American Journal of Mathematical and Computer Modelling
2017; 2(1): 6-12
http://www.sciencepublishinggroup.com/j/ajmcm
doi: 10.11648/j.ajmcm.20170201.12

Decoding of the Five-Error-Correcting Binary Quadratic Residue Codes

Yani Zhang, Xiaomin Bao, Zhihua Yuan, Xusheng Wu

School of Mathematics & Statistics, Southwest University, Chongqing, China

Email address:
xbao@swu.edu.cn (Xiaomin Bao)

To cite this article:
Yani Zhang, Xiaomin Bao, Zhihua Yuan, Xusheng Wu. Decoding of the Five-Error-Correcting Binary Quadratic Residue Codes. *American Journal of Mathematical and Computer Modelling.* Vol. 2, No. 1, 2017, pp. 6-12. doi: 10.11648/j.ajmcm.20170201.12

Received: October 29, 2016; **Accepted**: November 28, 2016; **Published**: January 6, 2017

Abstract: In this paper, a new efficient syndrome-weight decoding algorithm (NESWDA) is presented to decode up to five possible errors in a binary systematic (47, 24, 11) quadratic residue (QR) code. The main idea of NESWDA is based on the property cyclic codes together with the weight of syndrome difference. The advantage of the NESWDA decoding algorithm over the previous table look-up methods is that it has no need of a look-up table to store the syndromes and their corresponding error patterns in the memory. Moreover, it can be extended to decode all five-error-correcting binary QR codes.

Keywords: Cyclic Codes, Decoding, Quadratic Residue Code

1. Introduction

The well-known QR codes, introduced by Prange in his report [1] in 1957, are cyclic BCH codes with code rates greater than or equal to one-half and generally have large minimum distances so that most of the known QR codes are among the best-known codes. The code augmented by a parity bit, for instance, the (24, 12, 8) QR code has be used for numerous communication links, including the Voyager imaging system link of NASA [2]. In the past decades, a series of different decoding methods given in [6-15] have been developed to decode the QR code. However, these algebraic decoding techniques require a large number of complicated computations in a finite field. These complicated computations will lead to a time delay in the decoding procedures and therefore the decoding time will become unrealistic when the code length is large. Therefor, to reduce the decoding complexity, we introduce the NESWDA algorithm, which can be used to decode the (47,24,11) QR code. It allows for the correction of up to $t=\lfloor (d-1)/2 \rfloor=\lfloor (11-1)/2 \rfloor=5$ errors, where $\lfloor x \rfloor$ denotes the greatest integer less than or equal to x; t is the error-correcting capability and $d=11$ is the minimum Hamming distance of the code.

Recently, table look-up decoding algorithms have been developed to decode the (47, 24, 11) QR code. The full lookup table in the conventional table look-up decoding algorithm (CTLDA) needs $\sum_{i=1}^{5}\binom{47}{i}=1729647$ syndromes and their corresponding error patterns. It requires 1729647 × (6bytes + 3bytes) ≈ 14.85 Mbytes memory size to store the table. Such a large memory required makes the computation very complicated. An efficient algorithm named look-up table decoding (LTD) algorithm in [3] to decode the (47, 24, 11) QR code needs 1.05 Mbytes memory size to store the table. The well-known syndrome decoder [2]

Requires

$$\left(\sum_{i=1}^{5}\binom{47}{i}\right)\Big/47=1729647/47=36801$$

syndromes corresponding to error patterns stored in a table with a 36801 × (6bytes +3bytes)=1024 ≈ 323.45 Kbytes memory size, called the reduced lookup table (RLT). However, this memory size of the RLT is still so large that one needs to further reduce the memory size of the look-up table. To achieve this end, an efficient table look-up decoding algorithm (TLDA) in [4] is developed to decode the (47, 24, 11) QR code, the memory size of the developed condensed look-up table consists of 36.6 Kbytes, and Lin et al. [5] used a novel table look-up decoding algorithm, called the cyclic weight (CW) decoding algorithm, together with a memory size 20.43 Kbytes to decode the (47, 24, 11) QR code. As shown in this paper, the proposed NESWDA does not need a memory size to store the look-up table. The prime idea of the proposed

NESWDA is based on the weight of syndrome difference between the syndrome of the received word and the row vector of the transpose of the parity-check matrix. Moreover, no complicated computation in the finite field is required in the proposed NESWDA and it also can be extended to decode all five-error-correcting binary QR codes.

