Designs, Codes and Cryptography (2025) 93:3575–3591
https://doi.org/10.1007/s10623-025-01650-8

Entanglement-assisted quantum error-correcting codes via quasi-cyclic codes with complementary duals

Hajime Matsui[1] · Kakeru Kaneko[2]

Received: 15 July 2024 / Revised: 6 January 2025 / Accepted: 7 May 2025 /
Published online: 31 May 2025

Abstract

It is known that entanglement-assisted quantum error-correcting codes (EAQECCs), a type of quantum error correction codes, can be easily constructed using linear codes that satisfy the property called linear complementary duals (LCD). For quasi-cyclic (QC) codes, which are a class of linear codes, we have already published the methods for constructing the codes with properties such as self-orthogonality, self-duality and reversibility according to the prime-factor decomposition of $-1 + x^m$, which reduce the amount of calculation by assembling several small generator polynomial matrices into a large generator polynomial matrix. In this paper, we propose a method to construct LCD–QC codes according to the prime-factor decomposition. The main idea of this method is to decompose the generator polynomial matrix of the QC code into several small generator polynomial matrices corresponding to the prime factors and perform LCD determination, which leads to a reduction in the amount of calculation. As an application of our construction method, we create EAQECCs from the constructed LCD–QC codes, compare those minimum weights with the maximum values of the minimum weights of existing EAQECCs and find 15 EAQECCs with larger minimum weights than the existing ones.

Keywords Generator polynomial matrix · Chinese remainder theorem · Elementary row operation

Mathematics Subject Classification 94B05 · 81P73 · 11H06 · 11T71

Communicated by J.-L. Kim.

✉ Hajime Matsui
matsui@sci.kagoshima-u.ac.jp

Kakeru Kaneko
k.kaneko2810@gmail.com

[1] Mathematics and Informatics Program, Faculty of Science, Kagoshima University, 1-21-35 Korimoto, Kagoshima 890-0065, Japan

[2] Graduate School of Engineering, Toyota Technological Institute, 2-12-1 Hisakata, Tempaku, Nagoya 468-8511, Japan

1 Introduction

Quantum computers have attracted much attention because powerful algorithms such as Shor's and Grover's ones have been known. Quantum error-correcting codes are essential in the realization of them [19]. Calderbank–Shor–Steane (CSS) codes [2], a class of quantum error-correcting codes, can be easily constructed from classical error-correcting codes and are actively researched. On the other hand, entanglement-assisted quantum error-correcting codes (EAQECCs), another class of quantum error-correcting codes, offer various advantages by sharing quantum entanglement between transmitters and receivers. For example, more quantum states can be encoded and the error-correcting capability can be improved [1, 3]. EAQECCs can be constructed from a certain class of classical error-correcting codes as well as CSS codes as follows [20]. For a linear code $\mathcal{C}$ over $\mathbb{F}_q$ and its dual code ${}^{\perp}\mathcal{C}$, if $\mathcal{C} \cap {}^{\perp}\mathcal{C} = \{0\}$, then we say that $\mathcal{C}$ is a linear complementary dual (LCD) code or LCD. For a generator matrix $\mathcal{G}$ of $\mathcal{C}$, $\mathcal{C}$ is LCD if and only if $\det\left(\mathcal{G}\left(\mathcal{G}^{\top}\right)\right) \neq 0$ [11], where $\mathcal{G}^{\top}$ is the transpose of $\mathcal{G}$. From LCD codes, we can construct optimal EAQECCs in a sense, called maximal-entanglement EAQECCs [9]. On the other hand, it is known that, among linear codes, quasi-cyclic (QC) codes contain asymptotically good codes [6]. Then algorithms through mutually prime factor decomposition and Chinese Remainder Theorem (CRT) are proposed as efficient construction methods for self-dual QC codes [8] and for LCD codes with quasi-cyclicity (LCD–QC codes) [4], respectively. While we have mentioned the construction methods using the generator matrix $\mathcal{G}$ so far, in this paper we deal with the ones using the generator polynomial matrix G. For $R = \mathbb{F}_q[x]$, if an l-by-l square polynomial matrix G with entries in R satisfies $AG = fI$ for some l-by-l square polynomial matrix A with entries in R, where $f \in R$ divides $-1 + x^m \in R$ and I is the l-by-l identity matrix, then we say that G is a generator polynomial matrix. In the case of $f = -1 + x^m$, for a generator polynomial matrix G, $\mathcal{C} = \mathbb{L}G/(-1 + x^m)\mathbb{L}$ is a QC code of code length $n = ml$ and conversely, for a QC code $\mathcal{C}$, there exists a generator polynomial matrix G such that $\mathcal{C} = \mathbb{L}G/(-1 + x^m)\mathbb{L}$ [7], where $\mathbb{L} = R^l$ is an R-module of row vectors of length l with entries in R. One of the authors proposes construction methods for various types of QC codes with generator polynomial matrices, which reduce the matrix size compared to the generator matrices of linear codes and the computational complexities of algorithms [12, 14]. In [14], a construction method for self-orthogonal or self-dual QC codes with generator polynomial matrices G through mutually prime factor decomposition and CRT is shown, i.e., one shows that G corresponds to certain generator polynomial matrices $G_1, \ldots, G_t$ satisfying $A_i G_i = f_i I$, where $f_1, \ldots, f_t \in R$, $-1 + x^m = f_1 \cdots f_t$ and $\gcd(f_i, f_j) = 1$ with $1 \leq i < j \leq t$, cf. later Algorithm 1. In [16], we show construction methods for self-orthogonal or self-dual and/or reversible QC codes with generator polynomial matrices through mutually prime factor decomposition and CRT. Furthermore, in [17] we show a decision method for LCD–QC codes with generator polynomial matrices, i.e., a QC code $\mathcal{C}$ is LCD if and only if $\mathbb{L}G + \mathbb{L}H = \mathbb{L}$ as an R-module equation, where H is a generator polynomial matrix of ${}^{\perp}\mathcal{C}$. However, the construction of LCD–QC codes with generator polynomial matrices through mutually prime factor decomposition and CRT has not been studied.

In this paper, we propose a decision method for LCD–QC codes by mutually prime factor decomposition with generator polynomial matrices. Let $f^* = x^{\deg(f)} f(x^{-1}) \in R$ be the reciprocal polynomial of $f \in R$. For $f_i \in R$, if $f_i = \alpha f_i^*$ with some $0 \neq \alpha \in \mathbb{F}_q$, then f_i is called to be self-reciprocal, and for $f_i \neq f_j \in R$, if $f_i = \alpha f_j^*$ with some $0 \neq \alpha \in \mathbb{F}_q$, then f_i, f_j are called to be mutually reciprocal. Our decision method for LCD–QC codes replaces the condition of G with those of $G_1, \ldots, G_t$, i.e., a QC code $\mathcal{C}$ is

Table 1 Comparison of computational complexities among three decision methods for LCD–QC codes

[11]	[17]		Ours	
	–	With [15]	–	With [15]
$O(l^3m^3)$	$O(l^2m^3)$	$O(l^2m\log^2(m))$	$O\big(l^2\sum_{i=1}^{t}m_i^3\big)$	$O\big(l^2\sum_{i=1}^{t}m_i\log^2(m_i)\big)$

LCD if and only if $\mathbb{L}G_i + \mathbb{L}H_i = \mathbb{L}$ when f_i is self-reciprocal and $\mathbb{L}G_i + \mathbb{L}H_j = \mathbb{L}$ and $\mathbb{L}G_j + \mathbb{L}H_i = \mathbb{L}$ when f_i, f_j are mutually reciprocal, cf. later Theorem 1. Our decision method is effectively incorporated into the construction method and reduces its computational complexity compared to the methods [11, 17], cf. later Table 1. As numerical experiments, we construct LCD–QC codes over $\mathbb{F}_2$ up to $6 \le n \le 39$, $3 \le m \le 16$ or $2 \le l \le 6$ and over $\mathbb{F}_3$ up to $4 \le n \le 28$, $2 \le m \le 14$ or $2 \le l \le 6$, and find EAQECCs made from these LCD–QC codes with the largest minimum weights. We compare those of our EAQECCs with the existing ones in [3]. We find 15 LCD–QC codes that provide EAQECCs with higher error-correcting capability than those in [3].

The rest of this paper is organized as follows. In Sect. 2, we define the codes $\mathcal{C} = \mathbb{L}G/f\mathbb{L}$ and describe constructing QC codes through mutually prime factor decomposition of $-1+x^m$. In Sect. 3, as a decision method for LCD–QC codes, we explain the necessary and sufficient conditions for $G_1, \ldots, G_t$ where G is a generator polynomial matrix of an LCD–QC code. In Sect. 4, we construct LCD–QC codes over $\mathbb{F}_2$ and $\mathbb{F}_3$ of fixed m, l and demonstrate searching for EAQECCs with larger minimum weights than the existing ones.

