4)：Perform forward error correction decoding when $e = 2$ based on $(s \to \infty, t = 1)$ and BSC (ξ)

Input：the array of bits $UBitArray$

Output：the array of bits $YBitArray$ or null

```
for $i \leftarrow m - 1$ to $0$
    $UBitArray[i] \leftarrow$ not $UBitArray[i]$;
    for $j \leftarrow i - 1$ to $0$
        $UBitArray[j] \leftarrow$ not $UBitArray[j]$;
        $YBitArray \leftarrow$ Algorithm$(2)$ $\leftarrow UBitArray$;
        if $YBitArray =$ null then
            $UBitArray[j] \leftarrow$ not $UBitArray[j]$;
            $j \leftarrow j - 1$;
        else
            break;
        end if
    end for
    if $YBitArray =$ null then
        $UBitArray[i] \leftarrow$ not $UBitArray[i]$;
        $i \leftarrow i - 1$;
    else
        break;
    end if
end for
return $YBitArray$;
```

Algorithm (4) can traverse $C_m^2 = m$ times at most. If it traverses all times and then returns null, it means that the sequence U has more than two errors. By analogy, because $e \leq m$, a total of $\sum_{i=1}^{m} C_m^i = 2^m$ traversals are required to achieve all possible types of error correction. Obviously, P_{err} is the only decoding error probability during forward error correction decoding of BSC (ξ).

4.4 Forward error correction decoding of BEC (ε)

According to the BEC (ε) channel model, there is a probability ε such that symbol “0” or symbol “1” is received as symbol s. Suppose there are e symbols s in the sequence U. Taking $e = 3$ as an example, the erroneous symbols are marked as $s_i, s_j, s_k (i < j < k)$. The forward error correction algorithm is as follows.

Algorithm(5)：Perform forward error correction decoding when $e = 3$ based on $(s \to \infty, t =$

1) and BEC (ε)

Input: the array of bits $UBitArray$

Output: the array of bits $YBitArray$ or null

```
for $i \leftarrow 0$ to $1$
    $s_i \leftarrow i$;
    for $j \leftarrow 0$ to $1$
        $s_j \leftarrow j$;
        for $k \leftarrow 0$ to $1$
            $s_k \leftarrow k$;
            $YBitArray \leftarrow \text{Algorithm}(2) \leftarrow UBitArray$;
            if $YBitArray \neq$ null then
                break;
            end if
        end for
        if $YBitArray \neq$ null then
            break;
        end if
    end for
    if $YBitArray \neq$ null then
        break;
    end if
end for
return $YBitArray$;
```

When e is determined, a total of 2^e traversals are required to achieve all possible types of error correction. Similarly, P_{err} is the only decoding error probability during forward error correction decoding of BEC (ε).

4.5 Solution to the incremental problem of P_{err}

Let sequence Y be $Y = (y_1, y_2, \dots, y_i, \dots, y_l)$, and sequence U be $U = (u_1, u_2, \dots, u_j, \dots, u_m)$. The receiving end knows c and n, and s and tc is known, so p and $p(0)$ are known. From (3-16), we obtain

$$l = \frac{m}{H(Q, r_{max})} \tag{4-2}$$

Then substitute l into (2-7) to obtain P_{err}, and P_{err} is only related to l. When $j \geq 1$, $\frac{m-j}{H(Q, r_{max})} < l$, that is, it is not enough to decode l bits, so P_{err} becomes larger. When $m - j = 0$, $l = 0$ and $P_{err} = 1$, that is, when $j = m$, the error cannot be verified, and there is a problem of increasing P_{err}.

To ensure that P_{err} remains constant or descending, the following two methods can be adopted:

(1) After the sequence $Q = (x_1, x_2, \dots, x_i, \dots, x_l)$ is encoded, encode more than l symbols 0, that is, $x_{i>l} = 0$. Therefore, when the symbol $y_{i>l}$ in the sequence Y is decoded as a symbol “1”, $U \neq V$.

