List Decoding of q-ary Reed-Muller Codes

Ruud Pellikaan, Xin-Wen Wu *

Expanded version of the paper that appeared in:
IEEE Trans. Inform. Theory, vol. 50, No. 4, April 2004, pp. 679-682.

Abstract

The q-ary Reed-Muller codes $\mathcal{RM}_q(u,m)$ of length $n=q^m$ are a generalization of Reed-Solomon codes, which allow polynomials in m variables to encode the message. Using an idea of reducing the multivariate case to univariate case, randomized list-decoding algorithms for Reed-Muller codes were given in [1] and [27]. The algorithm in [27] is an improvement of the algorithm in [1], it works for up to $E<n(1-\sqrt{2u/q})$ errors but is applicable only to codes $\mathcal{RM}_q(u,m)$ with $u<q/2$. In this paper, we will propose some deterministic list-decoding algorithms for q-ary Reed-Muller codes. Viewing q-ary Reed-Muller codes as codes from order domains, we present a list-decoding algorithm for q-ary Reed-Muller codes, which is a straightforward generalization of the list-decoding algorithm of Reed-Solomon codes in [9]. The algorithm works for up to $n(1-\sqrt[m+1]{u/q})^m-1$ errors, and it is applicable to codes $\mathcal{RM}_q(u,m)$ with $u<q$. The algorithm can be implemented to run in time polynomial in the length of the codes. Following [12], we show that q-ary Reed-Muller codes are subfield subcodes of Reed-Solomon codes. We then present a second list-decoding algorithm for q-ary Reed-Muller codes. This algorithm works for codes with any rates, and achieves an error-correction bound $n(1-\sqrt{(n-d)/n})-1$. So the second algorithm achieves a better error-correction bound than the algorithm in [27], since when u is small, $n(1-\sqrt{(n-d)/n})=n(1-\sqrt{u/q})$. The implementation of the second algorithm requires $O(n)$ field operations in $\mathbb{F}_q$ and $O(n^3)$ field operations in $\mathbb{F}_{q^m}$ under some assumption. Also, we prove that q-ary Reed-Muller codes can be described as one-point AG codes. And using the algorithm of AG codes in [9], we give a third list-decoding

*Ruud Pellikaan is with the Department of Mathematics and Computing Science, Technical University of Eindhoven, P.O. Box 513, 5600 MB Eindhoven, The Netherlands. Email: g.r.pellikaan@TUE.nl. Xin-Wen Wu is with the Academy of Mathematics and System Sciences, Chinese Academy of Sciences, Beijing, 100080, China. Email: wxw@math08.math.ac.cn. This work was supported in part by the National Natural Science Foundation of China under grant 10071086.

algorithm for Reed-Muller codes. The third algorithm achieves an error-correction bound $n(1-\sqrt{u(q+1)^{m-1}/n})$. The time complexity of the third algorithm is also bounded from above by polynomials in the length of the codes.

Index Terms: List decoding, Guruswami-Sudan algorithm, q-ary Reed-Muller codes, order domain, subfield subcodes, one-point AG codes.

1 Introduction

Let C be a $[n, k, d]$ code over the finite field $\mathbb{F}_q$ with q elements. Let $E < n$ and b be positive integers, C is called (E, b)-decodable if every Hamming sphere of radius E in $\mathbb{F}_q^n$ contains at most b codewords. For any received word $\mathbf{y} = (y_1, \cdots, y_n) \in \mathbb{F}_q^n$, a codeword $\mathbf{c}$ contained in the Hamming sphere centered at $\mathbf{y}$ with radius E, i.e., $d(\mathbf{c}, \mathbf{y}) \leq E$, is called a *$E$-consistent codeword*. A list-decoding algorithm is an algorithm which tries to construct a list which includes all the E-consistent codewords. The parameter E can be greater than the traditional error-correction bound $\lfloor \frac{d-1}{2} \rfloor$. Thus, a list-decoding algorithm makes it possible to recover the information from errors beyond the traditional error-correction bound.

List decoding was introduced by Elias [5] and Wozencraft [29]. In [26], Sudan proposed a list-decoding algorithm for Reed-Solomon codes. Algebraic-geometric (AG) codes are a generalization of Reed-Solomon codes. In 1999, Shokrollahi and Wasserman generalized Sudan's algorithm and derived a list-decoding algorithm for AG codes [22]. Both of the list-decoding algorithms in [26] and [22] work only for codes of low rates. Guruswami and Sudan [9] later proposed improved list-decoding algorithms for Reed-Solomon and AG codes. The algorithms of Guruswami and Sudan have better error-correction capabilities than algorithms in [26] and [22], and work for codes with any rates. Besides AG codes, Reed-Solomon codes can be generalized in another way, by allowing multivariate polynomials to encode the message. These generalized codes are known as Reed-Muller (RM) codes. Let $\mathbb{F}_q[X_1, \cdots, X_m]$ be the ring of polynomials in m variables with coefficients in $\mathbb{F}_q$. Let $P_1, \cdots, P_n$ be an enumeration of the points of $\mathbb{F}_q^m$, where $n = q^m$. The q-ary Reed-Muller code $\mathcal{RM}_q(u, m)$ of order u in m variables is defined as

$$\mathcal{RM}_q(u, m) = \{(f(P_1), \cdots, f(P_n)) \mid f \in \mathbb{F}_q[X_1, \cdots, X_m],\ \deg(f) \leq u\}.$$

When $m = 1$, we get a Reed-Solomon code $C_{RS}(u) = \mathcal{RM}_q(u, 1)$. The list-decoding algorithm for Reed-Solomon code $C_{RS}(u)$ in [9] works for up to $E < n(1 - \sqrt{(n-d)/n})$ errors,

where n and d are the length and minimum distance of the code, respectively.

However, the algorithm in [9] for Reed-Solomon codes has no generalization in any straightforward way to decode Reed-Muller codes before this work. In [1] and [27] using an idea of reducing the multivariate case to the univariate case, randomized list-decoding algorithms for Reed-Muller codes $\mathcal{RM}_q(u,m)$ were given. The algorithm in [27] improves upon the algorithm in [1], and it works for up to $E < n(1-c\sqrt{u/q})$ errors, where c is a constant greater than $\sqrt{2}$. The algorithm works only for codes $\mathcal{RM}_q(u,m)$ with $u < q/2$.

The polynomial ring $\mathbb{F}_q[X_1, \cdots, X_m]$ can be viewed as a special example of *order domains*. In [7] the structure of order domains and *codes from order domains* were studied. It is well known that the codes from order domains include Reed-Solomon codes, AG codes, and q-ary Reed-Muller codes as special cases. Root-finding (or factorization) of a polynomial with coefficients in a ring is an important step of the known list-decoding algorithms. In [30], a root-finding algorithm was proposed for finding the roots in certain spaces of polynomials with coefficients in order domains, which gives a potential application to list decodings of the codes from order domains including Reed-Muller codes.