2 Preliminaries

Let $R = \mathbb{F}_q[x]$ be the ring of one-variable polynomials with coefficients in q-element finite field $\mathbb{F}_q$.

2.1 Generator polynomial matrices, *R*-modules and QC codes [13, 14, 16]

Let l be a positive integer and

$$M_l(R) = \left\{ \begin{pmatrix} b_{1,1} & \cdots & b_{1,l} \\ \vdots & & \vdots \\ b_{l,1} & \cdots & b_{l,l} \end{pmatrix} \;\middle|\; \begin{array}{l} b_{i,j} \in R, \\ 1 \le i \le l, \\ 1 \le j \le l \end{array} \right\}.$$

Let $I \in M_l(R)$ be the identity matrix. Let m be a positive integer and $f \in R\setminus F_q$ divide $-1 + x^m \in R$. For $G \in M_l(R)$, if there exists $A \in M_l(R)$ with $AG = fI$, then we say that G is a *generator polynomial matrix*. We can convert G to a generator matrix $\mathcal{G}$ of a linear code of code-length $n = ml$ over $\mathbb{F}_q$, cf. later Example 1.

Let $GL_l(R) = \{\gamma \in M_l(R) \big| \det(\gamma) \in \mathbb{F}_q \setminus \{0\}\}$. Among γG with $\gamma \in GL_l(R)$, we can choose $G = (g_{i,j}) \in M_l(R)$ satisfying the following three conditions.

- G is an upper-triangular matrix such as

$$G = \begin{pmatrix} g_{1,1} & g_{1.2} & \cdots & g_{1,l} \\ 0 & g_{2,2} & \cdots & g_{2,l} \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & g_{l,l} \end{pmatrix}.$$

- $g_{i,i}$ is monic, i.e., the coefficient of the highest degree monomial is equal to one.
- $\deg(g_{i,j}) < \deg(g_{j,j})$ for all $1 \le i < j \le l$.

If a generator polynomial matrix G satisfies the above three conditions, then we say that G is *reduced*.

Let

$$\begin{aligned}\mathbb{L} &= R^l = \{a = (a_1, \dots, a_l) \mid a_i \in R,\ 1 \le i \le l\},\\ \mathbb{L}/f\mathbb{L} &= (R/fR)^l = \{b = (b_1, \dots, b_l) \mid b_i \in R/fR,\ 1 \le i \le l\},\end{aligned}$$

where R/fR is the quotient ring of R by fR. For a subset $\mathcal{C} \subset \mathbb{L}/f\mathbb{L}$, if there exists a generator polynomial matrix G such that

$$\mathcal{C} = \mathbb{L}G/f\mathbb{L} = \{c \in \mathbb{L}/f\mathbb{L} \mid c = uG,\ u \in \mathbb{L}/f\mathbb{L}\},$$

then we say that $\mathcal{C}$ is an *R-module of* $\mathbb{L}/f\mathbb{L}$. If $f = -1 + x^m$, then an R-module $\mathcal{C}$ of $\mathbb{L}/f\mathbb{L}$ agrees with a quasi-cyclic (QC) code over $\mathbb{F}_q$ [7, 8]. For a general f, R-modules of $\mathbb{L}/f\mathbb{L}$ are linear codes over $\mathbb{F}_q$, but not necessarily QC codes.

2.2 Dual R-modules and QC codes

For a subset ${}^{\perp}\mathcal{C} \subset \mathbb{L}/f^*\mathbb{L}$, if there exists an R-module $\mathcal{C} \subset \mathbb{L}/f\mathbb{L}$ such that

$${}^{\perp}\mathcal{C} = \left\{ b = (b_1, \dots, b_l) \in \mathbb{L}/f^*\mathbb{L} \;\middle|\; \begin{array}{l} \sum_{i=1}^{l} b_i \left(c_i^{\langle m \rangle}\right) \equiv 0 \bmod f^*, \\ \forall c = (c_1, \dots, c_l) \in \mathcal{C} \end{array} \right\},$$

where for $1 \le i \le l$ and $c_i = \sum_{j=0}^{m-1} c_{i,j}x^j \in R$, $c_i^{\langle m \rangle} = c_{i,0} + \sum_{j=1}^{m-1} c_{i,m-j}x^j \in R$, then we say that ${}^{\perp}\mathcal{C}$ is the *dual* R-module of $\mathcal{C}$. If $\mathcal{C}$ is a linear code of code-length n and dimension k as a linear space over $\mathbb{F}_q$, then ${}^{\perp}\mathcal{C}$ is a linear code of code-length n and dimension $n - k$. If $f = -1 + x^m$, then the dual R-module ${}^{\perp}\mathcal{C}$ of $\mathcal{C}$ agrees with the dual linear code of a QC code $\mathcal{C}$ over $\mathbb{F}_q$. For an irreducible f, the dual R-module ${}^{\perp}\mathcal{C}$ of $\mathcal{C}$ agrees with the Hermitian dual linear code of $\mathcal{C}$ over $\mathbb{F}_q$. For $A = (a_{i,j})$ with $AG = fI$, let $H \in M_l(R)$ be

$$H = \operatorname{diag}\left[x^{m+\deg(a_{i,i})}\right] A^{\top}(x^{-1}) + (-1 + x^m) \operatorname{diag}\left[a_{i,i}^*\right], \tag{1}$$

where, for $c_1, \dots, c_l \in R$, $\operatorname{diag}[c_i] \in M_l(R)$ is the diagonal matrix whose ith entry is equal to c_i for all $1 \le i \le l$. Then H is a generator polynomial matrix of ${}^{\perp}\mathcal{C}$ and $BH = f^*I$ for some $B \in M_l(R)$.

2.3 LCD codes, LCD R-modules and LCD–QC codes

Let $\mathcal{C}$ be a linear code over $\mathbb{F}_q$. If $\mathcal{C} \cap {}^{\perp}\mathcal{C} = \{0\}$, where ${}^{\perp}\mathcal{C}$ is the dual linear code of $\mathcal{C}$, then we say that $\mathcal{C}$ is a *linear complementary dual code* or an *LCD code*. Note that $\mathcal{C} \cap {}^{\perp}\mathcal{C} = \{0\}$ if and only if $\mathcal{C} + {}^{\perp}\mathcal{C} = (\mathbb{F}_q)^n$ because $\dim\left(\mathcal{C} + {}^{\perp}\mathcal{C}\right) = \dim \mathcal{C} + \dim {}^{\perp}\mathcal{C} - \dim\left(\mathcal{C} \cap {}^{\perp}\mathcal{C}\right)$. Note also [11] that $\mathcal{C} \cap {}^{\perp}\mathcal{C} = \{0\}$ if and only if $\det\left(\mathcal{G}\left(\mathcal{G}^{\top}\right)\right) \neq 0$ if and only if $\det\left(\mathcal{H}\left(\mathcal{H}^{\top}\right)\right) \neq 0$, where $\mathcal{G}$ and $\mathcal{H}$ are generator and parity check matrices of $\mathcal{C}$, respectively.

Let $f \in R \setminus \mathbb{F}_q$ divide $-1 + x^m$ and be self-reciprocal and $\mathcal{C}$ be an R-module of $\mathbb{L}/f\mathbb{L}$. If $\mathcal{C} \cap {}^{\perp}\mathcal{C} = \{0\}$, where ${}^{\perp}\mathcal{C}$ is the dual R-module of $\mathcal{C}$, then we say that $\mathcal{C}$ is an *LCD R-module*. It is known [17] that there are the other equivalent conditions, where '$\Longleftrightarrow$' means 'if and only if', H is given by (1) and O is a zero matrix of appropriate size, as

$$\mathcal{C} \cap {}^{\perp}\mathcal{C} = \{0\} \iff \mathbb{L}G + \mathbb{L}H = \mathbb{L} \iff N \begin{pmatrix} G \\ H \end{pmatrix} = \begin{pmatrix} I \\ O \end{pmatrix} \text{ for some } N \in GL_{2l}(R). \tag{2}$$

If $f = -1 + x^m$, then we call an LCD R-module $\mathcal{C}$ an *LCD–QC code*.

2.4 Constructing EAQECCs via LCD codes

In this paper, an $[[n, k, d; c]]_q$ entanglement-assisted quantum error-correcting code, where we say an $[[n, k, d; c]]_q$ EAQECC or an $[[n, k, d; c]]_q$ code for short, encodes k qubits into n qubits with the help of c copies of maximally entangled states, i.e., c ebits, over $\mathbb{F}_q$ and d indicates the minimum distance of the code [1, 9, 10]. From now on, if $q = 2$, then q is omitted as $[[n, k, d; c]]$. It is known that EAQECC can be constructed using two classical error-correcting codes as follows.