(2) Let the symbol $v_{j>m} = 0$ in the sequence V, and the sequence V is decoded by the weighted model to obtain the sequence Y. When the symbol $y_{i>l}$ in the sequence Y is decoded as the symbol “1”, $U \neq V$.

According to the weighted coding process, $L_{2l} = L_{2l-1} = \cdots = L_{l+1} = L_l = V$. Since V is a decimal, no matter how many symbols “0” are added after V, it will not affect the value of V, such as $V = 0.365 = 0.36500\dots0$. Obviously, method (1) is equivalent to $v_{j>m} = 0$ in sequence V. When encoding, method (2) takes twice as much computation as method (1), so method (2) is generally used. Therefore, the sequence U is decoded by the method (2) to obtain a sequence Y with a length of $2l$. Let $Y_1 = (y_1, y_2, \dots, y_l)$ and $Y_2 = (y_{l+1}, y_{l+2}, \dots, y_{2l})$. When $Y_1 \in E$ and $y_{l+1} = y_{l+2} = \cdots = y_{2l} = 0$, $U = V$ and P_{err} remains unchanged. The above two methods are effective when m is a finite value. When $m \to \infty$, set the value of P_{err}, and sequence U can be segmented for error correction decoding.

Divide the sequence U and V into $\left\lceil \frac{m}{h} \right\rceil$ segments, where h is a known countable non-zero integer. Let $U_k = (u_k, u_{k+1}, u_{k+2}, \dots, u_{k+h})$ and $V_k = (v_k, v_{k+1}, v_{k+2}, \dots, v_{k+h})$. U_k is the k-th bit segment in the sequence U, V_k is the k-th bit segment in the sequence V, then $k = 0,1,2 \dots, \frac{m}{h}$. According to theorems 2.1 and 2.2, s and t are known, P_{err} is only related to l_k. Since $l_k = \frac{h}{H(Q, r_{max})}$, P_{err} is only related to h. According to method (1) and method (2), it can be obtained that at least $\frac{2h}{H(Q, r_{max})}$ bits must be decoded to make P_{err} unchanged or smaller, that is, at least U_k and U_{k+1} are required to decode sequence Y' with a length of $\frac{2h}{H(Q, r_{max})}$ (Y' is a binary sequence in sequence Y). When $Y' \in E$, $U_k = V_k$. Then the minimum data verification range that satisfies unchanged or smaller P_{err} is:

$$l_{min} = \frac{2h}{H(Q, r_{max})} \tag{4-3}$$

When $k = \left\lceil \frac{m}{h} \right\rceil$, it is necessary to define $U_{k+1} = (0,0, \dots, 0)$ according to method (2). When $y_{i>l} = 0$, $U_k = V_k$.

4.6 Error check range and forward error correction range

According to the analysis in section 4.5, the value of h can be customized according to the channel situation. Because U_k and U_{k+1} are needed to decode the sequence Y' of length $\frac{2h}{H(Q,r_{max})}$, when $Y' \in \bar{E}$, $U_k \neq V_k$ or $U_{k+1} \neq V_{k+1}$, then the bit error occurs in U_k or U_{k+1}. According to (4-3), since h is known, the forward error correction range is only related to $H(Q, r_{max})$, and the forward error correction range is U_k and U_{k+1} with a total of $2h$ adjacent bit. According to (4-3), the error check range is l_{min}.

The above analysis process and encoding and decoding are based on $\varphi(0) = r_{max}p(0)$ and $\varphi(1) = r_{max}p(1)$. When $\varphi(1) = 1$, according to Inference 3.3, $\varphi(0) \leq \frac{1}{t+1}$, then the weighted probability of symbol 0 in sequence Q is only related to t. Therefore, the maximum weighted probability $\varphi(0)$ of symbol 0 in the source processing methods $(s \to \infty, t = 1)$ and $(s = 1, t = 2)$ are $\frac{1}{2}$ and $\frac{1}{3}$ respectively.

Because $(s \to \infty, t = 1)$ and $(s = 1, t = 2)$ both make sequence Q have n symbols 0, according to (3-16), when $\varphi(1) = 1$, the symbol 1 in sequence Q does not carry information, so only the number of symbols 0 in the sequence Q determines the number of bits output after the weighted model is encoded.