In this paper, viewing q-ary Reed-Muller codes as codes from order domains, we present a list-decoding algorithm for q-ary Reed-Muller codes, which is a straightforward generalization of the list-decoding algorithm of Reed-Solomon codes by Guruswami and Sudan [9]. The algorithm works for up to $n(1-\sqrt[m+1]{u/q})^m - 1$ errors, and it is an effective algorithm for codes $\mathcal{RM}_q(u,m)$ with $u < q$. The algorithm can be implemented to run in time polynomial in the length of the codes.

Following [12], we show that q-ary Reed-Muller codes are subfield subcodes of Reed-Solomon codes. Using the list-decoding algorithm of Reed-Solomon codes, we then present a second list-decoding algorithm for q-ary Reed-Muller codes. This algorithm works for Reed-Muller codes with any rates, and achieves an error-correction bound $n(1-\sqrt{(n-d)/n})-1$. It is known that when $u < q$, the minimum distance $d = (q-u)q^{m-1}$ for the code $\mathcal{RM}_q(u,m)$ of length $n = q^m$, so we have $n(1-\sqrt{(n-d)/n}) = n(1-\sqrt{u/q})$. Thus, this algorithm works for up to $E < n(1-\sqrt{u/q})$ errors when the rate is small, which is better than the error-correction bound $n(1-\sqrt{2u/q})$ of the algorithm in [27]. In the case of fixed rate of codes, the implementation of the second algorithm requires $O(n)$ field operations in $\mathbb{F}_q$ and $O(n^3)$ field operations in $\mathbb{F}_{q^m}$.

Also, we prove that q-ary Reed-Muller codes can be described as one-point AG codes.

Then using list-decoding algorithm of AG codes by Guruswami and Sudan [9], we give a third list-decoding algorithm for q-ary Reed-Muller codes. The third algorithm achieves a error-correction bound $n(1-\sqrt{u(q+1)^{m-1}/n})$. Since $n=q^m$, when q is large this bound is very close to $n(1-\sqrt{u/q})$, the error-correction bound of the second algorithm. The time complexity of the third algorithm is also bounded from above by polynomials in the length of the codes.

This work is organized as follows. In Section 2, we give the terminology of order domains, and show that q-ary Reed-Muller codes are codes from order domains. The number of zeros of a multivariate polynomial is estimated by a generalization of the footprint bound in [6] using Gröbner basis theory. We then present a list-decoding algorithm for q-ary Reed-Muller codes, which is a straightforward generalization of the list-decoding algorithm of Reed-Solomon codes by Guruswami and Sudan [9]. In Section 3, we show that q-ary Reed-Muller codes are subfield subcodes of Reed-Solomon codes, and propose a second decoding algorithm for q-ary Reed-Muller codes, using Guruswami-Sudan's algorithm of Reed-Solomon codes. In Section 4, we prove that q-ary Reed-Muller codes can be described as one-point AG codes, then using list-decoding algorithm of AG codes by Guruswami and Sudan [9] we give a third list-decoding scheme for q-ary Reed-Muller codes. In Section 5, we compare the error-correction capabilities and the complexities of the proposed list-decoding algorithms with each other and with the algorithm in [27]. Finally in Section 6, we give conclusions.

2 Codes from Order Domains and Decoding

In this section, we give the definitions and properties on order domains necessary to state our results in sequel, and show that q-ary Reed-Muller codes are codes from order domains. Generalizing Guruswami-Sudan's list-decoding algorithm for Reed-Solomon codes [9], we present a list-decoding algorithm for q-ary Reed-Muller codes.

2.1 Order Domains and RM Codes

Let $\mathbb{F}$ be a field, a $\mathbb{F}$-algebra is a commutative ring (with a unit) that contains $\mathbb{F}$ as a unitary subring. Let $(\Gamma,+,0)$ be a commutative monoid. A partial order $<$ on Γ is called an admissible order if $0\leq\alpha$ for all $\alpha\in\Gamma$; and if $\alpha<\beta$, then $\alpha+\gamma<\beta+\gamma$ for all $\alpha,\beta,\gamma\in\Gamma$. If $(\Gamma,+,0)$ is a semigroup and $<$ is an admissible total order on Γ, then $(\Gamma,+,0,<)$ is called

a well-ordered semigroup, we sometimes call $(\Gamma, <)$ a well-order. If $<$ is an order on Γ, then $<$ is a order on $\Gamma \cup \{-\infty\}$ with $-\infty$ as the minimal element.

Let R be a $\mathbb{F}$-algebra, $(\Gamma, <)$ be a well-order. An *order function* is a surjective function

$$\rho : \ R \longrightarrow \Gamma \cup \{-\infty\},$$

satisfying the following properties:

(O.0) $\rho(f) = -\infty$ if and only if $f = 0$

(O.1) $\rho(af) = \rho(f)$ for all nonzero $a \in \mathbb{F}$

(O.2) $\rho(f + g) \leq \max\{\rho(f), \rho(g)\}$

(O.3) If $\rho(f) < \rho(g)$ and $h \neq 0$, then $\rho(fh) < \rho(gh)$

(O.4) If f and g are nonzero and $\rho(f) = \rho(g)$, then there exists a nonzero $a \in \mathbb{F}$ such that $\rho(f - ag) < \rho(g)$

for all $f, g, h \in R$.

Let R be a $\mathbb{F}$-algebra, $(\Gamma, <)$ a well-order and $\rho : \ R \longrightarrow \Gamma \cup \{-\infty\}$ an order function. Then (R, ρ, Γ) is called an *order structure* and R an *order domain* over $\mathbb{F}$. In this paper, we assume $\Gamma = \mathbb{N}_0^r = \{(n_1, \cdots, n_r) \mid n_i \text{ are nonnegative integers}\}$ for some r.

Let $<$ be an admissible well-order on $(\mathbb{N}_0^r, +)$, and R be a $\mathbb{F}$-algebra. A *weight function of rank* r on R is a order function from R to $\mathbb{N}_0^r \cup \{-\infty\}$ satisfying further

(O.5) $\rho(fg) = \rho(f) + \rho(g)$

for all $f, g \in R$. It can be proved that every order domain over $\mathbb{F}$ with a finitely generated value semigroup is finitely generated as an algebra over $\mathbb{F}$ and has a weight function.

Suppose R is an order domain over $\mathbb{F}$, then it is clear that there exists a basis $\{\varphi_\alpha \mid \rho(\varphi_\alpha) = \alpha \in \Gamma\}$ of R over $\mathbb{F}$. Let R_γ be the subspace of R generated by $\{\varphi_\alpha \mid \alpha \leq \gamma\}$, i.e., $R_\gamma = \{f \in R \mid \rho(f) \leq \gamma\}$. Then, $R = \bigcup_{\gamma \in \Gamma} R_\gamma$. For more properties, see [7].