The remainder of this paper is organized as follows: The background of the binary QR codes is briefly given in Section 2. The proposed NESWDA is described in Section 3. In Section 4, two examples are used to demonstrate the proposed NESWDA. Finally, this paper concludes with a brief summary in Section 5.

2. Background of the Binary QR Codes

The binary QR codes are a nice family of linear cyclic codes. Let (n,k,d) or $\left(\mathrm{n},(n+1)/2,d\right)$ denote the binary QR codes with generator polynomial $\mathrm{g}(x)$ over $GF(2)$. The length of this code is a prime number of the form $\mathrm{n}=8l\pm1$, where l is some integer. Also, let $\mathrm{k}=(\mathrm{n}+1)/2$ denote the message length or information length, and d denote the minimum Hamming distance of the code. The set Q_{n} of quadratic residues modulo n is the set of nonzero squares modulo n; that is, $Q_{\mathrm{n}}=\left\{j \mid j\equiv x^2 \bmod n, 1\le x\le n-1\right\}$. If n = 47, then its quadratic residue set is

$$Q_{47}=\{1,2,3,4,6,7,8,9,12,14,16,17,18,21,24,25,27,28,32,34,36,37,42\}.$$

Let the symbol C_{47} denote the binary (47, 24, 11) QR code. Let α be a root of primitive irreducible polynomial $\mathrm{p}(x)=1+x^5+x^{23}$ such that α is a generator of the multiplicative group of all nonzero elements in $GF(2^{23})$. Then, the element $\beta=\alpha^u$ where $u=(2^{23}-1)/47=178$, 481, is a primitive 47th root of unity in $GF(2^{23})$. The generator polynomial $g(x)$ of the binary (47, 24, 11) QR code is defined by

$$g(x)=\prod_{i\in Q_{47}}(x-\beta^i)=1+x+x^2+x^3+x^5+x^6+x^7+x^9+x^{10}+x^{12}+x^{13}+x^{14}+x^{18}+x^{19}+x^{23}$$

where the degree of $g(x)$ is 23, which is the multiplicative order of the integer 2 modulo the code length 47; that is, $2^{23}\equiv 1 \bmod 47$. The (47, 24, 11) QR code generated by this manner can correct up to five errors.

A codeword of binary QR code is a polynomial $c(x)=c_0+c_1x+c_2x^2+\cdots+c_{n-1}x^{n-1}$ such that it is a multiple of the generator polynomial $g(x)$. If the codeword $c(x)$ is transmitted through a noisy channel, then the received polynomial $r(x)=r_0+r_1x+r_2x^2+\cdots+r_{n-1}x^{n-1}$ can be expressed as the sum of the codeword polynomial $c(x)$ and the error polynomial $e(x)=e_0+e_1x+e_2x^2+\cdots+e_{n-1}x^{n-1}$. For simplicity, let the message or information, codeword, error pattern, received word, and syndrome be expressed as the binary vector forms $\mathrm{m}=(m_0,m_1,\cdots,m_{k-1})$, $\mathrm{c}=(c_0,c_1,\cdots,c_{n-1})$, $\mathrm{e}=(e_0,e_1,\cdots,e_{n-1})$, $\mathrm{r}=(r_0,r_1,\cdots,r_{n-1})$ and $\mathrm{s}=(s_0,s_1,\cdots,s_{n-k-1})$, respectively. The systematic codeword of the vector form is given by $\mathrm{c}=\mathrm{m}G$, where G is called the systematic generator matrix. Let A be a $k\times(n-k)$ matrix and I_k be a $k\times k$ identity matrix, and G can be expressed as $\mathrm{G}=\left[I_k \mid A\right]_{k\times n}$. The parity-check matrix H can be expressed as $H=\left[A^T \mid I_{n-k}\right]_{(n-k)\times n}$, where A^T denotes the $(n-k)\times k$ transpose matrix of A. The vector form of the syndrome is defined by $\mathrm{s}=\mathrm{r}H^T$, where H^T denotes the $n\times(n-k)$ transpose matrix of H; that is, H^T can be expressed as

$$H^T=\begin{bmatrix}A\\ I_{n-k}\end{bmatrix}=\begin{bmatrix}h_0\\ h_1\\ \vdots\\ h_{n-1}\end{bmatrix} \tag{1}$$

