List Decoding Algorithm based on Voting in Gröbner Bases for General One-Point AG Codes

Olav Geil, Ryutaroh Matsumoto, Member, IEEE, and Diego Ruano

Abstract—We generalize the unique decoding algorithm for one-point AG codes over the Miura-Kamiya $C_p$ curves proposed by Lee, Bras-Amorós and O’Sullivan to general one-point AG codes, without any assumption. We also extend their unique decoding algorithm to list decoding, modify it so that it can be used with the Feng-Rao improved code construction, analyze its error correcting capability that has not been done in the original proposal except one-point Hermitian codes, remove the unnecessary computational steps so that it can run faster, and analyze its computational complexity in terms of multiplications and divisions in the finite field. As a unique decoding algorithm, the proposed one is as fast as the BMS algorithm for one-point Hermitian codes, and as a list decoding algorithm it is much faster than the algorithm by Beelen and Brander.

Index Terms—algebraic geometry code, Gröbner basis, list decoding

I. INTRODUCTION

We consider the list decoding of one-point algebraic geometry (AG) codes. Guruswami and Sudan [21] proposed the well-known list decoding algorithm for one-point AG codes, which consists of the interpolation step and the factorization step. The interpolation step has large computational complexity and many researchers have proposed faster interpolation steps, see [5] Figure 1).

By modifying the unique decoding algorithm [25] for primal one-point AG codes, we propose another list decoding algorithm based on voting in Gröbner bases whose error correcting capability is higher than [21] and whose computational complexity is smaller than [5], [21] in many cases. A decoding algorithm for primal one-point AG codes was proposed in [29], which was a straightforward adaptation of the original Feng-Rao majority voting for the dual AG codes [13] to the primal ones. The Feng-Rao majority voting in [29] for one-point primal codes was generalized to multi-point primal codes in [6] Sec. 2.5]. The one-point primal codes can also be decoded as multi-point dual codes with majority voting [4], [9], [10]. Lee, Bras-Amorós and O’Sullivan [25] proposed another unique decoding (not list decoding) algorithm for primal codes based on the majority voting inside Gröbner bases. The module used by them [24] is a curve theoretic generalization of one used for Reed-Solomon codes in [2]. An interesting feature in [25] is that it did not use differentials and residues on curves for its majority voting, while they were used in [6], [29]. The above studies [6], [25], [29] dealt with the primal codes. Chen [8], Elbrønd Jensen et al. [12] and Bras-Amorós et al. [7] studied the error correction capability of the Feng-Rao [13] or the BMS algorithm [37], [38] with majority voting beyond half the designed distance that are applicable to the dual one-point codes.

There was room for improvements in the original result [25], namely, (a) they have not analyzed the error-correcting capability except the one-point Hermitian codes, (b) they have not analyzed the computational complexity, (c) they assumed that the maximum pole order used for code construction is less than the code length, and (d) they have not shown how to use the method with the Feng-Rao improved code construction [14]. We shall (1) prove that the error-correcting capability of the original proposal is always equal to half of the bound in [3] for the minimum distance of one-point primal codes (Proposition 6), (2) generalize their algorithm to work with any one-point AG codes, (3) modify their algorithm to a list decoding algorithm, (4) remove the assumptions (c) and (d) above, (5) remove unnecessary computational steps from the original proposal, (6) analyze the computational complexity in terms of the number of multiplications and divisions in the finite field. The proposed algorithm is implemented on the Singular computer algebra system [20], and we verified that the proposed algorithm can correct more errors than [5], [21] with manageable computational complexity.

This paper is organized as follows: Sec. II introduces notations and relevant facts. Sec. III improves [25] in various ways, and the differences to the original [25] are summarized in Sec. II-H. Sec. IV shows that the proposed modification to [25] works as claimed. Sec. V compares its computational complexity with the conventional methods. Sec. VI concludes the paper.

II. NOTATION AND PRELIMINARY

Our study heavily relies on the standard form of algebraic curves introduced independently by Pellikaan [19] and Miura [32], which is an enhancement of earlier results [31], [34]. Let $F/F_q$ be an algebraic function field of one variable over a finite field $F_q$ with $q$ elements. Let $g$ be the genus of $F$. Fix $n+1$ distinct places $Q, P_1, \ldots, P_n$ of degree one in $F$ and a nonnegative integer $u$. We consider the following one-point
algebraic geometry (AG) code

\[ C_q = \{ ev(f) \mid f \in \mathcal{L}(uQ) \} \tag{1} \]

where \( ev(f) = (f(P_1), \ldots, f(P_m)) \). Suppose that the Weierstrass semigroup \( H(Q) \) at \( Q \) is generated by \( a_1, \ldots, a_t \), and choose \( t \) elements \( x_1, \ldots, x_t \) in \( F \) whose pole divisors are \( (x_i)_Q = a_iQ \) for \( i = 1, \ldots, t \). We do not assume that \( a_1 \) is the smallest among \( a_1, \ldots, a_t \). Without loss of generality we may assume the availability of such \( x_1, \ldots, x_t \), because otherwise we cannot find a basis of \( C_q \) for every \( u \). Then we have that \( \mathcal{L}(\infty \infty) = \mathcal{L}(Q) \) is equal to \( F_q[x_1, \ldots, x_t] \) [34]. We express \( \mathcal{L}(\infty \infty) \) as a residue class ring \( F_q[x_1, \ldots, x_t]/I \) of the polynomial ring \( F_q[x_1, \ldots, x_t] \), where \( x_1, \ldots, x_t \) are transcendental over \( F_q \), and \( I \) is the kernel of the canonical homomorphism sending \( x_i \) to \( x_i \), Pellikaan and Miura [19]. [32] identified the following convenient representation of \( \mathcal{L}(\infty \infty) \) by using the Gröbner basis theory [11]. The following review is borrowed from [23]. Hereafter, we assume that the reader is familiar with the Gröbner basis theory in [11].

Let \( N_0 \) be the set of nonnegative integers. For \( (m_1, \ldots, m_t) \), \( (n_1, \ldots, n_t) \) \( \in N_0^t \), we define the weighted reverse lexicographic monomial order \( > \) such that \( (m_1, \ldots, m_t) > (n_1, \ldots, n_t) \) if \( a_1m_1 + \cdots + a_tm_t > a_1n_1 + \cdots + a_tn_t \) or \( a_1m_1 + \cdots + a_tm_t = a_1n_1 + \cdots + a_tn_t \) and \( m_1 = n_1, m_2 = n_2, \ldots, m_{t-1} = n_{t-1}, m_t < n_t \), for some \( 1 \leq t \). Note that a Gröbner basis of \( I \) with respect to \( > \) can be computed by [34] Theorem 15) or [41], Proposition 2.17, starting from any defining equations of \( F/F_q \).

**Example 1:** According to [22], Example 3.7],

\[ u^3v + v^3 + u = 0 \]

is an affine defining equation for the Klein quartic over \( F_8 \). There exists a unique \( F_8 \)-rational place \( P \) such that \( (v)_P = 3Q \), \( (uv)_P = 5Q \), and \( (u^2v)_P = 7Q \). The numbers 3, 5 and 7 is the minimal generating set of the Weierstrass semigroup at \( P \). Choosing \( v \) as \( x_1 \), \( uv \) as \( x_2 \) and \( u^2v \) as \( x_3 \), by [41], Proposition 2.17, we can see that the standard form of the Klein quartic is given by

\[ X_2^3 + X_3X_1, X_3X_2 + X_1^3 + X_2X_1^2 + X_2X_1 + X_3, \]

which is the reduced Gröbner basis with respect to the monomial order \( > \). We can see that \( a_1 = 3, a_2 = 5 \), and \( a_3 = 7 \).

**Example 2:** Consider the function field \( F_9(u_1, v_2, v_3) \) with relations

\[ v_3^2 + v_2 = u_1^4, \quad v_3^4 + v_2 = (v_2/u_1)^4. \tag{2} \]

This is the third function field in the asymptotically good tower introduced by Garcia and Stichtenoth [16]. Substituting \( v_2 \) with \( u_1^2u_2^3 \) and \( v_3 \) with \( u_2u_3^2 \) in Eq. (2) we have affine defining equations

\[ u_1^2u_2^3 + u_2 - u_1^3 = 0, \quad u_2^2u_3^4 + u_3 - u_2^3 = 0. \]

The function \( u_1 \) has a unique place \( Q \) in \( F_9(u_1, u_2, u_3) = F_9(u_1, v_2, v_3) \). The minimal generating set of the Weierstrass semigroup \( H(Q) \) at \( Q \) is 9, 12, 22, 28, 32 and 35 [42], Example 4.11]. It has genus 22 and 77 \( F_q \)-rational points except \( Q \) [16].

Define six functions \( x_1 = u_1, \quad x_2 = u_1^2u_2, \quad x_3 = u_1^2u_2u_3, \quad x_4 = u_1^3u_2^3u_3, \quad x_5 = ((u_1u_2)^2 + 1)u_2u_3 \) and \( x_6 = ((u_1u_2)^2 + 1)u_2^2u_3 \). We have \( (x_1)_Q = 9Q, \quad (x_2)_Q = 12Q, \quad (x_3)_Q = 22Q, \quad (x_4)_Q = 35Q, \quad (x_5)_Q = 28Q \) and \( (x_6)_Q = 32Q \) [43]. From this information and [41], Proposition 2.17 we can compute the 15 polynomials in the reduced Gröbner basis of the ideal \( I \subset F_9[x_1, \ldots, x_6] \) defining \( \mathcal{L}(\infty \infty) \) as \( (X_1^2 - X_2 + X_3, \quad X_4X_3 - X_2^3, \quad X_5X_3 - X_2^2X_1, \quad X_6X_3 - X_2X_1^2, \quad X_7X_3 - X_2X_1, \quad X_8X_3 - X_2^3) \).