It is clear that $R = \mathbb{F}_q[X_1, \cdots, X_m]$ is an order domain. Let $\Gamma = \mathbb{N}_0^m$, and let $<$ be the graded lexicographical order. Then the following mapping defines a weight function

$$\begin{array}{rccl} \rho: & \mathbb{F}_q[X_1, \cdots, X_m] & \longrightarrow & \mathbb{N}_0^m \cup \{-\infty\} \\ & X^\alpha & \longmapsto & \alpha. \end{array}$$

In $\Gamma = \mathbb{N}_0^m$, under the graded lexicographical order $<$

$$(0, \cdots, 0) < (1, 0, \cdots, 0) < (0, 1, 0, \cdots, 0) < \cdots < (0, \cdots, 0, 1)$$

$$< (2, 0, \cdots, 0) < (1, 1, 0, \cdots, 0) < \cdots < (0, \cdots, 0, 2) < \cdots .$$

In $\mathbb{F}_q[X_1, \cdots, X_m]$, we have correspondingly

$$1 < X_1 < X_2 < \cdots < X_m < X_1^2 < X_1 X_2 < \cdots < X_m^2 < \cdots .$$

For any $f \in \mathbb{F}_q[X_1, \cdots, X_m]$, let $\mathrm{ev}(f) = (f(P_1), \cdots, f(P_n))$. We then define an evaluation map

$$\mathrm{ev}: \ \mathbb{F}_q[X_1, \cdots, X_m] \longrightarrow \mathbb{F}_q^n.$$

Let $\alpha := (0, \cdots, 0, u) \in \mathbb{N}_0^m$. Then, we have

$$\mathcal{RM}_q(u, m) = \mathrm{ev}(R_\alpha).$$

2.2 Generalized Footprint Bound

In this subsection we will give some lemmas which will be used to prove the correctness of a list-decoding algorithm that we will give in next subsection. The main results in this subsection are in fact generalizations of the footprint bound [6] of the number of zeros of a polynomial.

Definition 2.1 *Let $P = (x_1, \ldots, x_m) \in \mathbb{F}_q^m$. Let $f \in \mathbb{F}_q[X_1, \ldots, X_m]$ be a polynomial of degree d. Define $f_P(X_1, \ldots, X_m) = f(X_1 + x_1, \ldots, X_m + x_m)$. Then $f_P(0) = f(P)$ and we can write*

$$f_P = f_{P,r} + \cdots + f_{P,d},$$

where $f_{P,j}$ is homogeneous of degree j in $X_1, \ldots, X_m$ and both $f_{P,r}$ and $f_{P,d}$ are nonzero. Then r is the multiplicity *of f at P, denoted by $r_P(f)$.*

Definition 2.2 *The* zeroset *or* vanishing set $Z(f)$ *of a polynomial* $f \in \mathbb{F}_q[X_1, \ldots, X_m]$ *is defined by*

$$Z(f) = \{\ P \in \bar{\mathbb{F}}_q^m \mid f(P) = 0\ \}.$$

If we want to restrict our attention to the zeros in $\mathbb{F}_q^m$ *we define*

$$Z(f, \mathbb{F}_q) = \{\ P \in \mathbb{F}_q^m \mid f(P) = 0\ \}.$$

The vanishing or zeroset $Z(I)$ *of an ideal* I *in* $\mathbb{F}_q[X_1, \ldots, X_m]$ *is defined by*

$$Z(I) = \{P \in \bar{\mathbb{F}}_q^m \mid f(P) = 0 \text{ for all } f \in I\ \}.$$

Definition 2.3 *Let* $I(q, m)$ *be the ideal in* $\mathbb{F}_q[X_1, \ldots, X_m]$ *generated by the elements* $X_i^q - X_i$ *for all* $i = 1, \ldots, m$. *Let* $I(q, r, m) = I(q, m)^r$.

Gröbner basis theory tells us that the dimension of $\mathbb{F}_q[X_1, \ldots, X_m]/I(q, r, m)$ over $\mathbb{F}_q$ is equal to the size of the footprint $\Delta(I(q, r, m))$. We take the Gröbner basis with respect to the total degree lexicographic order. The footprint of an ideal I in $\mathbb{F}_q[X_1, \ldots, X_m]$ is the set of all exponents $a \in \mathbb{N}_0^m$ such that X^a is not a leading term of an element of I. Clearly the monomial

$$X_1^{qe_1} \cdots X_m^{qe_m}$$

is the leading monomial of

$$(X_1^q - X_1)^{e_1} \cdots (X_m^q - X_m)^{e_m}$$

which is an element of $I(q, r, m)$ if $e_1 + \cdots + e_m = r$. Hence

$$\Delta(I(q, r, m)) \subseteq \Delta(X_1^{qe_1} \cdots X_m^{qe_m} \mid e_1 + \cdots + e_m = r)$$

Lemma 2.1 *The footprint* $\Delta(X_1^{qe_1} \cdots X_m^{qe_m} \mid e_1 + \cdots + e_m = r)$ *is the disjoint union of the sets*

$$\{\ a \in \mathbb{N}_0^m \mid qb_i \leq a_i < q(b_i + 1) \text{ for all } i\ \}$$

where $b \in \mathbb{N}_0^m$ *and* $b_1 + \cdots + b_m \leq r - 1$.

Proof: By definition $\Delta(X_1^{qe_1} \cdots X_m^{qe_m} \mid e_1 + \cdots + e_m = r)$ is equal to the set of all $a \in \mathbb{N}_0^m$ such that for all $e \in \mathbb{N}_0^m$ with $e_1 + \cdots + e_m = r$ there exists an i such that $a_i < qe_i$.

(1) Suppose that there exists an $b \in \mathbb{N}_0^m$ such that $b_1 + \cdots + b_m \leq r-1$ and $qb_i \leq a_i < q(b_i+1)$ for all i. Let $e \in \mathbb{N}_0^m$ such that $e_1 + \cdots + e_m = r$. Then

$$b_1 + \cdots + b_m \leq r - 1 < r = e_1 + \cdots + e_m.$$

Hence there exists an i such that $b_i + 1 \leq e_i$, so $a_i < q(b_i+1) \leq qe_i$.

(2) Conversely, suppose that for all $e \in \mathbb{N}_0^m$ with $e_1 + \cdots + e_m = r$ there exists an i such that $a_i < qe_i$. Define $b_i = \lfloor a_i/q \rfloor$, $i = 1, \cdots, n$. Then $qb_i \leq a_i < q(b_i+1)$ for all i. Now we have to show that $b_1 + \cdots + b_m \leq r-1$. Otherwise $b_1 + \cdots + b_m \geq r$. Hence there exists an $e \in \mathbb{N}_0^m$ such that $e_1 + \cdots + e_m = r$ and $e_i \leq b_i$ for all i. Then $qe_i \leq qb_i \leq a_i$ for all i, which is a contradiction. □

Lemma 2.2 *Let $u \in \mathbb{N}_0$. Then,*

$$|\{a = (a_1, \cdots, a_m) \in \mathbb{N}_0^m \mid a_1 + \cdots + a_m \leq u\}| = \begin{pmatrix} u+m \\ m \end{pmatrix}.$$

Let $R = \mathbb{F}_q[X_1, \ldots, X_m]$. Let $\mathcal{M}_i = (X_1 - x_{i1}, \ldots, X_m - x_{im})$ be the maximal ideal of the point $P_i = (x_{i1}, \ldots, x_{im})$, where $i = 1, \ldots, N = q^m$. The ideals $\cap_{i=1}^N \mathcal{M}_i$ and $I(q,m)$ are radical and have the same zeroset $\{P_1, \ldots, P_N\}$. Hence the ideals are the same

$$I(q,m) = \cap_{i=1}^N \mathcal{M}_i.$$

Now for arbitrary r we have that

$$I(q,r,m) = I(q,m)^r = (\cap_{i=1}^N \mathcal{M}_i)^r \subseteq (\cap_{i=1}^N \mathcal{M}_i^r).$$