For C_{47}, A has the following form:

$$\begin{pmatrix}
1&1&1&1&0&1&1&1&0&1&1&0&1&1&1&0&0&0&1&1&0&0&0\\
0&1&1&1&1&0&1&1&1&0&1&1&0&1&1&1&0&0&0&1&1&0&0\\
0&0&1&1&1&1&0&1&1&1&0&1&1&0&1&1&1&0&0&0&1&1&0\\
0&0&0&1&1&1&1&0&1&1&1&0&1&1&0&1&1&1&0&0&0&1&1\\
1&1&1&1&1&0&0&0&0&0&0&1&1&0&0&0&1&1&0&1&0&0&1\\
1&0&0&0&1&0&1&1&0&1&1&0&0&0&1&0&0&1&0&1&1&0&0\\
0&1&0&0&0&1&0&1&1&0&1&1&0&0&0&1&0&0&1&0&1&1&0\\
0&0&1&0&0&0&1&0&1&1&0&1&1&0&0&0&1&0&0&1&0&1&1\\
1&1&1&0&0&1&1&0&0&0&0&0&0&0&1&0&0&1&1&1&1&0&1\\
1&0&0&0&0&1&0&0&0&1&1&0&1&1&1&1&0&0&0&0&1&1&0\\
0&1&0&0&0&0&1&0&0&0&1&1&0&1&1&1&1&0&0&0&0&1&1\\
1&1&0&1&0&1&1&0&0&1&1&1&0&1&0&1&1&1&1&1&0&0&1\\
1&0&0&1&1&1&0&0&0&1&0&1&0&1&0&0&1&1&0&0&1&0&0\\
0&1&0&0&1&1&1&0&0&0&1&0&1&0&1&0&0&1&1&0&0&1&0\\
0&0&1&0&0&1&1&1&0&0&0&1&0&1&0&1&0&0&1&1&0&0&1\\
1&1&1&0&0&1&0&0&1&1&1&0&0&1&0&0&1&0&1&0&1&0&0\\
0&1&1&1&0&0&1&0&0&1&1&1&0&0&1&0&0&1&0&1&0&1&0\\
0&0&1&1&1&0&0&1&0&0&1&1&1&0&0&1&0&0&1&0&1&0&1\\
1&1&1&0&1&0&1&1&1&1&1&1&1&0&0&1&0&1&0&1&0&0&1\\
0&1&1&1&0&1&0&1&1&1&1&1&1&0&0&1&0&1&0&1&0&0&1\\
1&1&0&0&1&1&0&1&1&0&0&1&0&0&1&0&1&0&0&1&1&0&0\\
0&1&1&0&0&1&1&0&1&1&0&0&1&0&0&1&0&1&0&0&1&1&0\\
0&0&1&1&0&0&1&1&0&1&1&0&0&1&0&0&1&0&1&0&0&1&1\\
1&1&1&0&1&1&1&0&1&1&0&1&1&1&0&0&0&1&1&0&0&0&1
\end{pmatrix}$$

Figure 1. *A of* C_{47}.

3. Decoding Algorithm and Theorems

Definition 1. The Hamming weight of a binary vector a is denoted by w(a), and the Hamming distance between a and b

is denoted by d(a, b) = w(a+b).

Theorem 1. Let $a=(a_0,a_1,\cdots,a_{n-1})$ and $b=(b_0,b_1,\cdots,b_{n-1})$ be two binary vectors, then

$$\mathrm{w(a+b)} = \mathrm{w(a)} + \mathrm{w(b)} - 2\sum_{i=0}^{n-1} a_i b_i. \tag{2}$$

Corollary 1. If $a_i b_i = 0$ for $0 \le i \le n-1$, then

$$\mathrm{w(a+b)} = \mathrm{w(a)} + \mathrm{w(b)}. \tag{3}$$

The following Theorem is useful to compute the syndrome of the received word when the received word shifts one bit to the right.