**Example 3:** For the curve in Example 1 we have \( y_0 = 1, \quad y_1 = x_3, \quad y_2 = x_2 \).

Let \( V_0 \) be the unique valuation in \( F \) associated with the place \( Q \). The semigroup \( H(Q) \) is equal to \( \{ (a_1 - v_Q(y_j)) \mid 0 \leq i \leq j < a_1 \} \) [33], Lemma 2.6. For each nonzero \( s \in H(Q) \) there is a unique monomial \( x^i_jy^j \in \mathcal{L}(\infty \infty) \) with \( 0 < j < a_1 \) such that \( -v_Q(x^i_j) = s \), by [28], Proposition 3.18, let us denote this monomial by \( \varphi_s \). Let \( \Gamma \subset H(Q) \), we may consider the one-point codes

\[ C_T = \{ (ev(\varphi_s)) \mid s \in \Gamma \}, \tag{4} \]

where \( \langle \rangle \) denotes the \( F_q \)-linear space spanned by \( \cdot \). Since considering linearly dependent rows in a generator matrix has no merit, we assume

\[ \Gamma \subseteq \mathcal{H}(Q), \tag{5} \]

where \( \mathcal{H}(Q) = \{ u \in H(Q) \mid C_u \neq C_{u-1} \} \). One motivation for considering these codes is that it was shown in [33] how to increase the dimension of the one-point codes without decreasing the lower bound \( d_{AG} \) for the minimum distance.
The bound $d_{AG}(C_T)$ is defined for $C_T$ as follows [3]: For $s \in \Gamma$, let

$$\lambda(s) = \# \{ j \in H(Q) \mid j + s \in \bar{H}(Q) \}.$$  

(6)

Then $d_{AG}(C_T) = \min \{ \lambda(s) \mid s \in \Gamma \}$. It is proved in [17] that $d_{AG}$ gives the same estimate for the minimum distance as the Feng-Rao bound [11] for one-point dual AG codes when both $d_{AG}$ and the Feng-Rao bound can be applied, that is, when the dual of a one-point code is isometric to a one-point code. Furthermore, it is also proved in [17] that $d_{AG}(C_T)$ can be obtained from the bounds in [4], [9], [10], hence $d_{AG}$ can be understood as a particular case of these bounds [4], [9], [10].

III. Procedure of New List Decoding based on Voting in Gröbner Bases

A. Overall Structure

Suppose that we have a received word $\bar{r} \in F_q^n$. We shall modify the unique decoding algorithm proposed by Lee et al. [25] so that we can find all the codewords in $C_T$ in Eq. (4) within the Hamming distance $r$ from $\bar{r}$. The overall structure of the modified algorithm is as follows:

1) Precomputation before getting a received word $\bar{r}$,

2) Initialization after getting a received word $\bar{r}$,

3) Termination criteria of the iteration, and

4) Main part of the iteration.

Steps 2 and 4 are based on [25]. Steps 1 and 3 are not given in [25]. Each step is described in the following subsections in Sec. [11]. We shall analyze time complexity except the precomputation part of the algorithm.

B. Modified Definitions for the Proposed Modification

We retain notations from Sec. [11]. In this subsection, we modify notations and definitions in [25] to describe the proposed modification to their algorithm. We also introduce several new notations. Define a set $\Omega_1 = \{ x_1 y_1 z_i^2 \mid 0 \leq i, 0 \leq j < a_1, k = 0, 1 \}$. Our $\Omega_1$ is $\Omega$ in [25]. Recall also that $\Omega_0 = \{ \varphi \mid s \in H(Q) \}$.

Since the $F_q[x_1]$-module $L(\infty) \otimes L(\infty)$ has a free basis $\{ y_i, y_j \mid 0 \leq j < a_1 \}$, we can regard $\Omega_1$ as the set of monomials in the Gröbner basis theory for modules. We introduce a monomial order on $\Omega_1$ as follows. For given two monomials $x_1^i y_1 z_1^a_1$ and $x_1^j y_1 z_1^a_1$, first rewrite $y_1$ and $y_j$ by $x_2$, ..., $x_t$ defined in Sec. [11] and get $x_1^i y_1 z_1^a_1 = x_1^i x_2^{a_1} \cdots x_t^{a_1} + \gamma_1$ and $x_1^j y_1 z_1^a_1 = x_1^j x_2^{a_1} \cdots x_t^{a_1} + \gamma_2$. For a nonsing $s \in H(Q)$, we define the monomial order $x_1^{i_1} x_2^{i_2} \cdots x_t^{i_t} \leq x_1^{j_1} x_2^{j_2} \cdots x_t^{j_t}$ parametrized by $s$ if $i_1 + s - v_Q(x_1 x_2^{i_2} \cdots x_t^{i_t}) < i_1 + s - v_Q(x_1 x_2^{j_2} \cdots x_t^{j_t})$ or $i_1 + s - v_Q(x_1 x_2^{i_2} \cdots x_t^{i_t}) = i_1 + s - v_Q(x_1 x_2^{j_2} \cdots x_t^{j_t})$ and $i_1 = j_1$, $i_2 = i_2$, ..., $i_t = i_t$, and $i_t > j_t$ for some $1 \leq t \leq t_1$. Observe that the restriction of $\leq$ to $\Omega_1$ is equal to $\leq$ defined in Sec. [11]. In what follows, every Gröbner basis, leading term, and leading coefficient is obtained by considering the Gröbner basis theory for modules, not for ideals.

For $f \in L(\infty) \otimes L(\infty)$, $\gamma(f)$ denotes the number of nonzero terms in $f$ when $f$ is expressed as an $F_q$-linear combination of monomials in $\Omega_1$. $\gamma_1(f)$ denotes the number of nonzero terms whose coefficients are not in $F_q$.

For the code $C_T$ in Eq. (4), define the divisor $D = P_1 + \cdots + P_n$. Define $L(G - Q) = \bigcup_{t=1}^\infty L(G - iQ)$ for a divisor $G$ of $F_q$. Then $L(G - \infty)$ is an ideal of $L(\infty)$ [27]. Let $\eta_i$ be any element in $L(G - \infty)$ such that $\lim \eta_i = x_1 y_1$ for some $j$. Then by [25] Proposition 1, $[\eta_0, \ldots, \eta_{a_1-1}]$ is a Gröbner basis for $L(D - \infty)$ with respect to $\leq$, as an $F_q[x_1]$-module. For a nonnegative integer $s$, define $\Gamma(\leq) = \{ \psi \in \Gamma \mid \psi' > s \}$, and prec$(s) = \max \{ \psi' \in H(Q) \mid \psi' < s \}$.

We define prec$(0) = -1$.

C. Precomputation before Getting a Received Word

Before getting $\bar{r}$, we need to compute the Pellikaan-Muira standard form of the algebraic curve, $y_0 = 1$, $y_1$, ..., $y_{a_1-1}$, and $\varphi_s$ for $s \in H(Q)$ as defined in Sec. [11]. Also compute $\eta_0$, $\ldots$, $\eta_{a_1-1}$, which can be done by [27].

For each $(i, j)$, express $y_i y_j$ as an $F_q$-linear combination of monomials in $\Omega_0$. Such expressions will be used for computing products and quotients in $L(\infty)$ as explained in Sec. [III-D].

From the above data, we can easily know $L(\infty)$, which will be used in Eqs. (14) and (22).

Find elements $\varphi_s \in \Omega_s$ with $s \in \bar{H}(Q)$. There are $n$ such elements, which we denote by $\psi_1, \ldots, \psi_n$ such that $-v_Q(\psi_i) < -v_Q(\psi_{i+1})$. Compute the $n \times n$ matrix

$$M = \begin{bmatrix}
\psi_1(P_1) & \cdots & \psi_1(P_n) \\
\vdots & \ddots & \vdots \\
\psi_n(P_1) & \cdots & \psi_n(P_n)
\end{bmatrix}^{-1}. 
$$

(7)

D. Multiplication and Division in an Affine Coordinate Ring

In both original unique decoding algorithm [25] and our modified version, we need to quickly compute the product $g h$ of two elements $g, h$ in the affine coordinate ring $L(\infty)$.

In our modified version, we also need to compute the quotient $g / h$ depending on the choice of iteration termination criterion described in Sec. [III-D]. Since the authors could not find quick computational procedures for those tasks in $L(\infty)$, we shall present such ones here.

1) Multiplication in an Affine Coordinate Ring: The normal form of $g$, for $g \in L(\infty)$, is the expression of $g$ written as an $F_q$-linear combination of monomials $\varphi_s \in \Omega_0$. $g, h$ are assumed to be in the normal form. We propose the following procedure to compute the normal form of $g h$. Let the normal form of $y_i y_j$ be

$$g h = \sum_{k=0}^{a_1-1} \gamma_k f_{i,j,k}(x_1).$$

with $f_{i,j,k}(x_1) \in F_q[x_1]$, which is computed in Sec. [III-C].

$X_1, Y_1, \ldots, Y_{a_1-1}$ are variables over $F_q$ without an algebraic relation among them.

1) Assume that $g$ and $h$ are in their normal forms. Change $y_1$ to $Y_1$ and $x_1$ to $X_1$ in $g, h$ for $i = 1, \ldots, a_1 - 1$. Recall that $y_0 = 1$. Denote the results by $G, H$.

2) Compute $G H$. This step needs

$$\gamma(g) \times \gamma(h)$$

(8)

multiplications in $F_q$. 

3) Let $GH = \sum_{0 \leq i, j < a_1} Y_i Y_j F_{G,H,i,j}(X_1)$. Then we have
\[
g h = \sum_{0 \leq i, j < a_1} F_{G,H,i,j}(x_1) \sum_{k=0}^{a_i-1} y_k f_{i,j,k}(x_1). \tag{9}
\]

Therefore, the total number of multiplications in $\mathbf{F}_q$ in this step is at most
\[
\sum_{0 \leq i, j < a_1} \gamma(F_{G,H,i,j}(x_1)) \sum_{k=0}^{a_i-1} y_k f_{i,j,k}(x_1)). \tag{10}
\]

Therefore, the total number of multiplications in $\mathbf{F}_q$ is at most
\[
\gamma(g) \times \gamma(h) + \sum_{0 \leq i, j < a_1} \gamma(F_{G,H,i,j}(x_1)) \sum_{k=0}^{a_i-1} y_k f_{i,j,k}(x_1)). \tag{11}
\]

Define Eq. (11) as multi($g,h$).