This induces an R-linear map

$$\varphi_{q,r} : \mathbb{F}_q[X_1, \ldots, X_m]/I(q,r,m) \longrightarrow \oplus_{i=1}^N \mathbb{F}_q[X_1, \ldots, X_m]/\mathcal{M}_i^r.$$

Theorem 2.3 *The map $\varphi_{q,r}$ is an isomorphism and for the number of elements of the footprint we have that*

$$|\Delta(I(q,r,m))| = \binom{m+r-1}{r-1} q^m.$$

Proof: (1) If $i \neq j$ then the maximal ideals $\mathcal{M}_i$ and $\mathcal{M}_j$ are distinct, hence $\mathcal{M}_i + \mathcal{M}_j = R$ is the whole ring. But also $\mathcal{M}_i^r + \mathcal{M}_j^r = R$, since for the zerosets we have that

$$Z(\mathcal{M}_i^r + \mathcal{M}_j^r) = Z(\mathcal{M}_i^r) \cap Z(\mathcal{M}_j^r) = \{P_i\} \cap \{P_j\} = \emptyset.$$

The Chinese Remainder Theorem [15, II §2, page 64] gives an isomorphism

$$\mathbb{F}_q[X_1, \ldots, X_m]/(\cap_{i=1}^N \mathcal{M}_i^r) \cong \oplus_{i=1}^N \mathbb{F}_q[X_1, \ldots, X_m]/\mathcal{M}_i^r.$$

The map $\varphi_{q,r}$ is clearly surjective. Therefore the dimension of $\mathbb{F}_q[X_1, \ldots, X_m]/I(q, r, m)$ is at least the dimension of $\oplus_{i=1}^N \mathbb{F}_q[X_1, \ldots, X_m]/\mathcal{M}_i^r$ over $\mathbb{F}_q$. The last mentioned space has dimension $\binom{m+r-1}{r-1} q^m$, since for every i we have that the polynomials

$$(X_1 - x_{i1})^{a_1} \cdots (X_m - x_{im})^{a_m}$$

with $a_1 + \cdots + a_m \leq r - 1$ represent a basis for the vector space $\mathbb{F}_q[X_1, \ldots, X_m]/\mathcal{M}_i^r$ over $\mathbb{F}_q$.

(2) The dimension of $\mathbb{F}_q[X_1, \ldots, X_m]/I(q, r, m)$ over $\mathbb{F}_q$ is equal to the size of the footprint $\Delta(I(q, r, m))$ and

$$\Delta(I(q, r, m)) \subseteq \Delta(X_1^{qe_1} \cdots X_m^{qe_m} \mid e_1 + \cdots + e_m = r)$$

which is the disjoint union of the sets

$$\{\ a \in \mathbb{N}_0^m \mid qb_i \leq a_i < q(b_i + 1) \text{ for all } i\ \}$$

where $b \in \mathbb{N}_0^m$ and $b_1 + \cdots + b_m \leq r - 1$, by Lemma 2.1. Therefore

$$|\Delta(I(q, r, m))| \leq \binom{m + r - 1}{r - 1} q^m.$$

(3) Combining (1) and (2) gives the desired result. □

Definition 2.4 *Let $f \in R$. The ideals $I(q, m, f)$ and $I(q, r, m, f)$ in R are defined by*

$$I(q, m, f) = \langle f \rangle + I(q, m) \quad \textit{and} \quad I(q, r, m, f) = \langle f \rangle + I(q, r, m).$$

Lemma 2.4 *Let $f \in R$. If t is the number of points in $\mathbb{F}_q^m$ where f has at least multiplicity r, then*

$$\dim_{\mathbb{F}_q} R/I(q, r, m, f) \geq \binom{m + r - 1}{r - 1} t.$$

Proof: Theorem 2.3 gives that for every $f \in R$ also the induced map

$$\varphi_{q,r}(f) : R/I(q,r,m,f) \longrightarrow \oplus_{i=1}^{N} R/(\mathcal{M}_i^r + \langle f \rangle).$$

is an isomorphism. If the multiplicity of f at P_i is at least r, then $f \in \mathcal{M}_i^r$ and $R/(\mathcal{M}_i^r + \langle f \rangle)$ has dimension $\binom{m+r-1}{r-1}$. $\square$

Lemma 2.5 *Let $f \in R$. Let $w = \lfloor v/q \rfloor$. If $\deg(f) \leq v < qr$, then an upper bound for the dimension of $R/I(q,r,m,f)$ over $\mathbb{F}_q$ is given by*

$$\binom{m+r-1}{m} q^m + (v - qw)\binom{m+r-w-2}{m-1} q^{m-1} - \binom{m+r-w-1}{m} q^m.$$

Proof: 1) Define the set $\Delta(q,r,m) := \Delta(I(q,r,m))$. In Theorem 2.3 it was shown that

$$\Delta(q,r,m) = \{\ a \in \mathbb{N}_0^m \mid \textstyle\sum_{i=1}^{m} \lfloor a_i/q \rfloor \leq r-1\ \}.$$

Define for $a \in \mathbb{N}_0^m$ the sets $\bigtriangledown(q,r,m,a)$ and $\Delta(q,r,m,a)$ by

$$\bigtriangledown(q,r,m,a) = \{b \in \mathbb{N}_0^m \mid b \in \Delta(q,r,m), a_i \leq b_i \text{ for all } i\ \}.$$

$$\Delta(q,r,m,a) = \{b \in \mathbb{N}_0^m \mid b \in \Delta(q,r,m), b_i < a_i \text{ for some } i\ \} = \Delta(q,r,m) \setminus \bigtriangledown(q,r,m,a).$$

Let $\Delta(q,r,m,f)$ be the delta set of the ideal $I(q,r,m,f)$. If X^a is the leading monomial of f, then

$$\Delta(q,r,m,f) \subseteq \Delta(q,r,m,a).$$

The dimension of $R/I(q,r,m,f)$ is equal to the size of $\Delta(q,r,m,f)$, so we are looking for an upper bound for this size. For fixed q, r and m we denote $\bigtriangledown(q,r,m,a)$ and $\Delta(q,r,m,a)$ by $\bigtriangledown(a)$ and $\Delta(a)$, respectively.