Lemma 1 (Cf. [20]) *Let $\mathcal{H}_1$ and $\mathcal{H}_2$ be parity check matrices of $[n, k_1, d_1]_q$ and $[n, k_2, d_2]_q$ linear codes, respectively. Then an $[[n, k_1+k_2-n+c, \min\{d_1, d_2\}; c]]_q$ EAQECC is obtained, where $c = rank(\mathcal{H}_1\mathcal{H}_2^{\top})$.*

Remark 1 Let $\mathcal{C}$ be an $[n, k, d]_q$ LCD code with a parity check matrix $\mathcal{H}$. In Lemma 1, if $\mathcal{H}_1 = \mathcal{H}_2 = \mathcal{H}$, then an $[[n, k, d; n-k]]_q$ EAQECC is obtained. If an $[[n, k, d; c]]_q$ EAQECC $\mathcal{Q}$ satisfies $c = n - k$, then we say that $\mathcal{Q}$ is *maximal-entanglement*. The family of maximal-entanglement EAQECCs contains the ones that have asymptotically good parameters [18].

2.5 Constructing QC codes via mutually prime factor decomposition

Let $f, f_1, \ldots, f_t \in R\backslash\mathbb{F}_q$ divide $-1+x^m$, $f = f_1 \cdots f_t$ and $\gcd(f_i, f_j) = 1$ for all $1 \leq i < j \leq t$. For $i = 1, \ldots, t$, we define $e_i \in R$ by $e_i \equiv (f/f_i)b_i \mod f$ and $\deg(e_i) < \deg(f)$, where $b_i \in R$ satisfies $(f/f_i)b_i \equiv 1 \mod f_i$. We call $e_1, \ldots, e_t$ *idempotent elements*. They satisfy the following three conditions [5].

(i) $e_i e_i \equiv e_i \mod f$,
(ii) $e_i e_j \equiv 0 \mod f$ if $i \neq j$,
(iii) $e_1 + \cdots + e_t = 1$.

We quote Algorithm 1 for converting $G_1, \ldots, G_t$ to G and Algorithm 2 for converting G to $G_1, \ldots, G_t$ [14, 17].

Algorithm 1 Converting $G_1, \ldots, G_t$ to G via mutually prime factor decomposition

```
Input: $G_1, \cdots, G_t \in M_l(R)$ with $A_1G_1 = f_1I, \ldots, A_tG_t = f_tI$ where $\gcd(f_i, f_j) = 1, 1 \leq i < j \leq t$.
Output: $G \in M_l(R)$ with $AG = fI = f_1 \cdots f_tI$.
  for $i = 1$ to $t$ do
    $b_i$ with $(f/f_i)b_i \equiv 1 \mod f_i$
    $e_i \equiv (f/f_i)b_i \mod f$ and $\deg(e_i) < \deg(f)$
  end for
  $N \begin{pmatrix} e_1G_1 \\ \vdots \\ e_tG_t \end{pmatrix} = \begin{pmatrix} G \\ O \end{pmatrix}$ for some $N \in GL_{tl}(R)$
```

```
Algorithm 2 Converting $G$ to $G_1, \dots, G_t$ via mutually prime factor decomposition
Input: $G \in M_l(R)$ with $AG = fI = f_1 \cdots f_t I$ where $\gcd(f_i, f_j) = 1, 1 \le i < j \le t$.
Output: $G_1, \dots, G_t \in M_l(R)$ with $A_1 G_1 = f_1 I, \dots, A_t G_t = f_t I$.
    $N_1 \begin{pmatrix} G \\ f_1 I \end{pmatrix} = \begin{pmatrix} G_1 \\ O \end{pmatrix}, \dots, N_t \begin{pmatrix} G \\ f_t I \end{pmatrix} = \begin{pmatrix} G_t \\ O \end{pmatrix}$ for some $N_1, \dots, N_t \in GL_{2l}(R)$
```

Let $\{G\}_f$ be the set of all reduced generator polynomial matrices with $AG = fI$ for some $A \in M_l(R)$.

Proposition 1 (Cf. [14]) *Let*

$$\pi : \{G\}_f \to \{G_1\}_{f_1} \times \cdots \times \{G_t\}_{f_t} \quad [G \mapsto (G_1, \dots, G_t)]$$

be a map defined by $\mathbb{L}G + f_1\mathbb{L} = \mathbb{L}G_1, \dots, \mathbb{L}G + f_t\mathbb{L} = \mathbb{L}G_t$. *Moreover, let*

$$\varpi : \{G_1\}_{f_1} \times \cdots \times \{G_t\}_{f_t} \to \{G\}_f \quad [(G_1, \dots, G_t) \mapsto G]$$

be a map defined by $\mathbb{L}e_1G_1 + \cdots + \mathbb{L}e_tG_t = \mathbb{L}G$. *Then* π *is a bijective map and its inverse map* π^{-1} *is equal to* ϖ.

3 Deciding LCD–QC codes via mutually prime factor decomposition

In this section, we show Theorem 1, a method for efficiently deciding and constructing LCD–QC codes via mutually prime factor decomposition. In order to prove the theorem, we introduce Propositions 2 and 3. The notations are as in Sect. 2.5.

Proposition 2 *Let* $f, f_1, f_2 \in R \setminus \mathbb{F}_q$ *divide* $-1 + x^m$, $\gcd(f_1, f_2) = 1$, $f = f_1 f_2$ *and be self-reciprocal, respectively. Let* $G \in \{G\}_f$, $\pi(G) = (G_1, G_2) \in \{G_1\}_{f_1} \times \{G_2\}_{f_2}$ *and* H, H_1, H_2 *be given by* (1), *respectively. Then*

$$\mathbb{L}G + \mathbb{L}H = \mathbb{L} \iff \mathbb{L}G_1 + \mathbb{L}H_1 = \mathbb{L}G_2 + \mathbb{L}H_2 = \mathbb{L}.$$

Proof ($\Longrightarrow$) From Proposition 1, $\mathbb{L}G + f_1\mathbb{L} = \mathbb{L}G_1$. Because of a one-to-one correspondence between G and H, there exists a map $\{H\}_{f^*} \to \{H_1\}_{f_1^*} \times \{H_2\}_{f_2^*}$ and $\mathbb{L}H + f_1^*\mathbb{L} = \mathbb{L}H_1$. Because f_1 is self-reciprocal, $\mathbb{L}H + f_1\mathbb{L} = \mathbb{L}H_1$. We add $f_1\mathbb{L} + f_1\mathbb{L}$ to both sides of $\mathbb{L}G + \mathbb{L}H = \mathbb{L}$ and then

$$\mathbb{L}G+\mathbb{L}H=\mathbb{L} \Longrightarrow \mathbb{L}G+\mathbb{L}H+f_1\mathbb{L}+f_1\mathbb{L}=\mathbb{L}+f_1\mathbb{L}+f_1\mathbb{L}=\mathbb{L} \iff \mathbb{L}G_1+\mathbb{L}H_1=\mathbb{L}.$$

Similarly, $\mathbb{L}G_2+\mathbb{L}H_2=\mathbb{L}$. ($\Longleftarrow$) It follows from $\mathbb{L}G_1+\mathbb{L}H_1=\mathbb{L}$ that $\mathbb{L}e_1G_1+\mathbb{L}e_1H_1=\mathbb{L}e_1$. Similarly, $\mathbb{L}e_2G_2+\mathbb{L}e_2H_2=\mathbb{L}e_2$. We add these both sides and it follows from $\mathbb{L}e_1G_1+\mathbb{L}e_2G_2=\mathbb{L}G$ and $\mathbb{L}e_1H_1+\mathbb{L}e_2H_2=\mathbb{L}H$ that $\mathbb{L}G+\mathbb{L}H=\mathbb{L}$. □

Proposition 3 *Let $f, f_1, f_2 \in R\setminus\mathbb{F}_q$ divide $-1+x^m$, $\gcd(f_1,f_2)=1$ and $f=f_1f_2$ and let f_1, f_2 be mutually reciprocal. Let $G\in\{G\}_f$, $\pi(G)=(G_1,G_2)\in\{G_1\}_{f_1}\times\{G_2\}_{f_2}$ and H, H_1, H_2 be given by (1), respectively. Then*

$$\mathbb{L}G+\mathbb{L}H=\mathbb{L} \iff \mathbb{L}G_1+\mathbb{L}H_2=\mathbb{L}G_2+\mathbb{L}H_1=\mathbb{L}.$$