We emphasize that when the characteristic of $\mathbf{F}_q$ is 2 and all the coefficients of defining equations belong to $\mathbf{F}_2$, which is almost always the case, then $\gamma(F_{G,H,i,j}(x_1)) \sum_{k=0}^{a_i-1} y_k f_{i,j,k}(x_1))$ in Eq. (11) is zero. This means that $\mathcal{L}(\infty Q)$ has little additional overhead over $\mathbf{F}_q[X]$ for computing products of their elements in terms of the number of $\mathbf{F}_q$-multiplications and divisions.

**Remark 4:** Define $(i,j)$ to be equivalent to $(i',j')$ if $y_i y_j = y_{i'} y_{j'} \in \mathcal{L}(\infty Q)$. Denote by $[i,j]$ the equivalence class represented by $(i,j)$. For $(i,j), (i',j') \in [i,j]$ we have $f_{i,j,k}(x_1) = f_{i',j',k}(x_1)$, which is denoted by $f_{i,j,k}(x_1)$. The right hand side of Eq. (9) can be written as
\[
\sum_{[i,j]} \left( \sum_{(i',j') \in [i,j]} F_{G,H,i',j'}(x_1) \right) \sum_{k=0}^{a_i-1} y_k f_{i,j,k}(x_1)). \tag{12}
\]

By using Eq. (12) instead of Eq. (9), we have another upper bound on the number of multiplications as
\[
\gamma(g) \times \gamma(h) + \sum_{[i,j]} \left( \sum_{(i',j') \in [i,j]} F_{G,H,i',j'}(x_1) \right) \gamma(\sum_{k=0}^{a_i-1} y_k f_{i,j,k}(x_1)). \tag{13}
\]

Since
\[
\gamma(F_{G,H,i',j'}(x_1)) \leq \sum_{(i',j') \in [i,j]} \gamma(F_{G,H,i',j'}(x_1)),
\]

we have Eq. (13) \leq Eq. (11). However, Eq. (13) is almost always the same as Eq. (11) over the curve in Example 2 and Eq. (13) will not be used in our computer experiments in Sec. VII.

2) **Computation of the Quotient:** Assume $h \neq 0$. The following procedure computes the quotient $g/h \in \mathcal{L}(\infty Q)$ or declares that $g$ does not belong to the principal ideal of $\mathcal{L}(\infty Q)$ generated by $h$.

1) Initialize $\sigma = 0$. Also initialize $\zeta = 0$.
2) Check if $-\nu(g) \in -\nu(h) + H(Q)$. If not, declare that $g$ does not belong to the principal ideal of $\mathcal{L}(\infty Q)$ generated by $h$, and finish the procedure.

3) Let $\varphi_t \in \Omega_0$ such that $-\nu(g) = -\nu(\varphi_t h)$. Observe that $\text{lc}(\varphi_t \Lambda M(h)) = \text{lc}(y_s \mod a_1 y_t \nu(h) \mod a_1)$ and that $\text{lc}(y_s \mod a_1 y_t \nu(h) \mod a_1)$ is precomputed as Sec. III-C. Let
\[
F_{g} \equiv t = \text{lc}(g) / \text{lc}(h) \times \text{lc}(\varphi_t \Lambda M(h)) \ \text{}. \tag{14}
\]

Precomputed in Sec. III-C

Computation of $\varphi_t$ needs one multiplication and one division in $\mathbf{F}_q$. Observe that $-\nu(g - \varphi_t h) < -\nu(g)$.

4) Compute the normal form of $\varphi_t \varphi_s h$, which requires at most multi($\varphi_t \varphi_s , h$) multiplications in $\mathbf{F}_q$. Increment $\epsilon$ by $2 + \text{multi}(\varphi_t \varphi_s , h)$.

5) Update $\sigma \leftarrow \sigma + \varphi_t \varphi_s$ and $g \leftarrow g - \varphi_t \varphi_s h$. If the updated $g$ is zero, then output the updated $\sigma$ as the quotient and finish the procedure. Otherwise go to Step 2. This step has no multiplication nor division.

Define quot($g,h$) as $\zeta$ after finishing the above procedure. quot($g,h$) is an upper bound on the number of multiplications and divisions in $\mathbf{F}_q$ in the above procedure. The program variable $\zeta$ is just to define quot($g,h$), and the decoding algorithm does not need to update $\zeta$. Observe also that the above procedure is a straightforward generalization of the standard long division of two univariate polynomials.

E. **Initialization after Getting a Received Word $\overline{r}$**

Let $(i_1, \ldots, i_n)^T = M \overline{r}$, where $M$ is defined in Eq. (7). Define $h_y = \sum_{j=1}^{n} i_j \psi_j$. Then we have $\text{ev}(h_y) = \overline{r}$. The computation of $h_y$ from $\overline{r}$ needs at most $n^2$ multiplications in $\mathbf{F}_q$.

Let $N = -\nu(h_y)$. For $i = 0, \ldots, a_1 - 1$, compute $g_i^{(N)} = \eta_i \in \mathcal{L}(\infty Q)$ and $f_i^{(N)} = y_s (\psi_s h_y) \in \mathcal{L}(\infty Q) \zeta + \mathcal{L}(\infty Q)$. The computation of $f_i^{(N)}$ needs at most multi($y_s, h_y$) multiplications in $\mathbf{F}_q$. Therefore, the total number of multiplications in the initialization is at most
\[
n^2 + \sum_{i=0}^{a_1-1} \text{multi}(y_s, h_y). \tag{15}
\]

Let $s = N$ and execute the following steps.

F. **Three Termination Criteria of the Iteration**

After finishing the initialization step in Sec. III-E we iteratively compute $f_i^{(s)}$ and $g_i^{(s)}$ with $N \geq s \in H(Q) \cup \{-1\}$ and $\nu_s$ with $N \geq s \in H(Q)$ from larger $s$ to smaller $s$. The single iteration consists of two parts: The first part is to check if an iteration termination criterion is satisfied. The second part is computation of $f_i^{(s)}$ and $g_i^{(s)}$ for $N \geq s \in H(Q) \cup \{-1\}$. We describe the first part in Sec. III-F.

Let $f_{\text{min}} = a_0 + a_1 \alpha_1$ having the smallest $-\nu(\alpha_1)$ among $f_0^{(s)}, \ldots, f_{a_1-1}^{(s)}$. In the following subsections, we shall propose three different procedures to judge whether or not iterations in the proposed algorithm can be terminated. In an actual implementation of the proposed algorithm, one criterion is chosen and the chosen one is consistently used throughout the iterations. The first one and the second one are different.
We shall compare the three criteria in Sec. [V-B]. Throughout this paper, \(\text{wt}(\vec{x})\) denotes the Hamming weight of a vector \(\vec{x} \in \mathbb{F}_q^n\).

1) **First Criterion for Judging Termination:** If
   a) If \(n_0/\alpha_1 > \tau\) then include the coefficients of \(-\alpha_0/\alpha_1 + \sum_{s \in \Theta \cup \varphi} w_s \varphi_s\) into the list of transmitted information vectors, and avoid proceeding with it.
   b) Otherwise compute ev\((\sum_{s \in \Theta} w_s \varphi_s)\). This needs at most
   \[
   \left\lfloor \frac{1}{\alpha} \right\rfloor \text{ multiplications and divisions in } \mathbb{F}_q.
   \]
   c) If
   \[
   \left( \text{ev}(-\alpha_0/\alpha_1 + \sum_{s \in \Theta \cup \varphi} w_s \varphi_s) - \bar{r} \right) \leq \tau,
   \]
   then include the coefficients of \(-\alpha_0/\alpha_1 + \sum_{s \in \Theta \cup \varphi} w_s \varphi_s\) into the list of transmitted information vectors, and avoid proceeding with it.

2) **Second Criterion for Judging Termination:**
   a) If \(d_{AG}(C_T) > 2\tau\) then declare “decoding failure” and finish.
   b) Otherwise compute ev\((\sum_{s \in \Theta} w_s \varphi_s)\). This needs at most
   \[
   n_0 \cdot \left( \sum_{s \in \Theta} w_s \varphi_s \right) \text{ multiplications and divisions in } \mathbb{F}_q.
   \]
   c) If
   \[
   \left( \text{ev}(-\alpha_0/\alpha_1 + \sum_{s \in \Theta \cup \varphi} w_s \varphi_s) - \bar{r} \right) \leq \tau,
   \]
   then include the coefficients of \(-\alpha_0/\alpha_1 + \sum_{s \in \Theta \cup \varphi} w_s \varphi_s\) into the list of transmitted information vectors, and avoid proceeding with it.

3) **Third Criterion for Judging Termination:**
   a) If \(-\varphi_Q(\alpha_1) \leq \tau\) then include the coefficients of \(-\alpha_0/\alpha_1 + \sum_{s \in \Theta \cup \varphi} w_s \varphi_s\) into the list of transmitted information vectors, and avoid proceeding with it.
   b) Otherwise compute ev\((\sum_{s \in \Theta} w_s \varphi_s)\). This needs at most
   \[
   n_0 \cdot \left( \sum_{s \in \Theta} w_s \varphi_s \right) \text{ multiplications and divisions in } \mathbb{F}_q.
   \]
   c) If
   \[
   \left( \text{ev}(-\alpha_0/\alpha_1 + \sum_{s \in \Theta \cup \varphi} w_s \varphi_s) - \bar{r} \right) \leq \tau,
   \]
   then include the coefficients of \(-\alpha_0/\alpha_1 + \sum_{s \in \Theta \cup \varphi} w_s \varphi_s\) into the list of transmitted information vectors, and avoid proceeding with it.

5) Finish the iteration no matter what happened in the above steps.

**G. Iteration of Pairing, Voting, and Rebasings**

The iteration of the original algorithm [25] consists of three steps, called pairing, voting, and rebasing. We will make a little change to the original. Our modified version is described below.