2) We claim that the maximal size of $\Delta(a)$, or equivalently the minimal size of $\bigtriangledown(a)$, for all $a \in \Delta(q,r,m)$ such that $\deg(a) := a_1 + \cdots + a_m \leq v$, is attained for $a = (v, 0, \ldots, 0)$. Notice that it is assumed that $v < rq$, so indeed $(v, 0, \ldots, 0)$ is an element of $\Delta(q,r,m)$. This claim will be shown by induction on r. In case $r = 1$ this is a well-known fact in extremal poset theory [3, 13, 14, 16] and used to find the generalized Hamming weights of Reed-Muller codes in [10, 28]. Now assume that the claim is shown for all $r^* < r$. If $a \in \Delta(q,r,m)$ and $a_i \geq q$ for some i, then without loss of generality we may assume that $i = m$. Consider the maps

$$b = (b_1, \ldots, b_m) \mapsto \tilde{b} := (b_1, \ldots, b_{m-2}, b_{m-1} + q, b_m - q) \text{ and}$$

$$b = (b_1, \ldots, b_m) \mapsto b' := (b_1, \ldots, b_{m-2}, b_{m-1}, b_m - q).$$

Then $\deg(\tilde{a}) \leq v < rq$ and $\tilde{a} \in \Delta(q, r, m)$, and $\deg(a') \leq v - q < (r-1)q$ and $a' \in \Delta(q, r-1, m)$. This gives bijections

$$\bigtriangledown(q, r, m, a) \to \bigtriangledown(q, r, m, \tilde{a})) \text{ and } \bigtriangledown(q, r, m, a) \to \bigtriangledown(q, r-1, m, a').$$

The induction hypothesis gives that the minimum is attained for $a = (v, 0, \ldots, 0)$. Therefore we may assume that $a_i < q$ for all i. Now let $r^* = 1$ and $q^* = rq$ (for this proof it is not assumed that q is a prime power). Then

$$\bigtriangledown(q^*, r^*, m, a) = \bigtriangledown(q, r, m, a) \cup \bigtriangledown(q^*, r^*, m, (q, q, \ldots, q))$$

is a disjoint union. Hence

$$|\bigtriangledown(q^*, r^*, m, a)| = |\bigtriangledown(q, r, m, a)| + |\bigtriangledown(q^*, r^*, m, (q, q, \ldots, q))|$$

Since $r^* = 1$, we know that the minimum at the left hand side is attained for $a = (v, 0, \ldots, 0)$. Therefore this is also true for $\bigtriangledown(q, r, m, a)$. In this way we have shown the claim by induction.

3) Next the maximal size of $\Delta(q, r, m, a)$ is computed. Assume $a = (v, 0, \ldots, 0)$. Let $w = \lfloor v/q \rfloor$. Then

$$\Delta(q, r, m, a) = \{\ b \in \Delta(q, r, m) \mid b_1 < v\ \}$$

which is equal to the following union of disjoint sets

$$\bigcup_{s=0}^{w-1} \{\ b \in \Delta(q, r, m) \mid \textstyle\sum_{i=2}^{m} \lfloor b_i/q \rfloor \leq r - s - 1 \text{ and } s = \lfloor b_1/q \rfloor\ \} \bigcup$$

$$\{\ b \in \Delta(q, r, m) \mid \textstyle\sum_{i=2}^{m} \lfloor b_i/q \rfloor \leq r - w - 1 \text{ and } wq \leq b_1 < v\ \}$$

Now the map defined by $b = (b_1, b_2, \ldots, b_m) \mapsto (b_2, \ldots, b_{m-1})$ gives a q to 1 map between

$$\{\ b \in \Delta(q, r, m) \mid \textstyle\sum_{i=2}^{m} \lfloor b_i/q \rfloor \leq r - s - 1 \text{ and } s = \lfloor b_1/q \rfloor\ \}$$

and $\Delta(q, r-s, m-1)$, and a $v - wq$ to 1 map between

$$\{\ b \in \Delta(q, r, m) \mid \textstyle\sum_{i=2}^{m} \lfloor b_i/q \rfloor \leq r - w - 1 \text{ and } wq \leq b_1 < v\ \}$$

and $\Delta(q, r-w, m-1)$. Lemma 2.1 implies

$$|\Delta(q, r-s, m-1)| = \binom{m-2+r-s}{m-1} q^{m-1}.$$

Therefore

$$|\Delta(q,r,m,a)| = \sum_{s=0}^{w-1} \binom{m-2+r-s}{m-1} q^m + (v-qw)\binom{m-2+r-w}{m-1} q^{m-1}.$$

4) A similar argument gives the well-known identity

$$\binom{m+r-1}{m} = \sum_{s=0}^{r-1} \binom{m-2+r-s}{m-1}.$$

which is equal to

$$\sum_{s=0}^{w-1} \binom{m-2+r-s}{m-1} + \sum_{s=w}^{r-1} \binom{m-2+r-s}{m-1} =$$

$$\sum_{s=0}^{w-1} \binom{m-2+r-s}{m-1} + \sum_{i=0}^{r-w-1} \binom{m-2+r-w-i}{m-1} =$$

$$\sum_{s=0}^{w-1} \binom{m-2+r-s}{m-1} + \binom{m+r-w-1}{m}.$$

Therefore

$$\sum_{s=0}^{w-1} \binom{m-2+r-s}{m-1} = \binom{m+r-1}{m} - \binom{m+r-w-1}{m}.$$

Substituting this in the end result of (3) gives the desired upper bound. □

2.3 Decoding Algorithm

We assume that $R = \mathbb{F}_q[X_1, \cdots, X_m]$, $n = q^m$ is the length of the code $\mathcal{RM}_q(u,m) = \mathrm{ev}(R_\alpha)$, where $\alpha = (0, \cdots, 0, u)$. Let $(y_1, \cdots, y_n) \in \mathbb{F}_q^n$ be a received word. In the following we give a list-decoding algorithm which finds all the E-consistent codewords in $\mathcal{RM}_q(u,m)$ for an appropriate parameter E.

Algorithm 2.1 (*List Decoding of RM Codes over Order Domains*)

Input: n, t, $\alpha = (0, \cdots, 0, u) \in \mathbb{N}^m$, $(y_1, \cdots, y_n) \in \mathbb{F}_q^n$, and $P_i := (x_{i1}, \cdots, x_{im}) \in \mathbb{F}_q^m$, $i = 1, \cdots, n$.

Step 0: Choose the parameters r, v such that:

- (A.1) $s := \lfloor \frac{v}{u} \rfloor$ and

$$\sum_{j=0}^{s} \begin{pmatrix} v - ju + m \\ m \end{pmatrix} > n \begin{pmatrix} m + r \\ m + 1 \end{pmatrix};$$

- (A.2) $E := n - t$, $w := \lfloor \frac{v}{q} \rfloor$ and

$$\binom{m+r-1}{m} E < \binom{m+r-w-1}{m} q^m - (v - qw)\binom{m+r-w-2}{m-1} q^{m-1}.$$

Step 1: Find a nonzero polynomial $H(T) \in R[T]$ of the form

$$H(T) = H(X;T) = \sum_{j=0}^{s} h_j(X) T^j,$$

where

$$h_j(X) = \sum_{(j_1,\cdots,j_m) \le \gamma - j\alpha} h_{j_1,\cdots,j_m;j} X_1^{j_1} \cdots X_m^{j_m} \in R_{\gamma - j\alpha}, \quad j = 0, 1, \cdots, s,$$

with $\gamma := (0, \cdots, 0, v)$, such that for all $i = 1, \cdots, n$, and for any $j_1, \cdots, j_m, j \ge 0$ and $j_1 + \cdots + j_m + j \le r - 1$,

$$\begin{aligned} h^{(i)}_{j_1,\cdots,j_m;j} &:= \sum_{j'_1 \ge j_1} \cdots \sum_{j'_m \ge j_m} \sum_{j' \ge j} \begin{pmatrix} j'_1 \\ j_1 \end{pmatrix} \cdots \begin{pmatrix} j'_m \\ j_m \end{pmatrix} \begin{pmatrix} j' \\ j \end{pmatrix} h_{j'_1,\cdots,j'_m;j} x_{i1}^{j'_1 - j_1} \cdots x_{im}^{j'_m - j_m} y_i^{j' - j} \\ &= 0. \end{aligned}$$

Step 2: Using the root-finding algorithm in [30], find all the roots $f \in R_\alpha$ of the polynomial $H(T)$. For each such f, check if $f(P_i) = y_i$ for at least t values of $i \in \{1, \cdots, n\}$, and if so, include f in output list.