Take $(s = 1, t = 2)$ as an example, let the probability of symbol 0 in sequence X be $p = \frac{1}{2}$. Since $p = \frac{1}{2}$, $H(X) = 1$. $H\left(Q, \varphi(0) = \frac{1}{3}, \varphi(1) = 1\right) = -\log_2 \frac{1}{3}$, and we can obtain $R = 1/(-\log_2 \frac{1}{3})$. Therefore, each encoded symbol 0 in the sequence Q outputs $-\log_2 \frac{1}{3}$ bits, and define that the coded output $2h$ bits require α symbols 0, then:

$$\alpha = \lceil 2h/(-\log_2 \frac{1}{3}) \rceil \tag{4-4}$$

Since there are two combinations of boundaries in sequence Q with α symbols 0, one is composed of α symbols “10” (or “01”) and the other is composed of α symbols “110” (or “011”), so the length value of sequence Y' must belong to the integer interval $[2\alpha, 3\alpha]$. However, the length of the sequence Y' is unknown during decoding. When the error check range is set to 3α, the check of all sequences Y' is satisfied, that is, a sequence Y' with a length of 3α is decoded each time for error checking.

The maximum number of symbols 0 in the 3α symbols of sequence Q is $\lceil 3\alpha/2 \rceil$, and the minimum value is α. According to (4-4) and (3-16), the forward error correction range is $3h$ adjacent bits encoded by $\lceil 3\alpha/2 \rceil$ symbols 0, namely $\frac{3}{2}\lceil 2h/(-\log_2\frac{1}{3}) \rceil\,(-\log_2\frac{1}{3}) = 3h$.

Through the above analyses, the error check range and the forward error correction range are different because of different source processing methods and weighted probabilities. The specific situation requires specific analysis.

4.7 Limit the decoding error probability of error-correcting bits

Suppose there are e bit errors in the sequence U of length m, e in the BSC channel is related to the bit error rate ξ, and e in the BEC channel is related to the bit error rate ε. According to theorem 2.1 and 2.2, since $\lim\limits_{l\to\infty} P_{err} = 0$, any error in the sequence U can be found. According to Section 4.3 and Section 4.4, the bits' number of forward error correction can be limited to τ. When $e > \tau$, the receiving end notifies the transmitting end to retransmit the sequence V; when $e \le \tau$, the receiving end performs forward error correction decoding based on the sequence U. The probability of more than τ bit errors in m bits is:

$$P_{\mathrm{BSC}}(\tau) = \sum_{i=\tau+1}^{m} C_m^i \xi^i (1-\xi)^{m-i} \tag{4-5}$$

$$P_{\mathrm{BEC}}(\tau) = \sum_{i=\tau+1}^{m} C_m^i \varepsilon^i (1-\varepsilon)^{m-i} \tag{4-6}$$

Equation (4-5) and (4-6) are the probabilities of retransmitted sequence V of the BSC and BEC channels respectively, and τ can be defined as a known integer. Since the receiving end does not perform forward error correction decoding when $e > \tau$, the decoding error probabilities of forward error correction when $e \le \tau$ are respectively:

$$P_{err} = P_{\mathrm{BSC}}(\tau) + \mathrm{P}(Y \ne Q | Y \in E) \tag{4-7}$$

$$P_{err} = P_{\mathrm{BEC}}(\tau) + \mathrm{P}(Y \ne Q | Y \in E) \tag{4-8}$$

Since $\lim\limits_{l\to\infty} \mathrm{P}(Y \ne Q | Y \in E) = 0$, $\lim\limits_{l\to\infty} P_{err} = P_{\mathrm{BSC}}(\tau)$ in BSC channel, and $\lim\limits_{l\to\infty} P_{err} = P_{\mathrm{BEC}}(\tau)$ in BEC channel.