1) **Pairing:**
   \[
   g_i^{(s)} = \sum_{0 \leq j < a_1} c_{i,j} y_j + \sum_{0 \leq j < a_1} d_{i,j} y_j, \quad \text{with } c_{i,j}, d_{i,j} \in \mathbb{F}_q[x_1],
   \]
   and let \(\nu_i^{(s)} = \text{ev}(d_i)\). We assume that \(\text{ev}(g_i^{(s)}) = d_{i,j} y_j\). For \(0 \leq i < a_1\), as in [25], there are unique integers \(0 \leq i' < a_1\) and \(k_i\) satisfying
   \[
   -\varphi_Q(a_{i,j}) + s = a_1 k_i - \varphi_Q(y_j).
   \]
   Note that by the definition above
   \[
   i' = i + s \mod a_1,
   \]
   then do the following:

1) **Compute** \(\frac{a_0}{a_1} \in F\). This needs at most
   \[
   \text{quot}(a_0, a_1) \text{ multiplications and divisions in } \mathbb{F}_q.
   \]
   2) If \(a_0/a_1 \in L(\infty, Q)\) and \(a_0/a_1 \in F\) can be written as a linear combination of monomials in \(\varphi_s \in \varphi\), then do the following:
   a) If \(d_{AG}(C_T) > 2\tau\) or \(-\varphi_Q(\alpha_1) \leq \tau\) then include the coefficients of \(-\alpha_0/\alpha_1 + \sum_{s \in \Theta \cup \varphi} w_s \varphi_s\) into the list of transmitted information vectors, and avoid proceeding with it.
   b) Otherwise compute ev\((\sum_{s \in \Theta} w_s \varphi_s)\). This needs at most
   \[
   n_0 \cdot \left( \sum_{s \in \Theta} w_s \varphi_s \right) \text{ multiplications and divisions in } \mathbb{F}_q.
   \]
   c) If
   \[
   \left( \text{ev}(-\alpha_0/\alpha_1 + \sum_{s \in \Theta \cup \varphi} w_s \varphi_s) - \bar{r} \right) \leq \tau,
   \]
   then include the coefficients of \(-\alpha_0/\alpha_1 + \sum_{s \in \Theta \cup \varphi} w_s \varphi_s\) into the list of transmitted information vectors, and avoid proceeding with it.

3) **Second Criterion for Judging Termination:**
   a) If \(n_0 = 0\) and \(-\varphi_Q(\alpha_1) \leq \tau\) then include the vector \((w_s : s \in \Gamma)\) into the list of transmitted information vectors. Finish the iteration.
   b) If \(-\varphi_Q(\alpha_1) > \tau + g\) then finish the iteration.
   c) Otherwise compute ev\((\sum_{s \in \Theta} w_s \varphi_s)\). This needs at most
   \[
   n_0 \cdot \left( \sum_{s \in \Theta} w_s \varphi_s \right) \text{ multiplications and divisions in } \mathbb{F}_q.
   \]
   d) If
   \[
   \left( \text{ev}(-\alpha_0/\alpha_1 + \sum_{s \in \Theta \cup \varphi} w_s \varphi_s) - \bar{r} \right) \leq \tau,
   \]
   then include the vector \((w_s : s \in \Gamma)\) into the list of transmitted information vectors. Finish the iteration.

1Ali and Kuijper proved in [2] Theorem 12] that if the number \(\delta\) of errors satisfies \(2\delta < d_{RS}(C_1)\), where \(d_{RS}(C_1)\) is the minimum distance of \(C_1\) to Reed-Solomon code \(C_1\), then the transmitted codewords are obtained by Ali-Kuijper’s algorithm as \(-\alpha_0(\alpha_1)\). To one-point primal AG codes, \(d_{RS}(C_1)\) can be generalized as either \(d_{AG}(C_T)\) or \(n - s - g\). The former generalization \(d_{AG}(C_T)\) corresponds to the first criterion in Sec. [III-F] and the latter \(n - s - g\) corresponds to the second in Sec. [III-P].
and the integer \(-v_Q(a_{i,j})+s\) is a nongap if and only if \(k_i \geq 0\). Now let \(c_i = \deg_{x_i}(d_{F,i}) - k_i\). Note that the map \(i \mapsto i'\) is a permutation of \([0, 1, \ldots, a-1]\) and that the integer \(c_i\) is defined such that \(a_{i,j} = -v_Q(d_{F,i}y_j) + v_Q(a_{i,j}y_j) - s\).

2) Voting: For each \(i \in \{0, \ldots, a-1\}\), we set

\[
\mu_i = lc(a_{i,j}y_j), \quad w_{i,j} = -\frac{b_{i,j}[x^k]}{\mu_i}, \quad c_i = \max\{c_i, 0\},
\]

where \(b_{i,j}[x^k]\) denotes the coefficient of \(x_k\) of the univariate polynomial \(b_{i,j} \in F_q[x_1]\). We remark that the leading coefficient \(\mu_i\) must be considered after expressing \(a_{i,j}y_j\) by monomials in \(\Omega_0\).

Observe that \(lc(y_i\varphi_j) = lc(y_i y_j \mod a_i)\) and that \(lc(y_i y_j \mod a_i)\) is already precomputed as Sec. III-C. By using that precomputed table, computation of \(\mu_i\) needs one multiplication. The total number of multiplications and divisions in Eq. (22) is

\[
2a_1
\]

excluding negation from the number of multiplication.

Let

\[
\nu(s) = \frac{1}{a_1} \sum_{\sigma \in \Gamma} \max\{-v_Q(\eta_{i'}) + v_Q(y_i) - s, 0\}.
\]

We consider two different candidates depending on whether \(s \in \Gamma\) or not:

- If \(s \in H(Q) \setminus \Gamma\), set \(w = 0\).

- If \(s \in \Gamma\), let \(w\) be one of the element(s) in \(F_q\) with

\[
\sum_{w \neq w_{i,j}} c_i \geq \sum_{w = w_{i,j}} c_i - 2\tau + \nu(s).
\]

Let \(w_s = w\). If several \(w_s\)'s satisfy the condition above, repeat the rest of the algorithm for each of them.

3) Rebaseing: In all of the following cases, we need to compute the normal form of the product \(w\varphi_s \times \sum_{i=0}^{a-1} a_{i,j}y_j\), and the product \(w\varphi_s \times \sum_{i=0}^{a-1} c_i y_j\). For each \(i\), the number of multiplications is

\[
\leq \text{multi}(w\varphi_s, \sum_{j=0}^{a-1} a_{i,j}y_j) + \text{multi}(w\varphi_s, \sum_{j=0}^{a-1} c_i y_j),
\]

where multi(·, ·) is defined in Sec. III-D1.

- If \(w_{s,i} = w\), then let

\[
\gamma_i^{(s)}(z + w\varphi_s),
\]

where the parentheses denote substitution of the variable \(z\) and \(\nu_i^{(s)} = \nu_i\). The number of multiplications in this case is bounded by Eq. (27).

- If \(w_{s,i} \neq w\) and \(c_i > 0\), then let

\[
\gamma_i^{(s)}(z + w\varphi_s),
\]

where the parentheses denote substitution of the variable \(z\) and \(\nu_i^{(s)} = \nu_i\). The number of multiplications in this case is bounded by Eq. (27).

H. Difference to the Original Method

In this subsection, we review advantages of our modified algorithm over the original [25].

- Our version can handle any one-point primal AG codes, while the original can handle codes only coming from the \(C_{ab}\) curves [31]. This generalization is enabled only by replacing \(y_i\) in [25] by \(y_i\) defined in Sec. III-D1.

- Our version can find all the codewords within Hamming distance \(\tau\) from the received word \(\bar{r}\), while the original is a unique decoding algorithm.

- Our version does not compute \(f_i^{(s)}\), \(g_i^{(s)}\) for a Weierstrass gap \(s \notin H(Q)\), while the original computes them for \(N \geq s \notin H(Q)\).

- The original algorithm assumed \(u < n\), where \(u\) is as defined in Eq. (1). This assumption is replaced by another less restrictive assumption [4] in our version.

- Our version supports the Feng-Rao improved code construction [14], while the original does not. This extension is made possible by the change at Eq. (25).

- The first and the second termination criteria come from [2] Theorem 12 and do not exist in the original [25].

- The third termination criterion is essentially the same as the original [25], but examination of the Hamming distance between the decoded codeword and \(\bar{r}\) is added when \(2\tau \geq d_{AG}(C_1)\).

- The original [25] is suitable for parallel implementation on electric circuit similar to the Köetter architecture [24]. Our modified version retains this advantage.

IV. Theoretical Analysis of the Proposed Modification

In this section we prove that our modified algorithm can find all the codewords within Hamming distance \(\tau\) from the received word \(\bar{r}\). We also give upper bounds on the number of iterations in Sec. IV-D.
A. Supporting Lemmas

In Sec. [IV-A] we shall introduce several lemmas necessary in Secs. [IV-B] [IV-E]. Recall that the execution of our modified algorithm can branch when there are multiple candidates satisfying the condition (26). For a fixed sequence of determined \( w_{s,i} \), define \( \hat{\rho}^{(s)} = \vec{r} \) and recursively define \( \rho(s) = \hat{\rho}^{(s)} - \text{ev}(w_{s,i} \varphi_i) \). By definition \( \rho^{(s)} = \hat{\rho}^{(s)} - \text{ev}(\sum_{s \neq s'} w_{s,i} \varphi_i) \).

The following lemma explains why the authors include “Gröbner bases” in the paper title. The module \( I_{\rho(s)} \) was used in [3, 13, 20, 35, 36] but the use of \( I_{\rho(s)} \) with \( s < \max \Gamma \) was new in [25].

Lemma 5: Fix \( s \in H(Q) \cup \{-1\} \). Let \( \rho^{(s)} \) correspond to \( w_s \) (\( s \in \Gamma \)) chosen by the decoding algorithm. Define the \( \mathbb{F}_q[x_1] \)-submodule \( I_{\rho(s)} \) of \( L(\infty Q) \oplus L(\infty \mathbb{R}) \) by

\[ I_{\rho(s)} = \langle a_0 + a_1 \vec{r} \mid a_0, a_1 \in \mathbb{L}(\infty Q), \nu \rangle = \langle a_0 + a_1 \vec{r} \mid a_0, a_1 \in \mathbb{L}(\infty Q), \nu \rangle \geq 1, 1 \leq i \leq n \rangle, \tag{29} \]

where \( \rho^{(s)} = (r_1^{(s)}, \ldots, r_n^{(s)}) \). Then \( \rho^{(s)} \) is a Gröbner basis of \( I_{\rho(s)} \) with respect to \( \nu \), as an \( \mathbb{F}_q[x_1] \)-module.