□

2.4 Analysis of Algorithm 2.1

The task of finding a nonzero $H(T)$ in Step 1 is that of solving a system of homogeneous linear equations, where the undetermined coefficients $h_{j_1,\cdots,j_m;j}$ of $H(T)$ are the unknowns. To prove the existence of $H(T)$, it is sufficient to prove that the number of unknowns is strictly greater than the number of equations.

The following lemma gives the number of the undetermined coefficients of $H(T)$.

Lemma 2.6 *Assume* $\alpha = (0, \cdots, 0, u)$ *and* $\gamma = (0, \cdots, 0, v)$ *with* $s = \lfloor v/u \rfloor$. *Then*

$$|\{h_{j_1,\cdots,j_m;j} \mid (j_1, \cdots, j_m) \leq \gamma - j\alpha,\ j = 0, 1, \cdots, s\}| = \sum_{j=0}^{s} \begin{pmatrix} v - ju + m \\ m \end{pmatrix}.$$

Lemma 2.7 *Let* u *be a positive integer such that* $u < q$. *Define* $\mu := \sqrt[m+1]{u/q}$. *Choose a positive integer* r *such that*

$$r \geq \frac{qm\mu + 1}{q(1-\mu)}.$$

Define $v = \lceil q(r+m)\mu \rceil$ *and* $w = \lfloor v/q \rfloor$. *Let* E *a positive integer such that*

$$E < \frac{[q(r-w)]^m - (v - qw)m[q(m+r-w-2)]^{m-1}}{(m+r-1)^m}.$$

Let $t = n - E$. *Then conditions* $(A.1)$ *and* $(A.2)$ *of Algorithm 2.1 hold.*

Proof:

(1) The condition on r implies that $q(r+m)\mu + 1 \leq rq$. So

$$v = \lceil q(r+m)\mu \rceil < q(r+m)\mu + 1 \leq rq.$$

(2) It is easy to verify the following inequalities

$$\frac{(r+1)^m}{m!} \leq \binom{m+r}{m} \leq \frac{(m+r)^m}{m!}.$$

(3) Using the above inequalities we have

$$\sum_{j=0}^{s} \binom{v - ju + m}{m} \geq \sum_{i=0}^{s} \begin{pmatrix} m + iu \\ m \end{pmatrix} > \frac{u^m}{m!} \sum_{i=0}^{s} i^m \geq \frac{u^m s^{m+1}}{(m+1)!} = \frac{v^{m+1}}{u(m+1)!}.$$

(4) The definition of v gives $v^{m+1} \geq uq^m(m+r)^{m+1}$. Now $n = q^m$. Hence

$$\sum_{j=0}^{s} \binom{v - ju + m}{m} > \frac{v^{m+1}}{u(m+1)!} \geq \frac{n(m+r)^{m+1}}{(m+1)!} \geq n\binom{m+r}{m+1},$$

by (2). Therefore we have the desired inequality $(A.1)$ as a consequence of (3).

(5) The assumption on E implies

$$\frac{(m+r-1)^m}{m!}E < \frac{(r-w)^m}{m!}q^m - (v-qw)\frac{(m+r-w-2)^{m-1}}{(m-1)!}q^{m-1}.$$

Applying three times the inequalities (2) we get condition $(A.2)$:

$$\binom{m+r-1}{m}E < \binom{m+r-w-1}{m}q^m - (v-qw)\binom{m+r-w-2}{m-1}q^{m-1}.$$

□

Lemma 2.8 *Assume condition* $(A.1)$ *of Algorithm 2.1 holds. Then a nonzero polynomial* $H(T)$ *as sought in Step 1 of Algorithm 2.1 does exist.*

Proof: By Lemma 2.6, the number of the undetermined coefficients of $H(T)$ is equal to $\sum_{j=0}^{\lfloor \frac{v}{u} \rfloor} \binom{v-ju+m}{m}$. On the other hand, the number of constraints of $H(T)$ is equal to $n \cdot \left|\left\{(j_1, \cdots, j_m, j) \in \mathbb{N}_0^{m+1} \mid j_1 + \cdots + j_m + j \leq r-1\right\}\right|$, which is equal to $n \cdot \binom{m+r}{m+1}$. So by condition $(A.1)$, the number of unknowns is greater than the number of equations, we can get such a $H(T)$ by solving a homogeneous linear system. □

Let $f = f(X_1, \cdots, X_m) = \sum_{j_1, j_2, \cdots, j_m} a_{j_1 j_2 \cdots j_m} X_1^{j_1} \cdots X_m^{j_m} \in R = \mathbb{F}_q[X_1, \cdots, X_m]$. From Definition 2.1, $P = (x_1, \cdots, x_m) \in \mathbb{F}_q^m$ is a zero of f of multiplicity r if and only if $f_P(X_1, \cdots, X_m) = f(X_1 + x_1, \cdots, X_m + x_m) = \sum_{j_1, j_2, \cdots, j_m} b_{j_1 j_2 \cdots j_m} X_1^{j_1} \cdots X_m^{j_m}$ with that $b_{j_1 j_2 \cdots j_m} = 0$ for all $j_1, \cdots, j_m \geq 0, j_1 + \cdots + j_m \leq r-1$ and $b_{j_1 j_2 \cdots j_m} \neq 0$ for some $j_1 + \cdots + j_m = r$. The constraints for $H(X;T)$ in the Step 1 of Algorithm 2.1 mean that every $(P_i; y_i)$ is a zero of $H(X;T)$ of multiplicity $\geq r$. Let $H(X;T)$ be such a polynomial, then we can prove

Lemma 2.9 *Let* $f \in R$. *For any* $i \in \{1, \cdots, n\}$, *if* $f(P_i) = y_i$, *then* P_i *is a zero of* $H(f) = H(X; f(X))$ *of multiplicity* $\geq r$.