Theorem 2. Let $s(x)$ be the syndrome polynomial corresponding to a received polynomial $r(x)$. Also, let $r^{(1)}(x)$ be the polynomial obtained by cyclically shifting the coefficients of $r(x)$ one bit to the right. Then the remainder obtained when dividing $xs(x)$ by $g(x)$ is the syndrome $s^{(1)}(x)$ corresponding to $r^{(1)}(x)$.

For a detailed proof, see [2].

However, if the syndrome cyclically shifts many times, then the syndrome computation is quite time-consuming for dividing $xs(x)$ by $g(x)$ many times. The following theorem provides an efficient method to compute $s^{(i)}$ for $0 \le i \le n-1$, and it can save a lot of computational time.

Theorem 3. For the binary QR codes, let r_j be an element of r and h_j be the jth row vector of H^T for $0 \le j \le n-1$. Then the syndrome $s^{(i)}$ of $\mathrm{r}^{(i)}$ for $0 \le i \le n-1$ has the form

$$s^{(i)} = \sum_{j=0}^{n-1} r_j h_{[i+j]}, \tag{4}$$

where the suffix $[x]$ of h denotes x mod n.

Proof. Let $\mathrm{r}=(r_0,r_1,\cdots,r_{n-1})$ and $\mathrm{r}^{(i)}=(r_{n-i},\cdots,r_{n-1},r_0,\cdots,r_{n-i-1})$ for $0 \le i \le n-1$. We have

$$\begin{aligned}\mathrm{s}^{(i)} &= r_{n-i}h_0 + \cdots + r_{n-1}h_{i-1} + r_0 h_i + \cdots + r_{n-i-1}h_{n-1}\\ &= r_{n-i}h_{[n-i+i]} + \cdots + r_{n-1}h_{[n-1+i]} + r_0 h_{[0+i]} + \cdots + r_{n-i-1}h_{[n-i-1+i]}\\ &= \sum_{j=0}^{n-1} r_j h_{[i+j]}\end{aligned}$$

The proof is thus completed.

Theorem 3 reveals that the syndrome of $\mathrm{r}^{(i)}$ can be fast computed by the vector addition. Theorem 4 also provides an efficient method to simplify the decoding step by using the syndrome weight.

Theorem 4. For the binary QR codes, it is assumed that there are v errors in the received word, where $1 \le v \le t$ and $t=\lfloor (d-1)/2 \rfloor$. All v errors are in the parity-check bits if and only if the weight of syndrome $w(s)=v$.

For a detailed proof, see [4].

Theorem 5. For the binary QR codes, if v errors are in the information bits of the received word, where $1 \le v \le t$ and $t=\lfloor (d-1)/2 \rfloor$, then the weight of the corresponding syndrome vector satisfies

$$w(s) \ge d - v \ge t+1 \tag{5}$$

For a detailed proof, see [5].

Theorem 6. For the binary QR codes, let $\mathrm{e}=(e_0,e_1,\cdots,e_{n-1})$ be an error pattern and $\mathrm{e_m}=(e_0,\cdots,e_{k-1})$, $\mathrm{e_p}=(e_k\cdots,e_{n-1})$ be respectively its message section and parity check section. Assume that $w(\mathrm{e}_m) \ge 1$, $w(\mathrm{e}_p) \ge 1$ and $w(\mathrm{e}) \le t$, where $t=\lfloor (d-1)/2 \rfloor$, then the weight of the corresponding syndrome vector satisfies

$$w(s) \ge t+1. \tag{6}$$

Proof. As $\mathrm{w(s)} = w(\mathrm{e}H^T)$

$$\begin{aligned}&= w\left(\left[\mathrm{e}_m, 0_{1\times(n-k)}\right]H^T + \left[0_{1\times k}, \mathrm{e}_p\right]H^T\right)\\ &\ge \left(d - w(\mathrm{e}_m)\right) - w(\mathrm{e}_p)\\ &= d - \left(w(\mathrm{e}_m) + w(\mathrm{e}_p)\right)\\ &\ge d - t \ge t+1,\end{aligned}$$

the proof is thus completed.