Proof: This lemma is a generalization of [25] Proposition 11. We can prove this lemma in exactly the same way as the proof of [25] Proposition 11 with replacing \( y \) in [25] with \( y \) and \( s - 1 \) in [25] by \( \nu(s) \).

The following proposition shows that the original decoding algorithm [25] can correct errors up to half the bound \( d_{AG}(\Gamma_s) \), which was not claimed in [25].

Proposition 6: Fix \( s \in \Gamma \). Let \( \lambda(s) \) as defined in Eq. (6) and \( \nu(s) \) as defined in Eq. (24). Then \( \nu(s) = \lambda(s) \).

Proof: Let \( T_s = \{ j \in H(Q) \mid j \equiv i \mod a_1, j + s \in H(Q) \} \), then we have \( \lambda(s) = \#T_s \). Moreover, observe that

\[ H(Q) \setminus \vec{r}(Q) = \{-\nu(s_1^{(s)}, \nu_i^{(s)}) \mid i = 0, \ldots, a_1 - 1, k = 0, 1, \ldots \} \]

Therefore, for \( s \in \Gamma \) we have

\[ T_s = \{ j \in H(Q) \mid j \equiv i \mod a_1, j + s \in H(Q) \} \]

and

\[ \lambda(s) = \#T_s \]

where the third equality holds by Eq. (21). By the equalities above, we see

\[ \lambda(s) = \max \left\{ 0, -\nu(s_1^{(s)}, \nu_i^{(s)}, \nu_i^{(s)}, -\nu(s_1^{(s)}) \right\} \]

which proves the equality \( \nu(s) = \lambda(s) \).

Lee et al. [25] showed that their original decoding algorithm can correct up to \( \lfloor d_{LABO}(C_u) - 1 \rfloor / 2 \) errors, where \( d_{LABO}(C_u) = \min \{ \nu(s) \mid s \in H(Q), s \leq u \} \). However, they did not clarify whether or not \( d_{LABO}(C_u) \) is at least as large as the Goppa bound \( n - u \) except for one-point Hermite codes proved in [25] Proposition 12. It was unclear whether or not the original can correct errors half the Goppa bound like many other decoding algorithms, except for one-point Hermite codes. Besides, \( d_{AG}(C_u) \geq n - u \) [3, Proposition 37]. Proposition 6 implies that \( d_{LABO}(C_u) \) is equivalent to \( d_{AG}(C_u) \) for every one-point primal code \( C_u \), and therefore [3, Theorem 8] implies [25, Proposition 12]. In addition to this, we now know that the original decoding algorithm as well as our modified version can correct errors at least up to half the Goppa bound \( n - u \) for \( C_u \).

B. Lower Bound for the Number of Votes

In Sec. [IV-B] we discuss the number of votes \( \nu(s) \) which a candidate \( w_{s,i} \) receives. Since we study list decoding, we cannot assume the original transmitted codeword nor the error vector as in [25]. Nevertheless, the original theorems in [25] allow natural generalizations to the list decoding context.

Lemma 7: Fix \( s \in \Gamma \). For \( s' \in \Gamma^{\geq s} \), fix a sequence of \( w_{s'} \) chosen by the decoding algorithm, and define \( \rho^{(s)} \) corresponding to the chosen sequence of \( w_{s'} \). Fix \( \omega \in \mathbb{F}_q \).

Let \( d = (e_1, \ldots, e_n)^T \) be a nonzero vector with the minimum Hamming weight in the coset \( \rho^{(s)} - \text{ev}({\omega_1, \varphi_i}) + C_{s-1} \), where \( C_{s-1} \) is as defined in Eq. (1). Define

\[ J_s = \bigcap_{c_i \neq 0} L(P_i - \infty Q) \]

and

\[ = L\left(-\infty Q + \sum_{c_i \neq 0} P_i \right) \] (by 27).

Let \( \{e_1, \ldots, e_{n-1}\} \) be a Gröbner basis for \( J_s \) as an \( \mathbb{F}_q[x_1] \)-module with respect to \( \nu \) (for any integer \( s \)), such that \( \lambda(e_j) = \delta_j \).

Under the above notations, we have

\[ -\nu(\epsilon) + \nu(a_1, y_i) \geq \bar{a}_1 \delta_i, \]

\[ \min \{-\nu(\delta_i) + s, -\nu(\eta)\} \geq -\nu(d_{s', y_i}) \]

for \( i \) with \( w_{s,i} \neq \omega_s \), and

\[ \min \{-\nu(\delta_i) + s, -\nu(\eta)\} \geq -\nu(d_{s', y_i}) - a_1 \delta_i, \]

for \( i \) with \( w_{s,i} = \omega_s \).

Proof: The proof is the same as those of [25] Propositions 7 and 8, with replacing \( y_i \) in [25] by \( y_j \) and \( \delta_i \) in [25] by \( -\nu(\cdot) \).

The following lemma is a modification to [25] Proposition 9 for the list decoding.

Lemma 8: We retain notations from Lemma 7. We have

\[ a_1 \sum_{w_{s,i} = a_1} \bar{c}_i \geq a_1 \sum_{w_{s,i} \neq a_1} \bar{c}_i - 2a_1 \nu(\vec{r}) \]

and

\[ \sum_{w_{s,i} = a_1} \bar{c}_i \geq \sum_{w_{s,i} \neq a_1} -\nu(d_{s', y_i}) - \min \{-\nu(\delta_i) + s, -\nu(\eta)\} \]

and

\[ \geq \sum_{0 \leq i \leq a_1} -\nu(d_{s', y_i}) - \min \{-\nu(\delta_i) + s, -\nu(\eta)\} \]

and

\[ \sum_{w_{s,i} = a_1} \bar{c}_i \leq \sum_{w_{s,i} \neq a_1} -\nu(\epsilon) + \nu(a_1, y_i) \]

\[ \leq \sum_{0 \leq i \leq a_1} -\nu(\epsilon) + \nu(a_1, y_i) \].
Now we have a chain of inequalities
\[
\sum_{w_{i,j} \neq 0} a_i \tilde{c}_i - \sum_{w_{i,j} \neq 0} a_i \tilde{c}_i \\
\geq \sum_{0 \leq i < a_1} -v_Q(d_{f,f'}, y_{f'}) - \min\{-v_Q(e_i) + s, -v_Q(\eta_{f'})\} \\
- \sum_{0 \leq i < a_1} -v_Q(e_i) + v_Q(a_i y_i) \\
= \sum_{0 \leq i < a_1} -v_Q(d_{f,f'}, y_{f'}) - v_Q(a_i y_i) \\
- \min\{-v_Q(e_i) + s, -v_Q(\eta_{f'})\} + v_Q(e_i) \\
= \sum_{0 \leq i < a_1} -v_Q(\eta_{f'}) - v_Q(y_i) \\
+ \max\{+v_Q(e_i) - s, +v_Q(\eta_{f'})\} + v_Q(e_i) \\
= \sum_{0 \leq i < a_1} \max\{-v_Q(\eta_{f'}) + v_Q(y_i) - s, -v_Q(e_i) + v_Q(y_i)\} \\
- \sum_{0 \leq i < a_1} 2(-v_Q(e_i) + v_Q(y_i)) \\
\] 
(30)
where at Eq. (30) we used the equality
\[
\sum_{0 \leq i < a_1} -v_Q(d_{f,f'}, y_{f'}) \geq \sum_{0 \leq i < a_1} -v_Q(a_i y_i) \\
= \sum_{0 \leq i < a_1} -v_Q(d_{f,f'}, y_{f'}) \geq \sum_{0 \leq i < a_1} -2v_Q(y_i) \\
= a_1 n + \sum_{0 \leq i < a_1} -2v_Q(y_i) \\
= \sum_{0 \leq i < a_1} (-v_Q(\eta_{f'}) + v_Q(y_i)) + \sum_{0 \leq i < a_1} -2v_Q(y_i) \\
= \sum_{0 \leq i < a_1} -v_Q(\eta_{f'}) + \sum_{0 \leq i < a_1} -v_Q(y_i) \\
\]
shown in [25] Lemma 2 and Eq. (1)]. Finally note that
\[
\sum_{0 \leq i < a_1} 2(-v_Q(e_i) + v_Q(y_i)) \\
= \sum_{0 \leq i < a_1} 2a_1 \deg_{x_i}(e_i) = 2a_1 wt(\tilde{c}) \\
\]
by [25] Eq. (3)].

The following lemma is a modification to [25] Proposition 10] for list decoding, and provides a lower bound for the number of votes \(v(s)\) received by any candidate \(\omega_s \in F_q\), as indicated in the section title.

**Proposition 9:** We retain notations from Lemma 7. Let \(v(s)\) be as defined in Eq. (24). We have
\[
\sum_{w_{i,j} = 0} \tilde{c}_i \geq \sum_{w_{i,j} \neq 0} \tilde{c}_i - 2wt(\tilde{c}) + v(s). \\
\]
Proof: We have
\[
\sum_{0 \leq i < a_1} \max\{-v_Q(\eta_{f'}) + v_Q(y_i) - s, -v_Q(e_i) + v_Q(y_i)\} \\
\geq \sum_{0 \leq i < a_1} \max\{-v_Q(\eta_{f'}) + v_Q(y_i) - s, 0\} \\
\]
as \(-v_Q(e_i) + v_Q(y_i) \geq 0\) for \(0 \leq i < a_1\).

**C. Correctness of the Modified List Decoding Algorithm with the Third Iteration Termination Criterion**

In this subsection and the following sections, we shall prove that the proposed list decoding algorithm will find all the codewords within the Hamming distance \(\tau\) from the received word \(\bar{r}f\). Since the third iteration termination criterion is the easiest to analyze, we start with the third one.

Fix a sequence \(w_s\) for \(s \in \Gamma\). If \(wt(\bar{r}f - ev(\sum_{s \in \Gamma} w_s \varphi_s)) \leq \tau\) then the sequence \(w_s\) is found by the algorithm because of Proposition 9. When \(2\tau < d_{AG}(C_f)\), by Proposition 6 the decoding is not list decoding, and the algorithm just declares the sequence \(w_s\) as the transmitted information.