Proof: Let $G(X) := H(X; f(X))$. Denote

$$f^{(i)}(X) = f(X + P_i) - y_i = f(X_1 + x_{i1}, \cdots, X_m + x_{im}) - y_i.$$

Since $f(P_i) = y_i$, we have $f^{(i)}(0) = 0$. So there exist $f_k^{(i)}(X)$ for all $k = 1, \ldots, m$ such that

$$f^{(i)}(X) = \sum_{k=1}^{m} X_k f_k^{(i)}(X).$$

Hence

$$\begin{aligned} G(X) &= H(X; f(X)) \\ &= H_{(P_i,y_i)}(X - P_i; f(X) - y_i) \\ &= H_{(P_i,y_i)}(X - P_i; f^{(i)}(X - P_i)) \\ &= H_{(P_i,y_i)}(X_1 - x_{i1}, \cdots, X_m - x_{im}; \textstyle\sum_{k=1}^{m}(X_k - x_{ik}) f_k^{(i)}(X - P_i)) \end{aligned}$$

where $H_{(P_i,y_i)}(X; T) = H(X + P_i; T + y_i) = H(X_1 + x_{i1}, \cdots, X_m + x_{im}; T + y_i)$.

From Step 1 of Algorithm 2.1 we have

$$H_{(P_i,y_i)}(X; T) = \sum_{\mathbf{j}} \sum_{l} h_{\mathbf{j}l}^{(i)} X_1^{j_1} \cdots X_m^{j_m} T^l,$$

and $h_{\mathbf{j}l}^{(i)} = 0$ for $j_1 + \cdots + j_m + l \leq r - 1$. In other words, the first nonzero terms of $H_{(P_i,y_i)}(X; T)$ are

$$h_{\mathbf{j}l}^{(j)} X_1^{j_1} \cdots X_m^{j_m} T^l, \qquad \text{with } \; j_1 + \cdots + j_m + l \geq r.$$

Now we have $G(X)$ is equal to

$$H_{(P_i,y_i)}(X_1 - x_{i1}, \cdots, X_m - x_{im}; \sum_{k=1}^{m}(X_k - x_{ik}) f_k^{(i)}(X - P_i))$$

$$\sum_{\mathbf{j}} \sum_{l} h_{\mathbf{j}l}^{(i)} (X_1 - x_{i1})^{j_1} \cdots (X_m - x_{im})^{j_m} \left(\sum_{k=1}^{m} (X_k - x_{ik}) f_k^{(i)}(X - P_i) \right)^l.$$

So, $G(X + P_i)$ is equal to

$$\sum_{\mathbf{j}} \sum_{l} h_{\mathbf{j}l}^{(i)} X_1^{j_1} \cdots X_m^{j_m} \left(\sum_{\mathbf{l}, l_1 + \cdots l_m = l} \binom{l}{l_1, \ldots, l_m} X_1^{l_1} \cdots X_m^{l_m} (f_1^{(i)}(X))^{l_1} \cdots (f_m^{(i)}(X))^{l_m} \right) =$$

$$\sum_{\mathbf{j}} \sum_{l} \sum_{\mathbf{l}, l_1 + \cdots l_m = l} \binom{l}{l_1, \ldots, l_m} h_{\mathbf{j}l}^{(i)} X_1^{j_1 + l_1} \cdots X_m^{j_m + l_m} (f_1^{(i)}(X))^{l_1} \cdots (f_m^{(i)}(X))^{l_m}.$$

The first nonzero terms of $G(X + P_i)$ contain the monomials

$$X_1^{j_1 + l_1} \cdots X_m^{j_m + l_m},$$

where

$$(j_1 + l_1) + \cdots + (j_m + l_m) = j_1 + \cdots + j_m + l \geq r.$$

Thus, P_i is a zero of $G(X)$ of multiplicity $\geq r$. □

Lemma 2.10 *If $f \in R_\alpha$ is such that $f(P_i) = y_i$ for at least t values of $i \in \{1, \cdots, n\}$ and condition (A.2) holds, then $H(f) = 0$, i.e., $H(f)$ is identically zero as a polynomial in $R = \mathbb{F}_q[X_1, \cdots, X_m]$.*

Proof: Since $f \in R_\alpha$, we have $\rho(H(f)) \leq \gamma = (0, \cdots, 0, v)$. If $H(f)$ is not identically zero, then the dimension of $R/I(q, r, m, H(f))$ is at most

$$\binom{m+r-1}{m} q^m + (v - qw)\binom{m+r-w-2}{m-1} q^{m-1} - \binom{m+r-w-1}{m} q^m$$

by Lemma 2.5. On the other hand, $f(P_i) = y_i$ for at least t values of $i \in \{1, \cdots, n\}$, and from Lemma 2.9, for each such i, P_i is a zero of $H(f)$ of multiplicity $\geq r$. Thus, from Lemma 2.4,

$$\dim_{\mathbb{F}_q} R/I(q, r, m, (H(f)) \geq \begin{pmatrix} m+r-1 \\ r-1 \end{pmatrix} t.$$

From the two inequalities we derive a contradiction with condition (A.2). □

Theorem 2.11 *Let q be a power of a prime, u be a positive integer such that $0 < u < q$. Let $\mathcal{RM}_q(u, m)$ be the Reed-Muller code of length $n = q^m$. Let $\mu = \sqrt[m+1]{u/q}$. Let Δ be any real number such that $0 < \Delta \leq \min\left\{\frac{mn(1-\mu)}{\mu}, n\right\}$. Let r and M be integers defined by*

$$r = \left\lceil \frac{2m(m+\mu)n}{\Delta} \right\rceil \quad \text{and} \quad M = \left\lceil \frac{r+m}{\mu^m} \right\rceil.$$

Let E be an integer such that

$$E \leq n(1-\mu)^m - \Delta.$$

Then Algorithm 2.1 correctly finds all the E-consistent codewords in $\mathcal{RM}_q(u, m)$, and the number of the E-consistent codewords is at most M.

Proof: From the upper bound on Δ and the definition of r, we have

$$r \geq \frac{2m(m+\mu)n}{\Delta} > \frac{2m^2 n}{\Delta} \geq \frac{2m\mu}{1-\mu} \geq \frac{qm\mu + 1}{q(1-\mu)},$$

as is assumed in Lemma 2.7. Let v and w be defined as in Lemma 2.7.

Suppose that the following two inequalities hold

$$\left(\frac{q(r-w)}{m+r-1}\right)^m > n(1-\mu)^m - \Delta/2, \tag{2.1}$$

$$\frac{(v-qw)mq^{m-1}(m+r-w-2)^{m-1}}{(m+r-1)^m} < \Delta/2. \tag{2.2}$$

Then we have

$$\begin{aligned} E &\le n(1-\mu)^m - \Delta \\ &< \left[\left(\frac{q(r-w)}{m+r-1}\right)^m + \Delta/2\right] - \left[\frac{(v-qw)m[q(m+r-w-2)]^{m-1}}{(m+r-1)^m} - \Delta/2\right] - \Delta \\ &= \frac{[q(r-w)]^m - (v-qw)m[q(m+r-w-2)]^{m-1}}{(m+r-1)^m}. \end{aligned}$$

Therefore by Lemma 2.7, conditions (A.1) and (A.2) of Algorithm 2.1 hold. Then by Lemmas 2.8 and 2.10, Algorithm 2.1 correctly finds all the E-consistent codewords.