Proof ($\Longrightarrow$) From Proposition 1, $\mathbb{L}G+f_1\mathbb{L}=\mathbb{L}G_1$. Because f_1, f_2 are mutually reciprocal, $\mathbb{L}H+f_2\mathbb{L}=\mathbb{L}H_1$ and $\mathbb{L}H+f_1\mathbb{L}=\mathbb{L}H_2$. We add $f_1\mathbb{L}+f_1\mathbb{L}$ to both sides of $\mathbb{L}G+\mathbb{L}H=\mathbb{L}$ and then

$$\mathbb{L}G+\mathbb{L}H=\mathbb{L} \Longrightarrow \mathbb{L}G+\mathbb{L}H+f_1\mathbb{L}+f_1\mathbb{L}=\mathbb{L}+f_1\mathbb{L}+f_1\mathbb{L}=\mathbb{L} \iff \mathbb{L}G_1+\mathbb{L}H_2=\mathbb{L}.$$

Similarly, $\mathbb{L}G_1+\mathbb{L}H_2=\mathbb{L}$. ($\Longleftarrow$) It follows from $\mathbb{L}G_1+\mathbb{L}H_2=\mathbb{L}$ that $\mathbb{L}e_1G_1+\mathbb{L}e_1H_2=\mathbb{L}e_1$. Similarly, $\mathbb{L}e_2G_2+\mathbb{L}e_2H_1=\mathbb{L}e_2$. We add these both sides and it follows from $\mathbb{L}e_1G_1+\mathbb{L}e_2G_2=\mathbb{L}G$ and $\mathbb{L}e_1H_2+\mathbb{L}e_1H_2=\mathbb{L}H$ that $\mathbb{L}G+\mathbb{L}H=\mathbb{L}$. □

Lemma 2 *The notations are as in Proposition 3. Let $k_1=\dim(\mathcal{C}_1)$ and $k_2=\dim(\mathcal{C}_2)$, where $\mathcal{C}_1=\mathbb{L}G_1/f_1\mathbb{L}$ and $\mathcal{C}_2=\mathbb{L}G_2/f_2\mathbb{L}$. If two of the following three conditions are satisfied, the remaining condition is also satisfied.*

(1) $\mathbb{L}G_1+\mathbb{L}H_2=\mathbb{L}$,
(2) $\mathbb{L}G_2+\mathbb{L}H_1=\mathbb{L}$,
(3) $k_1=k_2$.

Proof Because $\deg(f_1)=\deg(f_2)$, $\dim(\mathbb{L}/f_1\mathbb{L})=\dim(\mathbb{L}/f_2\mathbb{L})$. Thus $n_1-k_1=\dim(^\perp\mathcal{C}_1)$ and $n_1-k_2=\dim(^\perp\mathcal{C}_2)$, where $n_1=\dim(\mathbb{L}/f_1\mathbb{L})$. Then $\mathbb{L}H_2/f_1\mathbb{L}={}^\perp\mathcal{C}_2$ and, for some $S_1\in M_l(R)$,

$$\mathbb{L}G_1+\mathbb{L}H_2=\mathbb{L}S_1 \iff k_1+n_1-k_2-\dim(\mathcal{C}_1\cap{}^\perp\mathcal{C}_2)=\dim(\mathbb{L}S_1/f_1\mathbb{L}). \quad (3)$$

Similarly, $\mathbb{L}H_1/f_2\mathbb{L}={}^\perp\mathcal{C}_1$ and, for some $S_2\in M_l(R)$,

$$\mathbb{L}G_2+\mathbb{L}H_1=\mathbb{L}S_2 \iff k_2+n_1-k_1-\dim(\mathcal{C}_2\cap{}^\perp\mathcal{C}_1)=\dim(\mathbb{L}S_2/f_2\mathbb{L}). \quad (4)$$

First, we show that (1) and (2) deduce (3). If $I=S_1=S_2$, then $n_1=\dim(\mathbb{L}S_1/f_1\mathbb{L})=\dim(\mathbb{L}S_2/f_2\mathbb{L})$. Because of (3) and (4), $\dim(\mathcal{C}_1\cap{}^\perp\mathcal{C}_2)=\dim(\mathcal{C}_2\cap{}^\perp\mathcal{C}_1)=0$ and 3). Next, we show that (3) and (1) deduce (2). Because $I=S_1$ and $k_1=k_2$ in (3), $\dim(\mathcal{C}_1\cap{}^\perp\mathcal{C}_2)=0$. Because $^\perp(\mathcal{C}_1\cap{}^\perp\mathcal{C}_2)={}^\perp\mathcal{C}_1+\mathcal{C}_2=n_1$ deduces $\dim(\mathcal{C}_2\cap{}^\perp\mathcal{C}_1)=0$. Thus from (4), $n_1=\dim(\mathbb{L}S_2/f_2\mathbb{L})$, $S_2=I$ and (2). Finally, we similarly show that (2) and (3) deduce (1). □

Remark 2 The notations are as in Proposition 3. In Lemma 2 and Proposition 3, $\mathbb{L}G_1+\mathbb{L}H_2=\mathbb{L}$ does not imply $\mathbb{L}G_2+\mathbb{L}H_1=\mathbb{L}$ and $k_1=k_2$ in general. Let $f_1=1+x+x^3$ and $f_2=1+x^2+x^3$. Let $G_1=\begin{pmatrix}1&0\\0&1\end{pmatrix}$ and $G_2=\begin{pmatrix}1&x\\0&f_2\end{pmatrix}$. Then $H_1=\begin{pmatrix}f_2&0\\0&f_2\end{pmatrix}$ and

$H_2 = \begin{pmatrix} f_1 & 0 \\ x^6 & 1 \end{pmatrix}$ are given by (1). The following $N_{12} \in GL_{2l}(R)$ satisfies $N_{12}\begin{pmatrix} G_1 \\ H_2 \end{pmatrix} = \begin{pmatrix} I \\ O \end{pmatrix}$, hence $\mathbb{L}G_1 + \mathbb{L}H_2 = \mathbb{L}$.

$$N_{12} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ f_1 & 0 & 1 & 0 \\ x^6 & 1 & 0 & 1 \end{pmatrix}, \quad N_{21} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ f_2 & x & 1 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix}.$$

However, the above $N_{21} \in GL_{2l}(R)$ satisfies $N_{21}\begin{pmatrix} G_2 \\ H_1 \end{pmatrix} = \begin{pmatrix} 1 & x \\ 0 & f_2 \\ \multicolumn{2}{c}{O} \end{pmatrix}$ and we do not have $\mathbb{L}G_2 + \mathbb{L}H_1 = \mathbb{L}$ and $k_1 = k_2$.

On the other hand, $k_1 = k_2$ does not imply $\mathbb{L}G_1 + \mathbb{L}H_2 = \mathbb{L}$ and $\mathbb{L}G_2 + \mathbb{L}H_1 = \mathbb{L}$ in general. Let $G_1' = \begin{pmatrix} 1 & x \\ 0 & f_1 \end{pmatrix}$ and $G_2' = \begin{pmatrix} 1 & x \\ 0 & f_2 \end{pmatrix}$. Then $H_1' = \begin{pmatrix} f_2 & 0 \\ x^6 & 1 \end{pmatrix}$ and $H_2' = \begin{pmatrix} f_1 & 0 \\ x^6 & 1 \end{pmatrix}$ are given by (1). The following $N_{12}', N_{21}' \in GL_{2l}(R)$ satisfy $N_{12}'\begin{pmatrix} G_1' \\ H_2' \end{pmatrix} = \begin{pmatrix} 1 & x \\ 0 & f_1 \\ \multicolumn{2}{c}{O} \end{pmatrix}$ and $N_{21}'\begin{pmatrix} G_2' \\ H_1' \end{pmatrix} = \begin{pmatrix} 1 & x \\ 0 & f_2 \\ \multicolumn{2}{c}{O} \end{pmatrix}$ and we do not have $\mathbb{L}G_1' + \mathbb{L}H_2' = \mathbb{L}$ and $\mathbb{L}G_2' + \mathbb{L}H_1' = \mathbb{L}$.

$$N_{12}' = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ f_1 & x & 1 & 0 \\ x^6 & 1+x+x^2+x^4 & 0 & 1 \end{pmatrix}, \quad N_{21}' = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ f_2 & x & 1 & 0 \\ x^6 & 1+x^2+x^3+x^4 & 0 & 1 \end{pmatrix}.$$

For self-reciprocal f, let $\{G\}_f^\star$ be

$$\{G\}_f^\star = \{G \in \{G\}_f \mid \mathbb{L}G + \mathbb{L}H = \mathbb{L}\}. \tag{5}$$

Then $\{G\}_f^\star$ is the set of all reduced generator polynomial matrices G of LCD R-modules with $AG = fI$ for some $A \in M_l(R)$.