On the other hand, \(2\tau \geq d_{AG}(C_f)\), the found sequence could correspond to a codeword more distant than Hamming distance \(\tau\), and the algorithm examines the Hamming distance between the found codeword and the received word \(\bar{r}f\).

Since computing \(ev(f)\) for \(f \in L(\infty Q)\) needs many multiplications in \(F_q\), the algorithm checks some sufficient conditions to decide the Hamming distance between the found codeword and the received word \(\bar{r}f\). Let \(\bar{r}^{f-1} = (r^{f-1}_1, \ldots, r^{f-1}_n)\). When \(\alpha_0 = 0\) in Sec. III-F3] by Lemma 5 we have
\[
wt(\bar{r}f - ev(\sum_{s \in \Gamma} w_s \varphi_s)) = wt(\bar{r}^{f-1}) \\
\leq \sum_{i \neq \varnothing} v_p(\alpha_1) \\
\leq -v_Q(\alpha_1), \\
\]
because Eq. (29) and \(\alpha_0 = 0\) implies that \(v_p(\alpha_1) \geq 1\) for \(r^{f-1} \neq \varnothing\). By the above equation, \(-v_Q(\alpha_1) \leq \tau\) implies that the found codeword is within Hamming distance \(\tau\) from \(\bar{r}f\). This explains why the algorithm can avoid computation of the evaluation map \(ev\) in Step 1] in Sec. III-F3.

In order to explain Step 2 in Sec. III-F3, we shall show that the condition of Step 2 in Sec. III-F3] implies that \(wt(\bar{r}f - ev(\sum_{s \in \Gamma} w_s \varphi_s)) > \tau.\) Suppose that \(wt(\bar{r}f - ev(\sum_{s \in \Gamma} w_s \varphi_s)) \leq \tau\). Then there exists \(\beta_1 \in L(\infty Q)\) such that \(v_p(\beta_1) \geq 1\) for \(r^{f-1} \neq \varnothing\), \(-v_Q(\beta_1) \leq \tau + g, \) and \(\beta_1 \in I_{f-1}^{-1}\). Because the leading term of \(\beta_1 z\) must be divisible by \(1\) by the property of Gröbner bases, we must have \(-v_Q(\alpha_1) \leq -v_Q(\beta_1)\). This explains why the algorithm can avoid computation of the evaluation map \(ev\) in Step 2 in Sec. III-F3.

Otherwise, the algorithm computes the Hamming distance between the found codeword and \(\bar{r}f\) in Steps 3 and 4 in Sec. III-F3.

**D. Correctness of the Modified List Decoding Algorithm with the Second Iteration Termination Criterion**

We shall explain why the second criterion in Sec. III-F2 correctly finds the required codewords. For explanation, we present slightly rephrased version of facts in [6].

**Lemma 10:** [6] Lemma 2.3] Let \(\beta_1 z + \beta_0 \in I_{f-2}\) with \(\varpi(\beta_1 z + \beta_0) = \varpi(\beta_1 z)\) with respect to \(<_s\) and \(-v_Q(\beta_1) < n - \tau - s\). If there exists \(f \in L(\infty Q)\) such that \(wt(ev(f) - \bar{r}f^{s}) \leq \tau\), then we have \(f = -\beta_0/\beta_1\).

**Proof:** Observe that \(\varpi(\beta_1 z + \beta_0) = \varpi(\beta_1 z)\) implies that \(-v_Q(\beta_0) \leq -v_Q(\beta_1) + s < n - \tau.\) The claim of Lemma 10 is
equivalent to \[ F \] Lemma 2.3] with A = (n−τ−1)Q and G = sQ. Note that the assumption \( \text{deg} A > (n + \text{deg} G)/2 + g - 1 \) was not used in \[ F \] Lemma 2.3] but only in \[ F \] Lemma 2.4].

Note that the following proposition was essentially proved in \[ F \] Proposition 2.10], [23 Sec. 14.2], and [39 Theorem 2.1] with \( b = 1 \).

**Proposition 11.** Let \( a_0 \) and \( a_1 \) be as in Sec. [III-F2]. If \( s < n - g - 2\tau \) and there exists \( f \in L(sQ) \) such that \( w(t) ≤ F\) at \( τ \), then we have \( f = -a_0/a_1 \).

**Proof:** Let \( g \in L(∞Q) \) such that \( g(P_i) = 0 \) if \( f(P_i) \neq f_i^{(s)} \), and assume that \( g \) has the minimum pole order at \( Q \) among such elements in \( L(∞Q) \). Then \( -\nu_0(g) ≤ τ + g \). One has that \( g - f \) is divisible by \( \text{Gr}(g - f) = \text{Gr}(g) \) with respect to \( s \). By the property of Gröbner bases, \( \text{Gr}(g) \) is divisible by \( \text{Gr}(f_i^{(s)}) \) for some \( i \), which implies \( -\nu_0(a_1) ≤ -\nu_0(g) ≤ τ + g \). By Lemma [10] we have \( f = -a_0/a_1 \).

We explain how the procedure in Sec. [III-F2] works as desired. When the condition in Step 1 in Sec. [III-F2] is true, then there cannot be codeword within Hamming distance \( τ \) from \( f_i^{(s)} \) by the same reason as Sec. [IV-C]. So the algorithm stops processing with \( f_i^{(s)} \).

When \( 2\tau < d_{AG}(C_T) \), then the algorithm declares \( -a_0/a_1 + \sum_{c \neq t \in T^O} w_c\varphi_c \) as the unique codeword. When \( 2\tau ≥ d_{AG}(C_T) \), then the algorithm examines the found codeword close enough to \( f \) in Steps [4a] and [4b] in Sec. [III-F2]. When \( -\nu_0(a_1) ≤ τ \) we can avoid computation of the evaluation map \( ev \) by the same reason as Sec. [IV-C] which is checked at Step [4a]. Otherwise we compute the codeword vector at Step [4b] and examine its Hamming distance to \( f_i^{(s)} \).

By Proposition [11], the codeword must be found at \( s = \max\{s' ∈ Γ | s' < n - 2\tau + g\} \). Therefore, we do not execute the iteration at \( s < \max\{s' ∈ Γ | s' < n - 2\tau + g\} \).

**E. Correctness of the Modified List Decoding Algorithm with the First Iteration Termination Criterion**

We shall explain why the first criterion in Sec. [III-F1] correctly finds the required codewords. The idea behind the first criterion is that there cannot be another codeword within Hamming distance \( τ \) from \( f_i^{(s)} \) when the algorithm already found one. So the algorithm can stop iteration with smaller \( s \) once a codeword is found as \( ev(-a_0/a_1 + \sum_{c \neq t \in T^O} w_c\varphi_c) \).

The algorithm does not examine conditions when \( -\nu_0(a_1) > τ + g \) by the same reason as Secs. [IV-C] and [IV-D]. When \( 2\tau < d_{AG}(C_T) \), then the algorithm declares \( -a_0/a_1 + \sum_{c \neq t \in T^O} w_c\varphi_c \) as the unique codeword.

When \( 2\tau ≥ d_{AG}(C_T) \), then the algorithm examines the found codeword close enough to \( f \) in Steps [2a] and [2c] in Sec. [III-F1].

When \( -\nu_0(a_1) ≤ τ \) we can avoid computation of the evaluation map \( ev \) by the same reason as Sec. [IV-C] which is checked at Step [2a]

By Proposition [11], the codeword must be found at some \( s ≥ \max\{s' ∈ Γ | s' < n - 2\tau + g\} \). Therefore, we do not execute the iteration at \( s < \max\{s' ∈ Γ | s' < n - 2\tau + g\} \).

**F. Upper Bound on the Number of Iterations**

For each \( s \) satisfying \( 2\nu(s) ≤ τ \), the number of accepted candidates satisfying Eq. (26) can be \( q \). Therefore, we have upper bounds for the number of iterations, counting executions of Rebasing in Sec. [III-G5] as

\[ \#(s ∈ H(Q) | \max Γ ≤ s < N) + \exp(q(\#(s ∈ Γ | 2\nu(s) ≥ τ))) \]

for the third criterion for judging termination, where \( \exp(q(\#(s ∈ Γ | 2\nu(s) ≥ τ))) \)

for the first and the second criteria for judging termination.

Observe that the list decoding can be implemented as \( \exp(q(\#(s ∈ Γ | 2\nu(s) ≥ τ))) \)

for the first and the second criteria for judging termination.

We have provided an upper bound on the number of multiplications and divisions at each step of the proposed algorithm. We simulated 1,000 transmissions of codewords with the one-point primal codes on Klein quartic over \( F_8 \) with \( n = 23 \) by using Examples [1] and [3], the one-point Hermitian codes over \( F_{16} \) with \( n = 64 \), and the one-point primal codes on the curve in Example [2] over \( F_9 \) with \( n = 77 \).

The program is implemented on the Singular computer algebra system [20]. The program used for this simulation is included in the source file of this eprint.

In the execution, we counted the number of iterations, (executions of Rebasing in Sec. [III-G5]), the sum of upper bounds on the number of multiplications and divisions given in Eqs. (9), (16), (17), (18), (19), (20), (23), (27) and (28), and the number of codewords found. Note also that Eq. (11) instead of Eq. (13) is used.

The parameter \( τ \) is set to the same as the number of generated errors in each simulation condition. \( N \) or \( R \) in the number of errors in Table [1] indicates that the error vector is generated toward another codeword nearest from the transmitted codeword or completely randomly, respectively. The distribution of codewords is uniform on \( C_T \). That of error vectors is uniform on the vectors of Hamming weight \( τ \).

In the code construction, we always try to use the Feng-Rao improved construction. Specifically, for a given designed distance \( δ \), we choose \( Γ = \{s ∈ H(Q) | \lambda(s) = ν(s) ≥ δ\} \), and construct \( C_T \) of Eq. (4). In the following, the designed distance is denoted by \( d_{AG}(C_T) \). It can be seen from Tables [1][3] and the following subsections that the computational complexity of the proposed algorithm tends to explode when the number of errors exceeds the error-correcting capability of the Guruswami-Sudan algorithm [21].