V: SIMULATION EXPERIMENT

The experiment uses $(s = 1, t = 1)$ as source processing method. In the case of $R = \frac{1}{2}$ and in order to compare the block error rate (BLER) of this method with Polar codes, LDPC codes and

Turbo codes, let the probability of symbol “0” in a binary source sequence X of length n be p. When $p=\frac{1}{2}$, the DMC channel’s transmission rate is the largest, so $nH(X)=n$. The length of the sequence Q is $l=\frac{5}{2}n$ through source processing. According to Inference 3.3, $t=2$ in order to ensure $R=\frac{1}{2}$, so $\varphi(0)\le\frac{1}{3}$ and $\varphi(1)=1$.

When $\varphi(0)=\frac{1}{4}$,

$$lH\left(Q,\varphi(0)=\frac{1}{4},\varphi(1)=1\right)=-n\log_2\frac{1}{4}=2n$$

Then we obtain

$$R=\frac{nH(X)}{lH\left(Q,\varphi(0)=\frac{1}{4},\varphi(1)=1\right)}=\frac{n}{2n}=\frac{1}{2} \tag{5-1}$$

5.1 Encoding

According to the weighting model, “01” is actually coded while encoding the symbol “0” in the sequence X, so $R_i=R_{i-1}\varphi(0)\,\varphi(1)$ and $L_i=L_{i-1}+R_{i-1}\varphi(0)^2$. Since “101” is actually coded while encoding the symbol “0” in the sequence X, $R_i=R_{i-1}\varphi(0)\varphi(1)^2$ and $L_i=L_{i-1}+R_{i-1}\varphi(0)^2\varphi(1)$.

The forward error correction algorithm is as follows:

Algorithm (6)：Weighted model coding based on $(s=1,t=2)$

Input：array $XBitArray$ of sequence X with length n

Output：the array of bits $VBitArray$

```
$R_0 \leftarrow 1; L_0 \leftarrow 0$;
$\varphi(0) \leftarrow \frac{1}{4}; \varphi(1) \leftarrow 1$;
for $i \leftarrow 1$ to $n$
    if $XBitArray[i-1]=0$ then
        $R_i=R_{i-1}\varphi(0)\,\varphi(1)$;
        $L_i=L_{i-1}+R_{i-1}\varphi(0)^2$;
    else
        $R_i=R_{i-1}\varphi(0)\varphi(1)^2$;
        $L_i=L_{i-1}+R_{i-1}\varphi(0)^2\varphi(1)$;
    end if
end for
$VBitArray \leftarrow L_n$;
return $VBitArray$;
```

5.2 Decoding and checking

Firstly, check whether the decoding result is correct while decoding, that is, if $Y \in E$, the decoding result is correct, or U is wrong. The forward error correction decoding algorithm is as follows:

```
Algorithm(7)：Error checking decoding based on $(s = 1, t = 2)$
Input：the array of bits $UBitArray$ and $c$
Output：the array of bits $YBitArray$ or null
$R_0 \leftarrow 1; L_0 \leftarrow 0; i \leftarrow 1; j \leftarrow 1; H \leftarrow 0; s_1 = s_2 \leftarrow -1;$
$U \leftarrow UBitArray;$
$\varphi(0) \leftarrow \frac{1}{4}; \varphi(1) \leftarrow 1;$
while $i < n$
    $H \leftarrow L_{j-1} + rp(0)R_{j-1};$
    if $U < H$ then
        if $s_2 = s_1 = -1$ then
            $s_1 \leftarrow 0;$
            $R_j = R_{j-1}\varphi(0);$
        else if $s_2 = -1$ and $s_1 = 1$ then
            $s_2 \leftarrow 1;$ $s_1 \leftarrow 0;$
            $R_j = R_{j-1}\varphi(0);$
        else if $s_2 = 1$ and $s_1 = 1$ then
            return null;
        else if $s_1 = 0$ then
            return null;
        end if
        $j \leftarrow j + 1;$
    else
        if $s_2 = s_1 = -1$ then
            $R_j \leftarrow R_{j-1}\varphi(1);$
            $L_j = L_{j-1} + R_{j-1}\varphi(0);$
        else if $s_2 = -1$ and $s_1 = 0$ then
            $R_j \leftarrow R_{j-1}\varphi(1);$
            $L_j = L_{j-1} + R_{j-1}\varphi(0);$
            $YBitArray[i] = 0;$
            $i \leftarrow i + 1;$
            $s_2 \leftarrow -1;$ $s_1 \leftarrow -1;$
        else if $s_2 = -1$ and $s_1 = 1$ then
            $R_j \leftarrow R_{j-1}\varphi(1);$
            $L_j = L_{j-1} + R_{j-1}\varphi(0);$
            $s_2 \leftarrow 1;$ $s_1 \leftarrow 1;$
        else if $s_2 = 1$ and $s_1 = 0$ then
            $R_j \leftarrow R_{j-1}\varphi(1);$
            $L_j = L_{j-1} + R_{j-1}\varphi(0);$
            $YBitArray[i] = 1;$
            $i \leftarrow i+1$;
            $s_2 \leftarrow -1$; $s_1 \leftarrow -1$;
        **else if** $s_2 = 1$ **and** $s_1 = 1$ **then**
            **return null**;
        **end if**
        $j \leftarrow j+1$;
    **end if**
**end while**
**return** $YBitArray$;
```