Given a received word r, the syndrome $s^{(i)}$ of $\mathrm{r}^{(i)}$ can be fast computed by theorem 3. According to theorem 4, if $1 \le w(s) \le 5$ then error positions are in the parity-check bits of r. If $1 \le w(s^{(k)}) \le 5$, then the error positions are in the information bits of r. Let h_j denote the jth row vector of H^T, where $0 \le j \le n-1$. Also let sd_z denote the syndrome difference between the syndromes of r and h_j in each decoding step z. By using the weight of sd_z, the error cases can be quickly determined. Let $u_0=(1,0,\cdots,0)$ be a k-tuples unit vector and u_i has only one nonzero component at the i th position, where $0 \le i \le k-1$. By using these properties the decoding algorithm can be constructed. Let case I, C, P denote the error position in the information bits, center bit, and parity-check bits of r, respectively. The decoding steps of the proposed NESWDA work as follows:

(1) (No error, P, PP, PPP, PPPP, and PPPPP cases) By theorem 3, compute s and $w(s)$. If $0 \le w(s) \le 5$, then the information vector is $\mathrm{m}=(r_0,r_1,\cdots,r_{k-1})$. Go to step (8).

(2) (I, II, III, IIII, and IIIII cases) By theorem 3, compute $s^{(k)}$ and $w(s^{(k)})$. If $1 \le w(s^{(k)}) \le 5$, then the corrected

information vector is $\mathrm{m}=(r_0,r_1,\cdots,r_{k-1})+(s^{(k)},0)$. Go to step (8).

(3) (IP, IPP, IPPP, IPPPP, C, CP, CPP, CPPP, and CPPPP cases) Compute the syndrome difference $sd_3=s-h_i$ for $0\le i\le k-1$ and $w(sd_3)$. If $w(sd_3)\le 4$, then the corrected information vector $\mathrm{m}=(r_0,r_1,\cdots,r_{k-1})+u_i$. Go to step (8).

(4) (IC, IIC, IIIC, IIIIC, IIP, IIIP, and IIIIP cases) Compute the syndrome difference $sd_4=s^{(k)}-h_i$ for $0\le i\le k-1$ and $w(sd_4)$. If $w(sd_4)\le 4$ and $i=0$, then the corrected information vector $\mathrm{m}=(r_0,r_1,\cdots,r_{k-1})+u_{k-1}+(sd_4,0)$. If $w(sd_4)\le 4$ and $0\le i\le k-1$, then the corrected information vector $\mathrm{m}=(r_0,r_1,\cdots,r_{k-1})+(sd_4,0)$. Go to step (8).

Table 1. *The number of error patterns in each decoding step of C_{47}.*

Steps	Cases	Number of error patterns
1	P, PP, PPP, PPPP, PPPPP	44551
2	I, II, III, IIII, IIIII	44551
3	IP,IPP,IPPP,IPPPP,C,CP,CPP,CPPP,CPPPP	261649
4	IC, IIC, IIIC, IIIIC, IIP, IIIP, IIIIP	261119
5	IIPP, IIPPP, ICP, ICPP, ICPPP	559153
6	IICP, IIICP, IIIPP	494615
7	IICPP	64009
Sum	35	$\sum_{i=1}^{5}\binom{47}{i}=1729647$

(5) (IIPP, IIPPP, ICP, ICPP, and ICPPP cases) Compute the syndrome difference $sd_5=s-h_i-h_j$ for $0\le i<j\le k-1$ and $w(sd_5)$. If $w(sd_5)\le 3$, then the corrected information vector $\mathrm{m}=(r_0,r_1,\cdots,r_{k-1})+u_i+u_j$. Go to step (8).

(6) (IICP, IIICP, and IIIPP cases) Compute the syndrome difference $sd_6=s^{(k)}-h_i-h_j$ for $0\le i<j\le k-1$ and $w(sd_6)$. If $w(sd_6)\le 3$ and $i=0$, then the corrected information vector $\mathrm{m}=(r_0,r_1,\cdots,r_{k-1})+u_{k-1}+(sd_6,0)$. If $w(sd_6)\le 3$ and $i>0$, then the corrected information vector $\mathrm{m}=(r_0,r_1,\cdots,r_{k-1})+(sd_6,0)$. Go to step (8).