**B. Comparison among the Three Proposed Termination Criteria**

In Sec. [III-G] we proposed three criteria for terminating iteration of the proposed algorithm. From Tables [1][3][1] one
can see the following. The first criterion has the smallest number of iterations, and the second is the second smallest. On the other hand, the first criterion has the largest number of multiplications and divisions. The second and the third have the similar numbers. Only the first criterion was proposed in \([18]\) and we see that the new criteria are better than the old one.

The reason is as follows: The computation of quotient \(a_0/\alpha_1\) at Step 1 in Sec. \(\text{III-F1}\) is costlier than updating \(f_1^{(s)}\) and \(g_1^{(s)}\) in Sec. \(\text{III-G3}\) and the first criterion computes \(a_0/\alpha_1\) many times, which cancels the effect of decrease in the number of iterations. On the other hand, the second criterion computes \(a_0/\alpha_1\) only once, so it has the smaller number of multiplications and divisions than the first.

The second criterion is faster when \(2\tau < d_{\text{AG}}(C_T)\), while the third tends to be faster when \(2\tau \geq d_{\text{AG}}(C_T)\). In addition to this, the ratio of the number of iterations in the second criterion to that of the third is smaller with \(2\tau < d_{\text{AG}}(C_T)\) than with \(2\tau \geq d_{\text{AG}}(C_T)\). We speculate the reason behind them as follows: When \(2\tau \geq d_{\text{AG}}(C_T)\) and a wrong candidate is chosen at Eq. \((26)\), after several iterations of Secs. \(\text{III-F}\) and \(\text{III-G}\) we often observe in our simulation that no candidate satisfies Eq. \((26)\) and the iteration stops automatically. Under such situation, the second criterion does not help much to decrease the number of iterations nor the computational complexity when a wrong candidate is chosen at Eq. \((26)\), and there are many occasions at which a wrong candidate is chosen at Eq. \((26)\) when \(2\tau \geq d_{\text{AG}}(C_T)\). On the other hand, when \(2\tau < d_{\text{AG}}(C_T)\), the second criterion helps to determine the transmitted information earlier than the third.

### C. Klein Quartic, \((d_{\text{AG}}(C_T), \tau) = (4, 1)\) or \((10, 4)\)

We can use \([4], [6], [9], [10], [29]\) to decode this set of parameters. It is essentially the forward elimination in the Gaussian elimination, and it takes roughly \(n^3/3\) multiplications. In this case \(n^3/3 = 4055\). The proposed algorithm has the lower complexity than \([4], [6], [9], [10], [29]\).

### D. Klein Quartic, \((d_{\text{AG}}(C_T), \tau) = (4, 2)\) or \((4, 3)\)

The code is \(C_u\) with \(u = 20\), \(\dim C_u = 18\). There is no previously known algorithm that can handle this case.

### E. Klein Quartic, \((d_{\text{AG}}(C_T), \tau) = (10, 5)\)

The code is \(C_u\) with \(u = 13\), \(\dim C_u = 11\). According to \([5, \text{Fig. 1}]\), we can use the original Guruswami-Sudan \([21]\) but it seems that its faster variant cannot be used. We need the multiplicity 7 to correct 5 errors. We have to solve a system of \(23(7+1)/2 = 644\) linear equations. It takes \(644^3/3 = 89,029,994\) multiplications in \(F_8\). The proposed algorithm is much faster.

### F. Klein Quartic, \((d_{\text{AG}}(C_T), \tau) = (10, 6)\)

The code is \(C_u\) with \(u = 13\), \(\dim C_u = 11\). There is no previously known algorithm that can handle this case.

### G. Hermitian, \((d_{\text{AG}}(C_T), \tau) = (6, 2)\) or \((20, 9)\)

We can use the BMS algorithm \([37], [38]\) for this case. The complexity of \([37], [38]\) is estimated as \(O(a_1n^2)\) and \(a_1n^2 = 24,576\). The complexity of the proposed algorithm seems comparable to \([37], [38]\). However, we are not sure which one is faster.

### H. Hermitian, \((d_{\text{AG}}(C_T), \tau) = (6, 3)\) or \((6, 4)\)

The code becomes the Feng-Rao improved code with designed distance 6. Its dimension is 55. In order to have the same dimension by \(C_u\) we have to set \(u = 60\), whose AG bound \([13]\) is 4 and the Guruswami-Sudan can correct up to 2 errors. The proposed algorithm finds all codewords in the improved code with 3 and 4 errors.

---

**TABLE I**

Decoding results of codes on the Klein quartic \((F_q = F_8, g = 3 and n = 23)\)

| #′ | \(d_{\text{AG}}(C_T)\) | # Errors \(= \tau\) | Termination Criterion \(\text{in Sec. III-F}\) | # Iterations | # Multiplications & Divisions in \(F_q\) | # Codewords Found |
|----|------------------|------------------|---------------------------------|----------------|-------------------------------|------------------|
| 18 | 4                | 1                | 1st                             | Avg. 8.00      | Avg. 1170.09                  | 1.00 1           |
|    |                  |                  |                                 | 1st 11.00      | 844.98                       |                  |
|    |                  |                  |                                 | 3rd 26.00      | 976.32                       |                  |
| 2  |                  |                  |                                 |               |                               |                  |
|    |                  |                  |                                 | 1st 190.05     | 26205.96                     | 1.34 3           |
|    |                  |                  |                                 | 2nd 200.64     | 8349.67                      |                  |
|    |                  |                  |                                 | 3rd 219.07     | 7813.76                      |                  |
| 3  |                  |                  |                                 |               |                               |                  |
|    |                  |                  |                                 | 1st 11996.34   | 1626658.69                   | 19.75 28         |
|    |                  |                  |                                 | 2nd 12055.56   | 608535.03                    |                  |
|    |                  |                  |                                 | 3rd 12436.00   | 580504.03                    |                  |
| 11 | 10               | 4                | 1st                             | Avg. 14.64     | Avg. 1324.76                  | 1.00 1           |
|    |                  |                  |                                 | 2nd 16.64      | 1161.52                      |                  |
|    |                  |                  |                                 | 3rd 25.64      | 1329.07                      |                  |
| 5  |                  |                  |                                 |               |                               |                  |
|    |                  |                  |                                 | 1st 35.20      | 3673.78                      | 1.00 1           |
|    |                  |                  |                                 | 2nd 38.20      | 2915.04                      |                  |
|    |                  |                  |                                 | 3rd 45.41      | 3072.08                      |                  |
| 6  |                  |                  |                                 |               |                               |                  |
|    |                  |                  |                                 | 1st 1507.95    | 164274.07                    | 1.11 3           |
|    |                  |                  |                                 | 2nd 1511.28    | 113592.10                    |                  |
|    |                  |                  |                                 | 3rd 1535.23    | 113472.30                    |                  |

---

10
I. Hermitian, \((d_{AG}(C_T), \tau) = (20, 10)\)

The code is \(C_u\) with \(u = 44\). The required multiplicity is 11, and the required designed list size is 14. The fastest algorithm for the interpolation step seems\[5, 5\] Example 4] estimates the complexity of their algorithm as \(O(\lambda^2 n^2 (\log \lambda n)^3 \log(\log \lambda n))\), where \(\lambda\) is the designed list size. Ignoring the log factor and assuming the scaling factor one in the big-O notation, the number of multiplications and divisions is \(\lambda^2 n^2 = 2, 202, 927, 104\). The proposed algorithm needs much fewer number of multiplications and divisions in \(F_{16}\).

J. Hermitian, \((d_{AG}(C_T), \tau) = (20, 11)\)

The Guruswami-Sudan algorithm\[21\] can correct up to 10 errors and there seems no previously known algorithm that can handle this case.

K. Garcia-Stichtenoth (Example 2), \((d_{AG}(C_T), \tau) = (6, 2), (10, 4), \text{ or } (20, 9)\)

We can use\[4, 6, 9, 10, 29\] to decode this set of parameters. It is essentially the forward elimination in the Gaussian elimination, and it takes roughly \(n^2/3\) multiplications. In this case \(n^2/3 = 152, 177\). The proposed algorithm has the lower complexity than\[4, 6, 9, 10, 29\].

L. Garcia-Stichtenoth (Example 2), \((d_{AG}(C_T), \tau) = (6, 3)\)

This is a Feng-Rao improved code with dimension 58. In order to realize a code with the same dimension, we have to set \(u = 79\) in \(C_u\). The Guruswami-Sudan algorithm\[21\] can correct no error in this set of parameters. There seems no previously known algorithm that can handle this case.

M. Garcia-Stichtenoth (Example 2), \((d_{AG}(C_T), \tau) = (10, 5)\)

This is a Feng-Rao improved code with dimension 52. In order to realize a code with the same dimension, we have to set \(u = 73\) in \(C_u\). The Guruswami-Sudan algorithm\[21\] can correct 2 errors in this set of parameters. There seems no previously known algorithm that can handle this case.

N. Garcia-Stichtenoth (Example 2), \((d_{AG}(C_T), \tau) = (20, 10)\)

This is an ordinary one-point AG code \(C_u\) with \(u = 58\) and dimension 37. The Guruswami-Sudan algorithm\[21\] can correct 10 errors with the multiplicity 154 and the designed list size 178. We have to solve a system of \(77 \times (154 + 1)154/2 = 918, 995\) linear equations. It takes \(918, 995^3/3 = 258, 712, 963, 551, 308, 291\) multiplications in \(F_9\). The proposed algorithm is much faster.
The Guruswami-Sudan algorithm \(21\) can correct up to 10 errors and there seems no previously known algorithm that can handle this case.

VI. CONCLUSION

In this paper, we modified the unique decoding algorithm for plane AG codes in \(25\) so that it can support one-point AG codes on any curve, and so that it can do the list decoding. The error correction capability of the original \(25\) and our modified algorithms are also analyzed.

We also proposed procedures to compute products and quotients in coordinate ring of affine algebraic curves, and by using those procedures we demonstrated that the modified decoding algorithm can be executed quickly. Specifically, its computational complexity is comparable to the BMS algorithm \(37, 38\) for one-point Hermitian codes, and much faster than the standard list decoding algorithms \(5, 21\) for several cases.