When Algorithm (7) returns to null, it means that U is erroneous. The method in section 4.3 or 4.4 should be adopted to achieve forward error correction and decoding.

5.3 Simulation experiment for BSC (ξ) channel

The simulation experiment generates a binary Bernoulli sequence X with a length of $n = 1024\text{bit}$, and the probability of symbol 0 is $p = \frac{1}{2}$. The length of sequence Q through $(s = 1, t = 2)$ source processing is $l = 2560$. According to Theorem 2.2, $P_{err} = 0$. Since $\varphi(0) = \frac{1}{4}$, $\varphi(1) = 1$, and symbol 1 does not carry information, the length of sequence V after sequence Q is encoded with weighted model is $lH\left(Q, \varphi(0) = \frac{1}{4}, \varphi(1) = 1\right) = -1024\log_2\frac{1}{4} = 2048\ \text{bit}$. Sequence V is transmitted via BSC (ξ). U is the binary sequence received by the receiving end.

According to the analyses in section 4.5, 4.6 and 4.7, let $h = 32$, where 32 is the bit length of the INT data type in the computer, so the data verification range is $3\alpha = 3\lceil 2h/(-\log_2\frac{1}{3})\rceil = 122$, and the forward error correction range is $3h = 96$. The decoding error probability is $P_{err} \leq 2.50254E-22$ from (2-10), (2-11) and (2-12). Then according to the forward error correction decoding logic in Section 4.3, let $\tau = 8,12,16,18$, then the decoding error probability is given by (4-7), and retransmit the current frame when the error correction decoding cannot be performed in the experiment, and then calculate the block error rate (BLER) by counting the number of retransmissions.

Turbo code was simulated based on WCDMA and LTE standards. The maximum iteration of the Log-MAP decoding algorithm is $I_{max} = 8$, and the code length is 1024.

The LDPC code was simulated based on the WiMax standard, using the standard BP algorithm. The maximum iteration $I_{max} = 200$, and the code length is 1056.

Polar code was simulated based on Cyclic Redundancy Code (CRC) -assisted successive-Cancellation List (SCL) decoding algorithm (CRC-Asistant SCL), with a list size of 32 and a maximum code length of 1024.

The number of frames of the simulated Binary input Additive White Gaussian Noise (BIAWGN) channel based on modulation and demodulation method of Binary Phase Shift Keying (BPSK) is greater than 10^5. The coding rate of the four coding methods is $R = \frac{1}{2}$. The experiment shows that the block error rate (BLER) of the method in this article, Turbo code, LDPC code and Polar code under different signal-to-noise ratios E_b/N_0(SNR) is shown in Fig. 8.