(7) (IICPP case) Compute the syndrome difference $sd_7=s-h_{k-1}-h_i-h_j$ for $0\le i<j\le k-2$ and $w(sd_7)$. If $w(sd_7)\le 2$, then the corrected information vector $\mathrm{m}=(r_0,r_1,\cdots,r_{k-1})+u_i+u_j+u_{k-1}$. Go to step (8).

(8) Stop.

Table 1 lists all the 35 error cases and the number of error patterns in each decoding step of C_{47} of the proposed NESWDA. The flowchart of the proposed NESWDA is shown in figure 1.

4. Example

In this section, two examples are presented to illustrate the proposed NESWDA.

Example 1. Let a information m = (0100000000000000000000000) be encoded into a C_{47} codeword

c = (01000000000000000000000001111011101101110001100).

If the received word

r = (11110000000000000000000001111011101101110001100),

then the error pattern

e = (10110000000000000000000000000000000000000000000),

which means a III error case. The decoding procedure is composed of the following steps.

(1) Compute s = (11010100010110000111101) and w(s) = 12. Since w(s) > 5, go to step 2.

(2) Compute s^{24} = (10110000000000000000000) and w(s^{24}) = 3 ≤ 5. The corrected information vector is

m = (11110000000000000000000) + (10110000000000000000000)

= (01000000000000000000000).

Go to stop.