The original decoding algorithm \(25\) allows parallel implementation on circuits like the Kötter architecture \(24\). Our modified algorithm retains this advantage. Moreover, if one can afford large circuit size, the proposed list decoding algorithm can be executed as quickly as the unique decoding algorithm by parallel implementation on a circuit.

REFERENCES

[1] W. W. Adams and P. Loustaunau, An Introduction to Gröbner Bases, ser. Graduate Studies in Mathematics. Providence, RI: American Mathematical Society, 1994, vol. 3.

[2] M. Ali and M. Kuijper, "A parametric approach to list decoding of Reed-Solomon codes using interpolation," IEEE Trans. Inform. Theory, vol. 57, no. 10, pp. 6718–6728, Oct. 2011, arXiv:1011.1040.

[3] H. E. Andersen and O. Geil, "Evaluation codes from order domain theory," Finite Fields Appl., vol. 14, no. 1, pp. 92–123, Jan. 2008.

[4] P. Beelen, "The order bound for general algebraic geometric codes," Finite Fields Appl., vol. 13, no. 3, pp. 665–680, Jul. 2007.

[5] P. Beelen and K. Brander, "Efficient list decoding of a class of algebraic-geometry codes," Adv. Math. Commun., vol. 4, no. 4, pp. 485–518, 2010.

[6] P. Beelen and T. Høholdt, "The decoding of algebraic geometry codes," in Advances in Algebraic Geometry Codes, ser. Coding Theory and Cryptology, E. Martínez-Moro, C. Munuera, and D. Ruano, Eds. World Scientific, 2008, vol. 5, pp. 49–98.

[7] M. Bras-Amorós and M. E. O’Sullivan, "The correction capability of the Berlekamp-Massey-Sakata algorithm with majority voting." Applicable Algebra in Engineering, Communication and Computing, vol. 17, no. 5, pp. 315–335, Oct. 2006.

[8] H. Chen, "On the number of correctable errors of the Feng-Rao decoding algorithm for AG codes," IEEE Trans. Inform. Theory, vol. 45, no. 5, pp. 1709–1712, Jul. 1999.

[9] I. M. Duursma, R. Kirov, and S. Park, "Distance bounds for algebraic geometric codes," J. Pure Appl. Algebra, vol. 215, no. 8, pp. 1863–1878, Aug. 2011.

[10] I. M. Duursma and S. Park, "Coset bounds for algebraic geometric codes," Finite Fields Appl., vol. 16, no. 1, pp. 36–55, Jan. 2010.

[11] D. Eisenbud, Commutative Algebra with a View Toward Algebraic Geometry, ser. Graduate Texts in Mathematics. Berlin: Springer-Verlag, 1995, vol. 150.

[12] H. Elbrønd Jensen, R. R. Nielsen, and T. Høholdt, "Performance analysis of a decoding algorithm for algebraic-geometric codes," IEEE Trans. Inform. Theory, vol. 45, no. 5, pp. 1712–1717, Jul. 1999.

[13] G. L. Feng and T. R. N. Rao, "Decoding algebraic geometric codes up to the designed minimum distance," IEEE Trans. Inform. Theory, vol. 39, no. 1, pp. 36–47, Jan. 1993.

[14] ——, "Improved geometric Goppa codes part I: basic theory," IEEE Trans. Inform. Theory, vol. 41, no. 6, pp. 1678–1693, Nov. 1995.

[15] M. Fujisawa, H. Matsui, M. Kurihara, and S. Sakata, "With a higher probability one can correct errors up to half the designed distance for primal codes from curves," in Proc. SITA2006, Hakodate, Hokkaido, Japan, Nov. 2006, pp. 101–104.

[16] A. Garcia and H. Stichtenoth, "A tower of Artin-Schreier extensions of function fields, attaining the Drinfeld-Vladut bound," Invent. Math., vol. 121, no. 1, pp. 211–222, Jul. 1995.

[17] O. Geil, C. Munuera, D. Ruano, and F. Torres, "On the order bounds for one-point AG codes," Adv. Math. Commun., vol. 5, no. 3, pp. 489–504, 2011.

[18] O. Geil, R. Matsumoto, and D. Ruano, "List decoding algorithms based on Gröbner bases for general one-point AG codes," Proc. ISIT 2012, submitted for possible presentation, Boston, MA, USA, Jul. 2012, arXiv:1201.6248.

[19] O. Geil and R. Pellikaan, "On the structure of order domains," Finite Fields Appl., vol. 8, no. 3, pp. 369–396, Jul. 2002.

[20] G.-M. Greuel, G. Pfister, and H. Schönemann, "Singular 3.0," Centre for Computer Algebra, University of Kaiserslautern, A Computer Algebra
[21] V. Guruswami and M. Sudan, “Improved decoding of Reed-Solomon and algebraic-geometry codes,” IEEE Trans. Inform. Theory, vol. 45, no. 4, pp. 1757–1767, Sep. 1999.

[22] T. Høholdt and R. Pellikaan, “On the decoding of algebraic-geometric codes,” IEEE Trans. Inform. Theory, vol. 41, no. 6, pp. 1589–1614, Nov. 1995.

[23] J. Justesen and T. Høholdt, A Course in Error-Correcting Codes, ser. EMS Textbooks in Mathematics. Zürich, Switzerland: European Mathematical Society Publishing House, 2004.

[24] R. Kötter, “A fast parallel implementation of a Berlekamp-Massey algorithm for algebraic-geometric codes,” IEEE Trans. Inform. Theory, vol. 44, no. 4, pp. 1353–1368, Jul. 1998.

[25] K. Lee, M. Bras-Amorós, and M. E. O’Sullivan, “Unique decoding of plane AG codes via interpolation,” 2012, IEEE Trans. Inform. Theory, Early Access. [Online]. Available: http://dx.doi.org/10.1109/TIT.2012.2182757, arXiv:1110.6251.

[26] K. Lee and M. E. O’Sullivan, “List decoding of Hermitian codes using Gröbner bases,” J. Symbolic Comput., vol. 44, no. 12, pp. 1662–1675, Dec. 2009.

[27] R. Matsumoto and S. Miura, “Finding a basis of a linear system with pairwise distinct discrete valuations on an algebraic curve,” J. Symbolic Comput., vol. 30, no. 3, pp. 309–323, Sep. 2000.

[28] ——, “On construction and generalization of algebraic geometry codes,” in Proc. Algebraic Geometry, Number Theory, Coding Theory, and Cryptography, T. Katsura et al., Eds., Univ. Tokyo, Japan, Jan. 2000, pp. 3–15. [Online]. Available: http://www.rmatsumoto.org/repository/weight-construct.pdf.

[29] ——, “On the Feng-Rao bound for the $L$-construction of algebraic geometry codes,” IEICE Trans. Fundamentals, vol. E83-A, no. 5, pp. 926–930, May 2000.

[30] C. P. McKeague, Elementary Algebra, 9th ed. Florence, KY 41022-6904, USA: Brooks Cole, 2012.

[31] S. Miura, “Algebraic geometric codes on certain plane curves,” Electronics and Communications in Japan (Part III: Fundamental Electronic Science), vol. 76, no. 12, pp. 1–13, Dec. 1993, (original Japanese version published as Trans. IEICE, vol. J75-A, no. 11, pp. 1735–1745, Nov. 1992).

[32] ——, “Linear codes on affine algebraic curves,” Trans. IEICE, vol. J81-A, no. 10, pp. 1398–1421, Oct. 1998 (Japanese).

[33] J. C. Rosales and P. A. García-Sánchez, Numerical semigroups, ser. Developments in Mathematics. New York: Springer, 2009, vol. 20.

[34] K. Saints and C. Heegard, “Algebraic-geometric codes and multidimensional cyclic codes: A unified theory and algorithms for decoding using Gröbner bases,” IEEE Trans. Inform. Theory, vol. 41, no. 6, pp. 1733–1751, Nov. 1995.

[35] S. Sakata, “On fast interpolation method for Guruswami-Sudan list decoding of one-point algebraic-geometry codes,” in Proc. AAECC-14, ser. Lecture Notes in Computer Science, S. Boztas and I. E. Shparlinski, Eds., vol. 2227. Melbourne, Australia: Springer-Verlag, Nov. 2001, pp. 172–181.

[36] ——, “Multivariate interpolation and list decoding,” in Proc. 3rd Asian-European Workshop on Information Theory, K. Kobayashi and H. Morita, Eds. Kamogawa, Chiba, Japan: Society of Information Theory and its Applications, Jun. 2003, pp. 28–31.

[37] S. Sakata, H. Elbrønd Jensen, and T. Høholdt, “Generalized Berlekamp-Massey decoding of algebraic-geometric codes up to half the Feng-Rao bound,” IEEE Trans. Inform. Theory, vol. 41, no. 5, pp. 1762–1768, Sep. 1995.

[38] S. Sakata, J. Justesen, Y. Madelung, H. Elbrønd Jensen, and T. Høholdt, “Fast decoding of algebraic-geometric codes to the designed minimum distance,” IEEE Trans. Inform. Theory, vol. 41, no. 5, pp. 1672–1677, Sep. 1995.

[39] M. A. Shokrollahi and H. Wasserman, “List decoding of algebraic-geometric codes,” IEEE Trans. Inform. Theory, vol. 45, no. 2, pp. 432–437, Mar. 1999.

[40] D. Umehara and T. Uyematsu, “One-point algebraic geometric codes from Artin-Schreier extensions of Hermitian function fields,” IEICE Trans. Fundamentals, vol. E81-A, no. 10, pp. 2025–2031, Oct. 1998.

[41] W. V. Vasconcelos, Computational Methods in Commutative Algebra and Algebraic Geometry, ser. Algorithms and Computation in Mathematics. Berlin: Springer-Verlag, 1998, vol. 2.

[42] C. Voss and T. Høholdt, “An explicit construction of a sequence of codes attaining the Tsfasman-Vlăduţ-Zink bound,” IEEE Trans. Inform. Theory, vol. 43, no. 1, pp. 128–135, Jan. 1997.