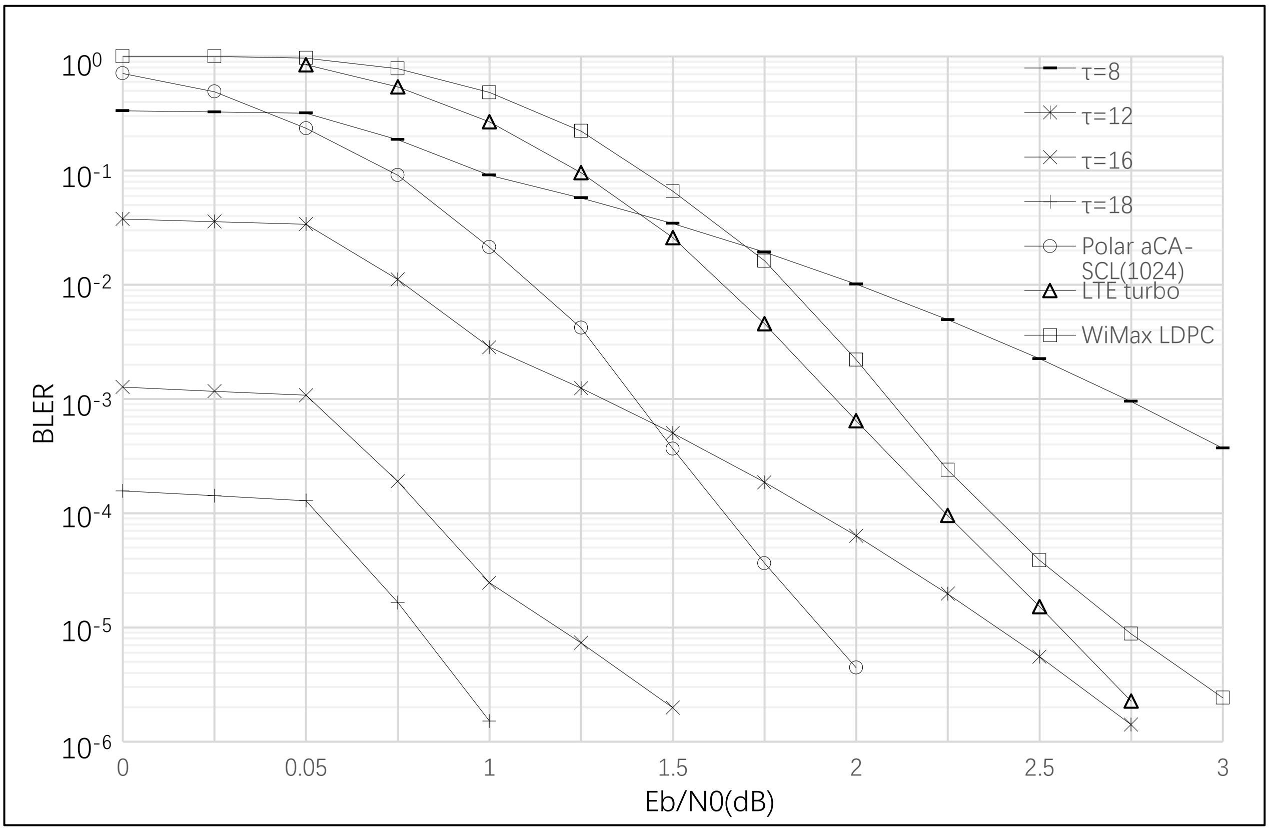

Figure 8: Performance comparison of the error-correction methods in this paper with the polar, turbo, and LDPC code in the case of unchanged τ

As shown in Fig. 8, the error correction performance of the method in this paper is better than the LDPC and polar code. When $\tau = 12$ and the signal-to-noise ratio is lower than 1.4dB, the method in this paper is better than Polar code. When $\tau = 18$, the binary weighted coding method reached a 1.0 dB gain compared to the polar code and 1.8–2.2 dB gain compared to the LDPC and turbo code.

According to Figure 8, the experimental results are in accordance with (4-5), and the value of τ remains unchanged at any signal-to-noise ratio. Next, experiments with different signal-to-noise ratios adopt different value τ, such as $\tau = 12$ at 0dB; $\tau = 13$ at 0dB to 1dB; $\tau = 15$ at 1dB to

1.5dB; $\tau = 18$ at 1.5dB or more. Through simulation experiments, we obtain that the block error rate (BLER) of the method in this paper, Turbo code, LDPC code and Polar code are shown in Fig. 9 under different signal-to-noise ratio E_b/N_0(SNR).