Theorem 1 *Let $f_i \in R \setminus \mathbb{F}_q$ for $i = 1, \ldots, s$ be self-reciprocal, $f_j, f_{j+t} \in R \setminus \mathbb{F}_q$ for $j = s+1, \ldots, s+t$ be mutually reciprocal, $-1+x^m = f_1 \cdots f_{s+2t}$ and $\gcd(f_u, f_v) = 1$ with $1 \leq u < v \leq s+2t$. Let a generator polynomial matrix $G_u \in M_l(R)$ satisfy $A_uG_u = f_uI$ for some $A_u \in M_l(R)$. Let H_u be given by (1). We define*

$$\{G_{j,t}\}^\star_{f_jf_{j+t}} = \varpi\left(\left\{\{G_j\}_{f_j} \times \{G_{j+t}\}_{f_{j+t}} \,\middle|\, \mathbb{L}G_j + \mathbb{L}H_{j+t} = \mathbb{L}G_{j+t} + \mathbb{L}H_j = \mathbb{L}\right\}\right),$$

i.e., $\{G_{j,t}\}^\star_{f_jf_{j+t}}$ is the set of $G_{j,t} = \varpi(G_j, G_{j+t}) \in \{G_{j,t}\}_{f_jf_{j+t}}$ for $(G_j, G_{j+t}) \in \{G_j\}_{f_j} \times \{G_{j+t}\}_{f_{j+t}}$ satisfying $\mathbb{L}G_j + \mathbb{L}H_{j+t} = \mathbb{L}G_{j+t} + \mathbb{L}H_j = \mathbb{L}$ in Algorithm 1 and is equal to $\{G\}_f^\star$ for $f = f_jf_{j+t}$ in (5). Then the following map that is equal to π of Proposition 1 restricted to $\{G\}^\star_{-1+x^m}$ is bijective.

$$\{G\}^\star_{-1+x^m} \to \prod_{i=1}^{s}\{G_i\}^\star_{f_i} \times \prod_{j=s+1}^{s+t}\{G_{j,t}\}^\star_{f_jf_{j+t}}.$$

Proof For mutually reciprocal f_j and f_{j+t}, $f_{j,t} = f_j f_{j+t}$ is self-reciprocal, and if $\mathbb{L}G_j + \mathbb{L}H_{j+t} = \mathbb{L}G_{j+t} + \mathbb{L}H_j = \mathbb{L}$, then $G_{j,t} = \varpi(G_j, G_{j+t})$ is LCD because of Proposition 3. The product of two self-reciprocal polynomials is also self-reciprocal. If $G_i, G_{j,t}$ are LCD, then $\varpi(G_i, G_{j,t})$ is LCD because of Proposition 2. Thus, from induction argument, G is in $\{G\}^{\star}_{-1+x^m}$. Conversely, for $G \in \{G\}^{\star}_{-1+x^m}$, $\pi(G) \in \prod\{G_i\}^{\star}_{f_i} \times \prod\{G_{j,t}\}^{\star}_{f_j f_{j+t}}$ by Propositions 2 and 3. Because π and ϖ are inverse each other, the proof is completed. □

Example 1 Let $q = 2, l = 3$ and $m = 7$. Let $f = -1 + x^7$, $f_1 = 1 + x$, $f_2 = 1 + x + x^3$ and $f_3 = 1 + x^2 + x^3$. Then $f = f_1 f_2 f_3$, where f_1 is self-reciprocal and f_2, f_3 are mutually reciprocal. Let

$$G_1 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & f_1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad G_2 = \begin{pmatrix} 1 & 0 & 1 + x^2 \\ 0 & 1 & x \\ 0 & 0 & f_2 \end{pmatrix}, \quad G_3 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & x^2 \\ 0 & 0 & f_3 \end{pmatrix}.$$

Then e_1, e_2, e_3 are

$$e_1 = 1 + x + \cdots + x^6, \quad e_2 = 1 + x + x^2 + x^4, \quad e_3 = 1 + x^3 + x^5 + x^6.$$

H_1 given by (1) is as follows.

$$H_1 = \begin{pmatrix} f_1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & f_1 \end{pmatrix}, \quad N_1 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & f_1 & 0 \\ f_1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & f_1 & 0 & 0 & 1 \end{pmatrix}.$$

Because $N_1 \in GL_{2l}(R)$ satisfies $N_1 \begin{pmatrix} G_1 \\ H_1 \end{pmatrix} = \begin{pmatrix} I \\ O \end{pmatrix}$, G_1 is LCD. Similarly, H_3 given by (1) is as follows, where, e.g., [1101] indicates $1 + x + x^3$.

$$H_3 = \begin{pmatrix} [1101] & 0 & 0 \\ 0 & [1101] & 0 \\ 0 & [000001] & 1 \end{pmatrix}, \quad N_2 = \begin{pmatrix} 1 & [0000011011] & [01011111] & 0 & 0 & [11011] \\ 0 & [100000111] & [0010011] & 0 & 0 & [0111] \\ 0 & [00000111] & [010011] & 0 & 0 & [111] \\ 0 & [000001101] & [1000001] & 0 & 0 & [1101] \\ [1101] & 0 & [101] & 1 & 0 & 0 \\ 0 & [1101] & [01] & 0 & 1 & 0 \end{pmatrix}.$$

Because $N_2 \in GL_{2l}(R)$ satisfies $N_2 \begin{pmatrix} G_2 \\ H_3 \end{pmatrix} = \begin{pmatrix} I \\ O \end{pmatrix}$, $\mathbb{L}G_2 + \mathbb{L}H_3 = \mathbb{L}$.

From Algorithm 1, $G = \begin{pmatrix} 1 & 0 & x^2 + x^4 + x^5 \\ 0 & 1 + x & 1 + x + x^3 + x^4 \\ 0 & 0 & 1 + x + x^2 + x^3 + x^4 + x^5 + x^6 \end{pmatrix}$ can be constructed by $N \in GL_{3l}(R)$ satisfying $N \begin{pmatrix} e_1 G_1 \\ e_2 G_2 \\ e_3 G_3 \end{pmatrix} = \begin{pmatrix} G \\ O \\ O \end{pmatrix}$. We can confirm that G is LCD by (2), but here we apply the method of [11]. Converting G to a generator matrix $\mathcal{G}$ of a linear code $\mathcal{C}$ over $\mathbb{F}_2$, we can confirm that $\mathcal{C}$ is an LCD code because $\det\left(\mathcal{G}\left(\mathcal{G}^{\top}\right)\right) \neq 0$.

[11]	$G_i \xrightarrow{\text{Algorithm 1}} G \xrightarrow{\text{convert}} \mathcal{G}$, deciding LCD by $\det\left(\mathcal{G}\left(\mathcal{G}^\top\right)\right) \neq 0$
[17]	$G_i \xrightarrow{\text{Algorithm 1}} G$, deciding LCD by (2)
Ours	G_i deciding LCD by Theorem 1 $\xrightarrow{\text{Algorithm 1}} G$

Fig. 1 Constructing generator polynomial matrices of LCD–QC codes via three decision methods

$$N = \begin{pmatrix} 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & [1001] & 0 & [0001] & 0 & 0 & [11] & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & [010001] & 0 & [111001] & 0 & 0 & [1011] & 0 \\ [0011] & 0 & [1111] & [1101] & 0 & 0 & [1011] & 0 & 0 \\ [11] & 0 & [1111] & [1101] & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & [011] & 0 & [1101] & 0 & 0 & 0 & 0 \\ 0 & 0 & [11] & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & [1111] & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix},$$

$$\mathcal{G} = \begin{pmatrix} 100000000000000010110 \\ 010000000000000001011 \\ 001000000000001000101 \\ 000100000000001100010 \\ 000010000000000110001 \\ 000001000000001011000 \\ 000000100000000101100 \\ 000000011000001101100 \\ 000000001100000110110 \\ 000000000110000011011 \\ 000000000011001001101 \\ 000000000001101100110 \\ 000000000000110110011 \\ 000000000000001111111 \end{pmatrix}.$$

Remark 3 There are the other two decision methods for LCD–QC codes. One is based on the generator matrix $\mathcal{G}$ of a linear code $\mathcal{C}$ [11] and the other is based on the global generator polynomial matrix G [17]. In this paper, we have proposed a decision method based on the local generator polynomial matrix G_i. Let $-1+x^m = f_1 \cdots f_t, m_i = \deg(f_i)$ with $1 \leq i \leq t$ and $m = m_1 + \cdots + m_t$. We summarize three decision methods for LCD–QC codes in Fig. 1.