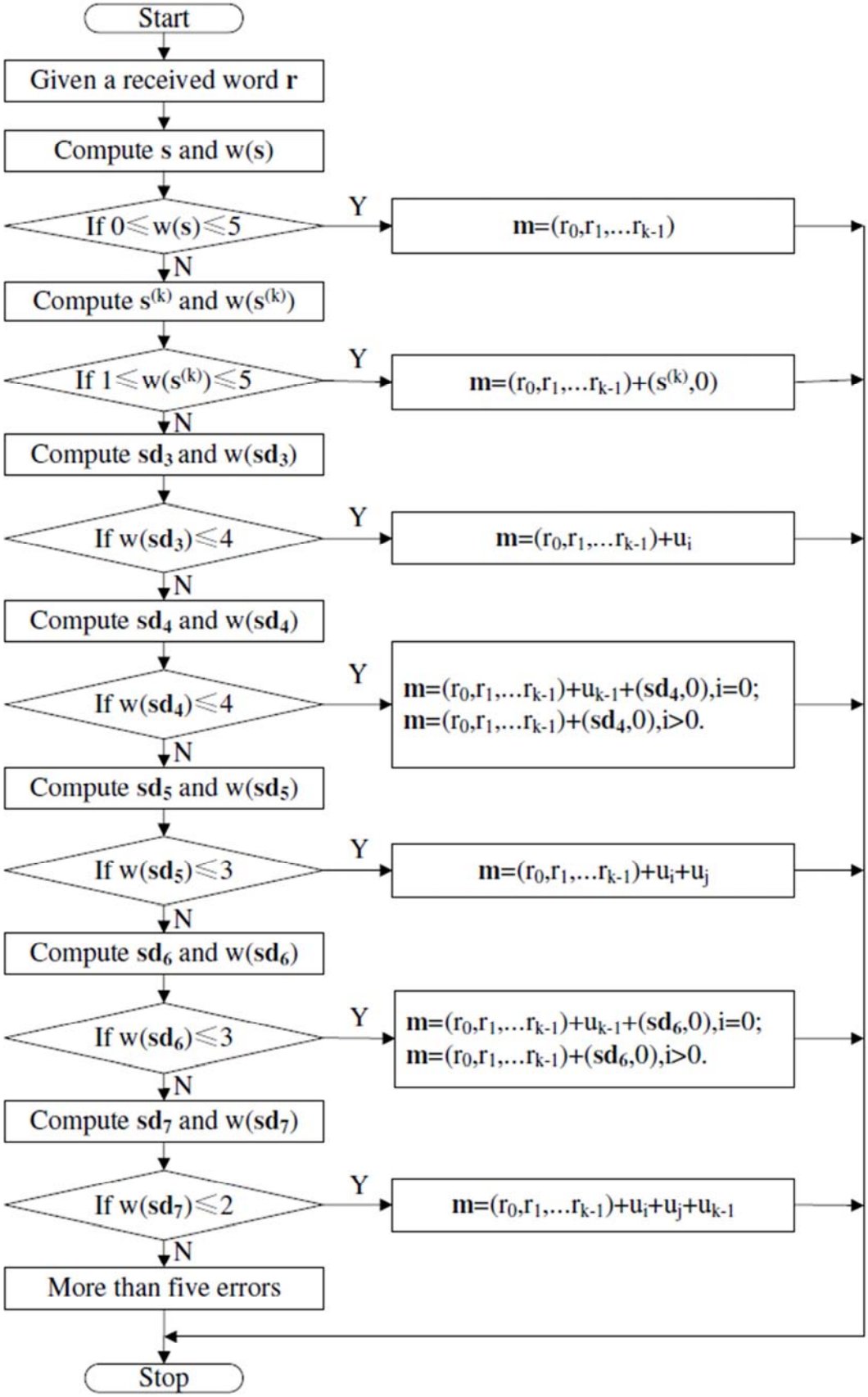

Figure 2. *Flowchart of the proposed NESWDA.*

Example 2. Let a information m = (010000000000000000000000) be encoded into a C_{47} codeword

c = (010000000000000000000000011110111011011101110001100).

If the received word

r = (011100000000000000000000100111011101101110001100),

then the error pattern

e = (001100000000000000000000101000000000000000000000),

which means a IICP case. The decoding procedure is composed of the following steps.

(1) Compute s = (100001011110101000001000) and w(s) = 9 > 5, go to step 2.

(2) Compute s^{24} = (111110101011010111011110) and w(s^{24}) = 16 > 5, go to step 3.

(3) Compute $sd_3 = s - h_i$ for $0 \le i \le 23$ and w(sd_3).

$sd_3 = s - h_0$ = (100001011110101000001000) − (111101110110111000110000)

= (011100101000010000111000),

w(sd_3) = 9.

$sd_3 = s - h_1$ = (100001011110101000001000) − (011110111011011100011000)

= (111111100101110100010000),

w(sd_3) = 13:

…

$sd_3 = s - h_9$ = (100001011110101000001000) − (100001000110111100001100)

= (000000011000010100000010),

w(sd_3) = 5.

…

$sd_3 = s - h_{23}$ = (100001011110101000001000) − (111001101101110001000010)

= (011000110011011001001010),

w(sd_3) = 11.

Since every w(sd_3) > 4, go to step 4.

(4) Compute $sd_4 = s^{24} - h_i$ for $0 \le i \le 23$ and w(sd_4).

$sd_4 = s^{24} - h_0$ = (111110101011010111011110) − (111101110110111000110000)

= (000011011101101110001100),

w(sd_4) = 12.

$sd_4 = s^{24} - h_1$ = (111110101011010111011110) − (011110111011011100011000)

= (100000010000001010100010),

w(sd_4) = 6.

…

$sd_4 = s^{24} - h_{23}$ = (111110101011010111011110) − (111001101101110001000010)

= (000111000110100111111111),

w(sd_4) = 14.

Since every w(sd_4) > 4, go to step 5.

(5) Compute $sd_5 = s - h_i - h_j$ for $0 \le i < j \le 23$ and w(sd_5).

$sd_5 = s - h_0 - h_1$ = (000010010011001100100000), w(sd_5) = 6.

$sd_5 = s - h_0 - h_2$ = (010011110101111110110100), w(sd_5) = 15.

…

$sd_5 = s - h_0 - h_{23}$ = (100101000101100001111010), w(sd_5) = 11.

$sd_5 = s - h_1 - h_2$ = (110000111000011010011100), w(sd_5) = 11.

…

$sd_5 = s - h_1 - h_{23}$ = (000110001000000101010010), w(sd_5) = 7.

…

$sd_5 = s - h_{22} - h_{23}$ = (010100000101001011101100), w(sd_5) = 10.

Since every w(sd_5) > 3, go to step 6.