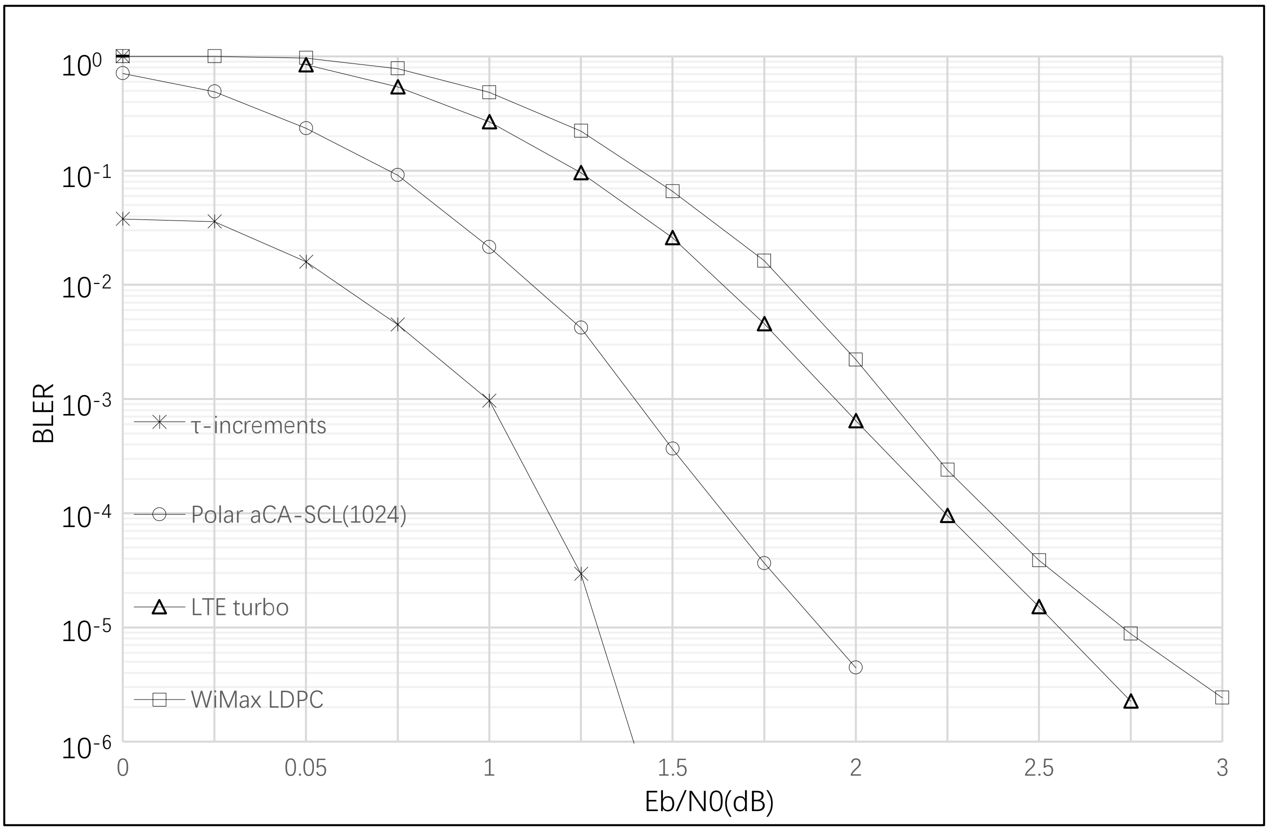

Figure 9: Performance comparison of the error-correction methods in this paper with the polar, turbo, and LDPC code when τ increases progressively

VI: CONCLUSION

A novel concept and specific method for error detection and correction was presented herein. Due to the variety of methods, this article presents examples of two methods, corresponding to different code rates and decoding error probability. It was proven theoretically that the channel capacity can be available. The method is simple and would be easily implemented in hardware and software, as it can adapt to interference in the channel, and the error-correction capabilities can be improved by increasing the value of τ. After controlling the size of τ, a channel coding method integrating forward error correction and data check and retransmission can be constructed. The calculation increases exponentially as τ increases. On the one hand, h is reduced based on the acceptable decoding error probability. On the other hand, the bit error position is related to the data check position. The error correction range can be refined according to the data check position to increase calculation efficiency. The weighted model coding in section 4.1 is also source coding, and an algorithm with dual coding of source and channel can be constructed in the future.