We compare the computational complexities of three methods. First, as for the decision method [11], the matrix size of $\mathcal{G}$ is k rows and $n = ml$ columns with $k \leq ml$. Then $\mathcal{G}\left(\mathcal{G}^\top\right)$ and $\det\left(\mathcal{G}\left(\mathcal{G}^\top\right)\right)$ are calculated by matrix product and Gaussian elimination with computational complexities $O(k^2 lm)$ and $O(k^3)$, respectively. Thus, from $k \leq ml$, the computational complexity of the decision method [11] is $O(l^3 m^3)$. Next, as for the decision method [17], A in $AG = fI$ is obtained simultaneously with G in the generation algorithm [12] of G, H is given by (1) without finite-field operation and N in (2) is calculated by Gaussian elimination for $\begin{pmatrix} G \\ H \end{pmatrix}$. The computational complexity of Euclidean algorithm for two polynomials with degree at most m is $O(m^3)$. We apply Euclidean algorithm for $2l$ polynomials in a column of $\begin{pmatrix} G \\ H \end{pmatrix}$, repeat it l times and calculate N with computational complexity $O(l^2m^3)$. Finally, as for our decision method, because the computational complexity for checking LCD of G_i is $O(l^2m_i^3)$ similarly and it is done for $i = 1, \ldots, t$, the computational complexity of ours is $O\big(l^2 \sum_{i=1}^{t} m_i^3\big)$. If we apply the fast multiplication method and Half-GCD (HGCD) in [15], the computational complexity of Euclidean algorithm for two polynomials with degree at most m is reduced to $O(m \log^2(m))$ and those of the method [17] and our decision method can be reduced. The estimation is summarized as Table 1, where our method is reduced compared to the other two.

In the actual LCD–QC-code construction, if we use our decision method, compared to the other ones, the calculation time can be further reduced than that shown in Table 1. While LCD decisions by [11, 17] are performed for G after making G from all G_i with Algorithm 1, our LCD decision is performed for all G_i before making G. Because these G's are automatically LCD, our decision method is suitable for the actual LCD–QC-code construction.

4 Numerical results

We construct $[n, k, d]$ LCD–QC codes over $\mathbb{F}_2$ and provide $[[n, k, d; n-k]]$ EAQECCs. Then the minimum weights d of EAQECCs are compared with the existing largest minimum weights of EAQECCs. In [3], the range of the largest minimum weights for EAQECCs is described as $[[n, k, d_{\text{lower}}\text{-}d_{\text{upper}}; c]]$, where d_{lower} is the lower bound of the minimum weights, i.e., the largest minimum weights of EAQECCs currently found, and d_{upper} is the theoretical upper bound of the minimum weights. The objective of our experiment here is to find the EAQECCs with the minimum weight that equals or exceeds d_{lower}. We note that, in [3], some values of the minimum weights are omitted. As an example, $[[n = 7, k = 2, d_{\text{lower}}\text{-}d_{\text{upper}}; c = 3]]$ with $d_{\text{lower}} = 4$ and $d_{\text{upper}} = 5$ is listed but $[[n = 7, k = 2, d_{\text{lower}}\text{-}d_{\text{upper}}; c = 4, 5]]$ is not listed. Because $[[n, k, d; c+1]]$ codes can be constructed from $[[n, k, d; c]]$ codes, we have [[7, 2, 4-5; 4]] and [[7, 2, 4-5; 5]] for $c = 4, 5$, respectively.

The LCD–QC-code search in this section is conducted in MATLAB on an AMD Ryzen Threadripper PRO 3995WX 64 Core PC. We show in Tables 2 and 3 the largest minimum weights of EAQECCs made from LCD–QC codes up to $6 \leq n \leq 39$, $3 \leq m \leq 16$ or $2 \leq l \leq 6$ for $q = 2$ and $4 \leq n \leq 28$, $2 \leq m \leq 14$ or $2 \leq l \leq 6$ for $q = 3$, where "–" indicates that the LCD–QC code in the parameter does not exist, the italic numbers and underlines indicate those that meet d_{lower} and the bold numbers and overlines indicate those that exceed d_{lower}. As shown in Tables 2 and 3, we have found various EAQECCs with minimum weight $d = d_{\text{lower}}$. We have also found 15 EAQECCs with minimum weight $d > d_{\text{lower}}$. Table 4 shows d_{lower} and d_{upper} for these (q, m, l, k)'s and Table 5 shows these generator polynomial matrices.

5 Conclusion

In this paper, we have proposed a method for constructing LCD–QC codes with generator polynomial matrices through mutually prime factor decomposition of $-1 + x^m$. While a generator matrix $\mathcal{G}$ as a linear code has $k \times n = kml$ $\mathbb{F}_q$-elements, a generator polynomial matrix G has at most $l \times ml$ $\mathbb{F}_q$-elements and it has fewer $\mathbb{F}_q$-elements than a generator matrix if $k > l$ ordinarily. Our main contribution is that we have proved a method of determining an LCD–QC code by attributing the global LCD property of G to the local LCD property of G_i,

Table 2 Comparison of the largest minimum weights of our EAQECCs made from LCD–QC codes with those of the existing EAQECCs in [3] for $q = 2$

m	$l\backslash k$	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39
3	2	3	3	2	$\underline{\mathit{2}}$	$\underline{\mathit{1}}$	$\underline{\mathit{1}}$																																	
	3	$\underline{\mathit{9}}$	$\underline{\mathit{6}}$	3	4	3	2	$\underline{\mathit{2}}$	$\underline{\mathit{2}}$	$\underline{\mathit{1}}$																														
	4	9	6	6	4	4	4	3	2	2	$\underline{\mathit{2}}$	$\underline{\mathit{1}}$	$\underline{\mathit{1}}$																											
	5	$\underline{\mathit{15}}$	$\underline{\mathit{10}}$	7	$\underline{\mathit{6}}$	5	6	4	4	4	$\underline{\mathit{3}}$	2	$\underline{\mathit{2}}$	$\underline{\mathit{2}}$	$\underline{\mathit{2}}$	$\underline{\mathit{1}}$																								
	6	15	$\underline{\mathit{10}}$	$\underline{\mathit{9}}$	8	7	7	6	$\underline{\mathit{6}}$	5	4	$\underline{\mathit{4}}$	$\underline{\mathit{4}}$	$\underline{\mathit{3}}$	2	2	$\underline{\mathit{2}}$	$\underline{\mathit{1}}$																						
4	2	–	–	–	3	–	–	–	$\underline{\mathit{1}}$																															
	3	–	–	–	5	–	–	–	2	–	–	–	$\underline{\mathit{1}}$																											
	4	–	–	–	7	–	–	–	5	–	–	–	2	–	–	–	$\underline{\mathit{1}}$																							
5	2	5	5	–	4	3	3	–	$\underline{\mathit{2}}$	$\underline{\mathit{1}}$	$\underline{\mathit{1}}$																													
	3	$\underline{\mathit{15}}$	$\underline{\mathit{10}}$	5	6	5	$\underline{\mathit{6}}$	4	4	4	$\underline{\mathit{3}}$	2	$\underline{\mathit{2}}$	$\underline{\mathit{2}}$	$\underline{\mathit{2}}$	$\underline{\mathit{1}}$																								
	4	15	$\underline{\mathit{10}}$	5	$\overline{\mathbf{10}}$	$\underline{\mathit{9}}$	8	5	6	6	6	$\underline{\mathit{5}}$	4	4	4	3	2	$\underline{\mathit{2}}$	$\underline{\mathit{2}}$	$\underline{\mathit{1}}$	$\underline{\mathit{1}}$																			
6	2	–	6	–	4	–	4	–	2	–	$\underline{\mathit{2}}$	-	$\underline{\mathit{1}}$																											
	3	–	9	–	8	–	6	–	$\underline{\mathit{6}}$	–	4	–	$\underline{\mathit{4}}$	–	2	–	$\underline{\mathit{2}}$	–	$\underline{\mathit{1}}$																					
7	2	7	7	–	–	–	4	$\underline{\mathit{4}}$	3	–	–	–	$\underline{\mathit{2}}$	$\underline{\mathit{1}}$	$\underline{\mathit{1}}$																									
	3	$\underline{\mathit{21}}$	$\overline{\mathbf{14}}$	7	–	–	8	7	7	6	–	–	4	4	4	3	–	–	$\underline{\mathit{2}}$	$\underline{\mathit{2}}$	$\underline{\mathit{2}}$	$\underline{\mathit{1}}$																		
	4	21	$\underline{\mathit{14}}$	7	7	–	$\overline{\mathbf{12}}$	$\overline{\mathbf{11}}$	$\underline{\mathit{10}}$	7	7	–	8	$\underline{\mathit{7}}$	$\overline{\mathbf{7}}$	$\underline{\mathit{6}}$	$\underline{\mathit{6}}$	–	$\underline{\mathit{4}}$	$\underline{\mathit{4}}$	$\underline{\mathit{4}}$	3	3	–	2	$\underline{\mathit{2}}$	$\underline{\mathit{2}}$	$\underline{\mathit{1}}$	$\underline{\mathit{1}}$											