(6) Compute $sd_6 = s^{24} - h_i - h_j$ for $0 \le i < j \le 23$ and w(sd_6).

$sd_6 = s^{24} - h_0 - h_1$ = (011101100110110010010100), w(sd_6) = 12.

$sd_6 = s^{24} - h_0 - h_2$ = (001100000000000000000000), w(sd_6) = 2.

Since w(sd_6) = 2 ≤ 3, and $i = 0$. The corrected information

$$\mathrm{m} = (r_0, r_1, \cdots, r_{23}) + u_{23} + (sd_6, 0) =$$
$$(01110000000000000000000001) +$$
$$(00000000000000000000000001) +$$
$$(00110000000000000000000000)$$
$$= (01000000000000000000000000).$$

Go to stop.

5. Conclusions

A new NESWDA decoding algorithm is developed to correct up to five errors for the approximate half-rate (47, 24, 11) QR code. The main idea behind the proposed NESWDA is based on the fact that it makes use of the properties of cyclic codes, the weight of syndrome difference. The proposed NESWDA has no need of a look-up table to store the syndromes and their corresponding error patterns in the memory. Moreover, it can be extended to decode all five-error-correcting binary QR codes.

References

[1] E. Prange, Some Cyclic Error-Correcting Codes with Simple Decoding Algorithms, Air Force Cambridge Research Center, Cambridge, MA, 1958 (TN-58-156).

[2] S. B. Wicker, Error Control Systems for Digital Communication and Storage, Prentice Hall, 1995.

[3] Y. H. Chen, T. K. Truong, C. H. Huang, C. H. Chien, A lookup table decoding of systematic (47, 24, 11)quadratic residue code, Information Science 179 (2009) 2470-2477.

[4] T. C. Lin, H. P. Lee, H. C. Chang, S. I. Chu, T. K. Truong, High speed decoding of the binary (47, 24, 11)quadratic residue code, Information Science 180 (2010) 4060-4068.

[5] T. C. Lin, H. P. Lee, H. C. Chang, T. K. Truong, A cyclic weight algorithm of decoding (47, 24, 11) quadratic residue code, Information Science 197 (2012) 215-222.

[6] G. Dubney, I. S. Reed, T. K. Truong, J. Yang, Decoding the (47, 24, 11) quadratic residue code using bit-error probability estimates, IEEE Transactions on Communications 57 (2009).

[7] R. He, I. S. Reed, T. K. Truong, X. Chen, Decoding the (47, 24, 11) quadratic residue code, IEEE Transactions on Information Theory 47 (2001) 1181-1186.

[8] T. C. Lin, T. K. Truong, and H. P. Lee, Algebraic decoding of the (41, 21, 9) quadratic residue code, Information Sciences 179 (2009) 3451-3459.

[9] X. Chen, I. S. Reed, T. Helleseth, and T. K. Truong, Use of Grobner bases to decode binary cyclic codes up to the true minimum distance, IEEE Transactions on Information Theory 40 (1994) 1654-1661.

[10] Y. H. Chen, T. K. Truong, Y. Chang, C. D. Lee, S. H. Chen, Algebraic decoding of Quadratic Residue codes using Berlekamp–Massey algorithm, Journal of Information Science and Engineering 23 (2007) 127-145.

[11] I. S. Reed, X. Chen, Error-Control Coding for Data Networks, Kluwer, Boston, MA, 1999.

[12] I. S. Reed, M. T. Shih, T. K. Truong, VLSI design of inverse-free Berlekamp-Massey algorithm, Processings IEE 138 (1991) 295-298.

[13] I. S. Reed, T. K. Truong, X. Chen, X. Yin, The algebraic decoding of the (41, 21, 9) Quadratic Residue code, IEEE Transactions on Information Theory 38(1992) 974-986.

[14] I. S. Reed, X. Yin, T. K. Truong, Algebraic decoding of the (32, 16, 8) Quadratic Residue code, IEEE Transactions on Information Theory 36 (1990) 876-880.

[15] I. S. Reed, X. Yin, T. K. Truong, J. K. Holmes, Decoding the (24, 12, 8) Golay code, IEE Proceedings 137 (3) (1990) 202-206.