VII: REFERENCES

[1] Erdal Arikan. Channel Polarization: A Method for Constructing Capacity-Achieving Codes for Symmetric Binary-Input Memoryless Channels. IEEE Transactions on Information Theory, Volume:55, Issue:7, 3051 - 3073, July 2009.

[2] R. G. Gallager, "Low-Density Parity-Check Codes," M.I.T. Press, Cambridge, Massachusetts, 1963.

[3] Axel Huebner,Kamil Sh. Zigangirov, Daniel J. Costello.Laminated turbo codes: A new class of block-convolutional codes. IEEE Transactions on Information Theory,Volume: 54 , Issue: 7 , July 2008

[4] C. E. Shannon. A mathematical theory of communication. Bell Syst. Tech. J., 27:379–423,623–656, 1948.

[5] Ian H.Witten, Radford M.Neal,John G.Cleary. Arithmetic Coding for Data Compression.Communications of the ACM. 1987,30(6):520~539.

[6] G. N. N. Martin, Range encoding: an algorithm for removing redundancy from a digitised message. Video & Data Recording Conference, held in Southampton July 24-27 1979.

[7] Marpe D , Schwarz H , Wiegand T . Context-based adaptive binary arithmetic coding in the H.264/AVC video compression standard[J]. IEEE Transactions on Circuits & Systems for Video Technology, 2003, 13(7):620-636.

[8] T.M.Cover and J.A.Thomas,Elements of Information Theory.New YorK,Wiley 1991.

[9] F.j.MacWilliams and N.J.A Slone.Theory of Error-Correcting Codes.Amsterdam.The Netherlands:North-Hollend.1977.

[10] Berrou C , Glavieux A . Near optimum error correcting coding and decoding: turbo-codes[J]. IEEE Transactions on Communications, 1996, 44(10):P.1261-1271.

[11] Ghassan M. Kraidy.On Progressive Edge-Growth Interleavers for Turbo Codes.IEEE Communications Letters,Volume: 20 , Issue: 2 , Feb. 2016.

[12] Fang Yuan,Bin Tian. Double-Parity-Check CA-SCL Encoding and Decoding for Polar Codes.2018 14th IEEE

International Conference on Signal Processing.

[13] Mishra A , Raymond A J , Amaru L G , et al. A successive cancellation decoder ASIC for a 1024-bit polar code in 180nm CMOS[C]// IEEE Asian Solid-state Circuits Conference. IEEE, 2012.

[14] Mao-Ching Chiu.Interleaved Polar (I-Polar) Codes.IEEE Transactions on Information Theory,Volume: 66 , Issue: 4 , April 2020.

[15] Ali Dehghan,Amir H. Banihashemi.On the Tanner Graph Cycle Distribution of Random LDPC, Random Protograph-Based LDPC, and Random Quasi-Cyclic LDPC Code Ensembles.IEEE Transactions on Information Theory,Volume: 64 , Issue: 6 , June 2018.

[16] Mansour M M , Shanbhag N R . High-Throughput LDPC decoders[J]. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 2004, 11(6):976-996.

[17] Eran Pisek,Dinesh Rajan,Joseph R. Cleveland.Trellis-Based QC-LDPC Convolutional Codes Enabling Low Power Decoders.IEEE Transactions on Communications,Volume: 63 , Issue: 6 , June 2015.

[18] Alireza Tasdighi,Amir H. Banihashemi,Mohammad-Reza Sadeghi.Symmetrical Constructions for Regular Girth-8 QC-LDPC Codes. IEEE Transactions on Communications, Volume: 65, Issue: 1, Jan. 2017.