Table 2 continued

m	$l\backslash k$	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39
8	2	–	–	–	–	–	–	–	5	–	–	–	–	–	–	–	$\underline{\mathit{1}}$																							
	3	–	–	–	–	–	–	–	$\underline{\mathit{8}}$	–	–	–	–	–	–	–	4	–	–	–	–	–	–	–	$\underline{\mathit{1}}$															
9	2	9	9	6	6	3	6	6	$\underline{\mathit{6}}$	5	4	3	3	2	2	2	$\underline{\mathit{2}}$	$\underline{\mathit{1}}$	$\underline{\mathit{1}}$																					
	3	$\underline{\mathit{27}}$	$\overline{\mathbf{18}}$	9	$\underline{\mathit{12}}$	9	$\overline{\mathbf{12}}$	$\overline{\mathbf{10}}$	$\overline{\mathbf{10}}$	$\underline{\mathit{9}}$	$\underline{\mathit{9}}$	8	6	6	$\underline{\mathit{6}}$	$\underline{\mathit{6}}$	$\underline{\mathit{6}}$	$\underline{\mathit{5}}$	$\underline{\mathit{4}}$	$\underline{\mathit{4}}$	$\underline{\mathit{4}}$	3	2	2	$\underline{\mathit{2}}$	$\underline{\mathit{2}}$	$\underline{\mathit{2}}$	$\underline{\mathit{1}}$												
10	2	–	$\underline{\mathit{10}}$	–	5	–	–	–	6	–	5	–	4	–	–	–	2	–	$\underline{\mathit{2}}$	–	$\underline{\mathit{1}}$																			
11	2	11	$\underline{\mathit{11}}$	–	–	–	–	–	–	–	6	$\underline{\mathit{6}}$	5	–	–	–	–	–	–	–	$\underline{\mathit{2}}$	$\underline{\mathit{1}}$	$\underline{\mathit{1}}$																	
	3	$\underline{\mathit{33}}$	$\overline{\mathbf{22}}$	11	–	–	–	–	–	–	12	$\underline{\mathit{11}}$	$\underline{\mathit{10}}$	$\overline{\mathbf{10}}$	–	–	–	–	–	–	6	$\underline{\mathit{5}}$	$\underline{\mathit{5}}$	4	–	–	–	–	–	–	$\underline{\mathit{2}}$	$\underline{\mathit{2}}$	$\underline{\mathit{2}}$	$\underline{\mathit{1}}$						
12	2	–	–	–	9	–	–	–	$\underline{\mathit{8}}$	–	–	–	$\underline{\mathit{6}}$	–	–	–	3	–	–	–	2	–	–	–	$\underline{\mathit{1}}$															
13	2	13	$\underline{\mathit{13}}$	–	–	–	–	–	–	–	–	–	$\overline{\mathbf{8}}$	$\underline{\mathit{7}}$	$\underline{\mathit{6}}$	–	–	–	–	–	–	–	–	–	$\underline{\mathit{2}}$	$\underline{\mathit{1}}$	$\underline{\mathit{1}}$													
	3	$\underline{\mathit{39}}$	$\overline{\mathbf{26}}$	13	–	–	–	–	–	–	–	–	12	12	12	11	–	–	–	–	–	–	–	–	$\underline{\mathit{6}}$	$\underline{\mathit{6}}$	$\underline{\mathit{6}}$	5	–	–	–	–	–	–	–	–	$\underline{\mathit{2}}$	$\underline{\mathit{2}}$	$\underline{\mathit{2}}$	$\underline{\mathit{1}}$
14	2	–	$\underline{\mathit{14}}$	–	7	–	–	–	–	–	–	–	8	–	$\overline{\mathbf{7}}$	–	$\underline{\mathit{6}}$	–	–	–	–	–	–	–	2	–	$\underline{\mathit{2}}$	–	$\underline{\mathit{1}}$											
15	2	15	15	10	$\underline{\mathit{12}}$	9	10	9	$\underline{\mathit{12}}$	10	10	9	8	$\underline{\mathit{8}}$	$\underline{\mathit{8}}$	7	6	5	$\overline{\mathbf{6}}$	3	4	$\underline{\mathit{4}}$	$\underline{\mathit{4}}$	2	2	2	2	$\underline{\mathit{2}}$	$\underline{\mathit{2}}$	$\underline{\mathit{1}}$	$\underline{\mathit{1}}$									
16	2	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	8	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	$\underline{\mathit{1}}$							

Table 3 Comparison of the largest minimum weights of our EAQECCs made from LCD–QC codes with those of the existing EAQECCs in [3] for $q = 3$

m	$l\backslash k$	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28
2	2	$\underline{\mathit{4}}$	2	$\underline{\mathit{2}}$	$\underline{\mathit{1}}$																								
	3	4	4	2	2	1	$\underline{\mathit{1}}$																						
	4	$\overline{\mathbf{8}}$	4	4	4	2	2	$\underline{\mathit{2}}$	$\underline{\mathit{1}}$																				
	5	$\underline{\mathit{10}}$	7	5	5	4	3	2	2	$\underline{\mathit{2}}$	$\underline{\mathit{1}}$																		
	6	10	8	7	6	5	5	4	3	2	$\underline{\mathit{2}}$	1	$\underline{\mathit{1}}$																
3	2	–	–	3	–	–	$\underline{\mathit{1}}$																						
	3	–	–	5	–	–	3	–	–	$\underline{\mathit{1}}$																			
	4	–	–	7	–	–	5	–	–	2	–	–	$\underline{\mathit{1}}$																
4	2	$\overline{\mathbf{8}}$	4	4	4	2	2	$\underline{\mathit{2}}$	$\underline{\mathit{1}}$																				
	3	8	8	7	6	4	4	4	3	2	$\underline{\mathit{2}}$	1	$\underline{\mathit{1}}$																
	4	$\underline{\mathit{16}}$	10	9	8	8	7	6	6	4	4	4	3	2	$\underline{\mathit{2}}$	$\underline{\mathit{2}}$	$\underline{\mathit{1}}$												
5	2	$\underline{\mathit{10}}$	5	–	5	4	3	–	2	$\underline{\mathit{2}}$	$\underline{\mathit{1}}$																		
	3	10	5	5	8	8	5	5	5	4	4	3	2	$\underline{\mathit{2}}$	1	$\underline{\mathit{1}}$													
6	2	–	–	6	–	–	5	–	–	2	–	–	$\underline{\mathit{1}}$																
	3	–	–	10	–	–	9	–	–	6	–	–	4	–	–	2	–	–	$\underline{\mathit{1}}$										
	4	–	–	14	–	–	12	–	–	10	–	–	8	–	–	4	–	–	4	–	–	2	–	–	$\underline{\mathit{1}}$				
7	2	$\underline{\mathit{14}}$	7	–	–	–	6	6	5	–	–	–	$\underline{\mathit{2}}$	$\underline{\mathit{2}}$	$\underline{\mathit{1}}$														
	3	14	7	7	–	–	11	10	7	7	–	–	6	5	5	4	–	–	2	$\underline{\mathit{2}}$	1	$\underline{\mathit{1}}$							
8	2	$\underline{\mathit{16}}$	8	8	8	8	7	6	6	4	4	4	2	2	$\underline{\mathit{2}}$	$\underline{\mathit{2}}$	$\underline{\mathit{1}}$												
	3	16	16	14	12	12	12	11	10	8	8	8	6	6	6	6	5	4	4	2	2	2	$\underline{\mathit{2}}$	1	$\underline{\mathit{1}}$				
9	2	–	–	–	–	–	–	–	–	6	–	–	–	–	–	–	–	–	$\underline{\mathit{1}}$										
10	2	$\underline{\mathit{20}}$	10	10	10	10	9	8	8	8	7	6	5	4	4	3	2	2	$\underline{\mathit{2}}$	$\underline{\mathit{2}}$	$\underline{\mathit{1}}$								
11	2	$\underline{\mathit{22}}$	11	–	–	–	–	–	–	–	8	7	6	–	–	–	–	–	–	–	$\underline{\mathit{2}}$	$\underline{\mathit{2}}$	$\underline{\mathit{1}}$						
12	2	–	–	12	–	–	10	–	–	9	–	–	8	–	–	4	–	–	2	–	–	2	–	–	$\underline{\mathit{1}}$				
13	2	$\underline{\mathit{26}}$	13	–	–	–	13	13	11	–	–	–	9	8	7	–	–	–	5	4	4	–	–	–	$\underline{\mathit{2}}$	$\underline{\mathit{2}}$	$\underline{\mathit{1}}$		
14	2	$\underline{\mathit{28}}$	14	14	7	–	12	12	12	10	7	–	10	9	9	8	7	–	4	4	4	4	2	–	2	2	$\underline{\mathit{2}}$	$\underline{\mathit{2}}$	$\underline{\mathit{1}}$

where $-1 + x^m = \prod_{i=1}^{t} f_i$ is the mutually prime factor decomposition and $G, G_i \in M_l(R)$ are generator polynomial matrices which satisfy $AG = (-1 + x^m)I$, $A_iG_i = f_iI$ for some $A, A_i \in M_l(R)$, respectively. It has been found that this decision method can construct LCD–QC codes faster than the other decision methods [11, 17] as shown in Table 1. Next, we have conducted experiments to build EAQECCs through LCD–QC codes and finding EAQECCs with the largest minimum weights in fixed parameters, where the results are summarized in Tables 2, 3, 4, and 5, and we have updated the values of [3] for 15 parameters.

There is an issue to be addressed in the future. It is to use k-Galois inner products [9] that include Euclidean and Hermitian inner products, where EAQECCs can be obtained from LCD codes with k-Galois inner products.

Table 4 Novel EAQECCs with minimum weights that exceed d_{lower} in Tables 2 and 3

q	m	l	k	d_{lower}	d	d_{upper}
2	5	4	4	9	10	16
	7	3	2	10	14	18
		4	6	10	12	21
			7	10	11	21
	9	3	2	14	18	24
			6	10	12	20
			7	9	10	18
			8	9	10	19
	11	3	2	16	22	30
			13	9	10	17
	13	2	12	7	8	11
		3	2	20	26	36
	14	2	14	6	7	10
	15	2	18	5	6	9
3	4	2	1	7	8	7

Table 5 Generator polynomial matrices of LCD–QC codes that produce EAQECCs in Tables 2 and 3

(q, m, l, k, d)	G
(2, 5, 4, 4, 10)	$\begin{pmatrix} [11] & [11] & [101] & [10111] \\ 0 & [100001] & 0 & 0 \\ 0 & 0 & [100001] & 0 \\ 0 & 0 & 0 & [100001] \end{pmatrix}$
(2, 7, 3, 2, 14)	$\begin{pmatrix} [1111111] & 0 & [1111111] \\ 0 & [1111111] & [1111111] \\ 0 & 0 & [10000001] \end{pmatrix}$
(2, 7, 4, 6, 12)	$\begin{pmatrix} [10111] & 0 & [10111] & [10111] \\ 0 & [11101] & [11101] & [011101] \\ 0 & 0 & [10000001] & 0 \\ 0 & 0 & 0 & [10000001] \end{pmatrix}$
(2, 7, 4, 7, 11)	$\begin{pmatrix} [1101] & [1011] & [1001111] & [10001] \\ 0 & [11101] & [11101] & [011101] \\ 0 & 0 & [10000001] & 0 \\ 0 & 0 & 0 & [10000001] \end{pmatrix}$
(2, 9, 3, 2, 18)	$\begin{pmatrix} [111111111] & 0 & [111111111] \\ 0 & [111111111] & [111111111] \\ 0 & 0 & [1000000001] \end{pmatrix}$
(2, 9, 3, 6, 12)	$\begin{pmatrix} [1001] & [11011] & [110101011] \\ 0 & [1000000001] & 0 \\ 0 & 0 & [1000000001] \end{pmatrix}$
(q, m, l, k, d)	G
(2, 9, 3, 7, 10)	$\begin{pmatrix} [111] & [011011111] & [001001111] \\ 0 & [1000000001] & 0 \\ 0 & 0 & [1000000001] \end{pmatrix}$

Table 5 continued

(q, m, l, k, d)	G
(2, 9, 3, 8, 10)	$\begin{pmatrix} [111] & [111] & [011010001] \\ 0 & [111111111] & [111111111] \\ 0 & 0 & [1000000001] \end{pmatrix}$
(2, 11, 3, 2, 22)	$\begin{pmatrix} [11111111111] & 0 & [11111111111] \\ 0 & [11111111111] & [11111111111] \\ 0 & 0 & [100000000001] \end{pmatrix}$
(2, 11, 3, 13, 10)	$\begin{pmatrix} 1 & [0010010111] & [1101110001] \\ 0 & [11111111111] & 0 \\ 0 & 0 & [11111111111] \end{pmatrix}$
(2, 13, 2, 12, 8)	$\begin{pmatrix} [11] & [1101000101111] \\ 0 & [10000000000001] \end{pmatrix}$
(2, 13, 3, 2, 26)	$\begin{pmatrix} [1111111111111] & 0 & [1111111111111] \\ 0 & [1111111111111] & [1111111111111] \\ 0 & 0 & [10000000000001] \end{pmatrix}$
(2, 14, 2, 14, 7)	$\begin{pmatrix} 1 & [1111010001] \\ 0 & [100000000000001] \end{pmatrix}$
(2, 15, 2, 18, 6)	$\begin{pmatrix} [11] & [0110101011] \\ 0 & [110001100011] \end{pmatrix}$
(3, 4, 2, 1, 8)	$\begin{pmatrix} [1111] & [1111] \\ 0 & [20001] \end{pmatrix}$

Acknowledgements This work was supported in part by JSPS KAKENHI Grant Number JP24K14831.

Funding Open access funding provided by Kagoshima University.

References

1. Brun T., Devetak I., Hsieh M.H.: Correcting quantum errors with entanglement. Science **314**(5798), 436–439 (2006).
2. Calderbank A.R., Shor P.W.: Good quantum error-correcting codes exist. Phys. Rev. A **54**, 1098–1105 (1996).
3. Grassl M.: Code Tables: bounds on the parameters of various types of codes, last updated on 27 May (2024). http://codetables.de/.
4. Güneri C., Özkaya B., Solé P.: Quasi-cyclic complementary dual codes. Finite Fields Their Appl. **42**, 67–80 (2016).
5. Güneri C., Özbudak F., Özkaya B., Saçıkara E., Sepasdar Z., Solé P.: Structure and performance of generalized quasi-cyclic codes. Finite Fields Their Appl. **47**, 183–202 (2017).
6. Kasami T.: A Gilbert–Varshamov bound for quasi-cycle codes of rate 1/2. IEEE Trans. Inf. Theory **20**(5), 679 (1974).

7. Lally K., Fitzpatrick P.: Algebraic structure of quasicyclic codes. Discret. Appl. Math. **111**(1–2), 157–175 (2001).
8. Ling S., Solé P.: On the algebraic structure of quasi-cyclic codes, I. Finite fields. IEEE Trans. Inf. Theory **47**(7), 2751–2760 (2001).
9. Liu X., Yu L., Hu P.: New entanglement-assisted quantum codes from k-Galois dual codes. Finite Fields Their Appl. **55**, 21–32 (2019).
10. Luo G., Ezerman M.F., Grassl M., Ling S.: How much entanglement does a quantum code need? (2022). arXiv:2207.05647v2.
11. Massey J.L.: Linear codes with complementary duals. Discret. Math. **106–107**, 337–342 (1992).
12. Matsui H.: On generator and parity-check polynomial matrices of generalized quasi-cyclic codes. Finite Fields Their Appl. **34**, 280–304 (2015).
13. Matsui H.: Multiplicative structure and Hecke rings of generator matrices for codes over quotient rings of Euclidean domains. MDPI Math. **5**(4), 82 (2017). https://doi.org/10.3390/math5040082.
14. Matsui H.: A modulus factorization algorithm for self-orthogonal and self-dual quasi-cyclic codes via polynomial matrices. IEICE Trans. Fundam. **E104–A**(11), 1649–1653 (2021).
15. Moenck R.T.: Fast computation of GCDs. In: STOC ’73: Proceedings of the Fifth Annual ACM Symposium on Theory of Computing, April 1973, pp. 142–151 (1973).
16. Ojiro N., Kaneko K., Matsui H.: An efficient algorithm for constructing reversible quasi-cyclic codes via Chinese remainder theorem. Finite Fields Their Appl. **89**, 102204 (2023).
17. Ojiro N., Matsui H.: On generator polynomial matrices of quasi-cyclic codes with linear complementary duals. Accepted by Journal of Algebra Combinatorics Discrete Structures and Applications on March 7, (2025).
18. Qian J., Zhang L.: Entanglement-assisted quantum codes from arbitrary binary linear codes. Des. Codes Cryptogr. **77**, 193–202 (2015).
19. Terhal B.M.: Quantum error correction for quantum memories. Rev. Mod. Phys. **87**(2), 307–346 (2015).
20. Wilde M.M., Brun T.A.: Optimal entanglement formulas for entanglement-assisted quantum coding. Phys. Rev. A **77**, 064302 (2008).