Now we are ready to prove that both (2.1) and (2.2) hold. Denote

$$f(r) := \frac{r-w}{m+r-1} = \frac{1-\frac{w}{r}}{1+\frac{m}{r}-\frac{1}{r}}, \qquad \text{and} \quad A := 1-\mu.$$

Let η be a small real number defined by $\eta = (m+\mu)/r$. Let w be defined as in Lemma 2.7. Then

$$1 - \frac{w}{r} \ge 1 - \frac{v}{rq} > 1 - \frac{q(r+m)\mu + 1}{rq} = A - \frac{qm\mu+1}{rq}.$$

Therefore

$$\begin{aligned} f(r) - A &= \tfrac{1-\frac{w}{r}}{1+\frac{m}{r}-\frac{1}{r}} - A \\ &> \tfrac{A-\frac{qm\mu+1}{rq}}{1+\frac{m}{r}-\frac{1}{r}} - A \\ &= -\tfrac{A(m-1)}{r+m-1} - \tfrac{qm\mu+1}{q(m+r-1)} \\ &> -\tfrac{A(m-1)}{r} - \tfrac{m\mu+1}{r} \\ &= -\tfrac{m+\mu}{r} = -\eta. \end{aligned}$$

Now by applying the Lagrange Intermediate-Value Theorem to the function $(A-x)^m$ we have

$$(A-\eta)^m - A^m = -m\eta(A-\xi)^{m-1},$$

where $0 < \xi < \eta$. We have $A - \eta < A - \xi < A < 1$. Since $r = \left\lceil \frac{2m(m+\mu)n}{\Delta} \right\rceil$ and $\Delta \leq n \leq 2mn(2-\mu) = 2mn(1+A)$, we have $r \geq \frac{m+\mu}{1+A}$. Thus, $A - \eta \geq -1$. So $-1 < A - \xi < 1$. Therefore we have $-m\eta(A-\xi)^{m-1} \geq -m\eta$, and

$$(f(r))^m - A^m > (A-\eta)^m - A^m \geq -m\eta = -\frac{m(m+\mu)}{r} \geq -\frac{\Delta}{2n}.$$

This implies (2.1).

On the other hand

$$\frac{(v-qw)mq^{m-1}(m+r-w-2)^{m-1}}{(m+r-1)^m} < \frac{mq^m}{r} \leq \Delta/2$$

by the definition of r. So (2.2) holds.

By Lemma 2.10, the number of E-consistent codewords is at most the degree $\deg_T(H(T)) = \lfloor \frac{v}{u} \rfloor$. And

$$\lfloor \frac{v}{u} \rfloor < \frac{q(r+m)\mu + 1}{u} = \frac{r+m}{\mu^m} + \frac{1}{u} \leq M + \frac{1}{u}.$$

Since $0 < \frac{1}{u} \leq 1$. So $\lfloor \frac{v}{u} \rfloor \leq M$. □

Remark 2.1: Let $\mu = \sqrt[m+1]{u/q}$. Let E be a positive integer such that

$$E \leq n\left(1 - \sqrt[m+1]{u/q}\right)^m - 1 \tag{2.3}$$

and

$$M = O\left(m^2 u^{-\frac{m}{m+1}} n^{1+\frac{1}{m+1}}\right).$$

Then by the theorem above, the q-ary Reed-Muller code $\mathcal{RM}_q(u,m)$ is (E,M)-decodable.

In fact, by the theorem above Algorithm 2.1 is a list-decoding algorithm for $\mathcal{RM}_q(u,m)$, which works for up to $n\left(1 - \sqrt[m+1]{u/q}\right)^m - \Delta$ errors, where $0 < \Delta \leq \min\{\frac{mn(1-\mu)}{\mu}, n\}$. Now let $\Delta = \min\{\frac{mn(1-\mu)}{\mu}, 1\}$. Then for any E satisfying

$$E \leq n\left(1 - \sqrt[m+1]{u/q}\right)^m - 1$$

we have $E \leq n\left(1 - \sqrt[m+1]{u/q}\right)^m - 1 \leq n\left(1 - \sqrt[m+1]{u/q}\right)^m - \Delta$. So the algorithm can correctly find all the E-consistent codewords for E satisfying the inequality above. Again by Theorem

2.11, the number of E-consistent codewords is at most M and

$$\begin{aligned} M &= O\left(\frac{r}{\mu^m}\right) = O\left(\frac{m^2 n}{\mu^m \Delta}\right) \\ &= O\left(\max\left\{\frac{m}{\mu^{m-1}-\mu^m}, \frac{m^2 n}{\mu^m}\right\}\right) \\ &= O\left(\frac{m}{\mu^{m-1}(1-\mu)} + \frac{m^2 n}{\mu^m}\right) \\ &= O\left(\frac{m}{\mu^{m-1}}\left(\frac{n}{E}\right)^{\frac{1}{m}} + \frac{m^2 n}{\mu^m}\right) \\ &= O\left(\frac{m n^{\frac{1}{m}}}{\mu^{m-1}} + \frac{m^2 n}{\mu^m}\right) \\ &= O\left(m^2 u^{-\frac{m}{m+1}} n^{1+\frac{1}{m+1}}\right). \end{aligned}$$

This implies that the number of E-consistent codewords is bounded by a polynomial in n for all m. □

Remark 2.2: The inequality (2.3) gives an error-correction bound of Algorithm 2.1. We see that Algorithm 2.1 works only for $u < q$, that is, this algorithm works only for Reed-Muller codes with low rates.

Let us see an example of Reed-Muller code of low rate. Let $q = 2^9 = 512$ and $m = 2$. Consider the Reed-Muller code $C = \mathcal{RM}_q(u, m)$ with $u = 2^3$. The parameters of C are

$$n = q^m = 2^{18}, \quad \text{and} \quad d = 504 \times 512 = 258048.$$

So the traditional error-correction bound is $\lfloor \frac{d-1}{2} \rfloor = 129023$.

Algorithm 2.1 works for up to $E = n(1 - \sqrt[m+1]{u/q})^m - 1 = 147455$ errors, which is much greater than the traditional error-correction bound. □

Now let us evaluate the time complexity of Algorithm 2.1. The task of Step 1 is solving a system of homogeneous linear equations to determine a polynomial $H(T)$, i.e., the coefficients $h_{j_1,\cdots,j_m;j}$ of $H(T)$. Let N denote the number of coefficients of $H(T)$. Using Gaussian elimination, Step 1 can be implemented to run using $O(N^3)$ field operations over $\mathbb{F}_q$.

Next, consider the number of field operations over $\mathbb{F}_q$ required to find all the roots $f \in R_\alpha$ of $H(T)$. We use the root-finding algorithm in [30]. From [30], for the polynomial $H(T) = h_0 + h_1 T + \cdots + h_s T^s$ returned from Step 1 of Algorithm 2.1, $h_0, h_1, \cdots, h_s$ can be written as polynomials in $\varphi_1, \varphi_2, \cdots$ with coefficients in $\mathbb{F}_q$. Let $\{\varphi_1, \varphi_2, \cdots, \varphi_k\}$ be a basis of R_α. The main tasks of Step 2 of Algorithm 2.1 is finding $f_1, f_2, \cdots, f_k \in \mathbb{F}_q$ such that $H(f_1\varphi_1 + \cdots + f_k\varphi_k) = 0$. Steps 1 and 3 of the root-finding algorithm in [30] calculate

the leading coefficients of $H_1(f_k\varphi_k)$ and $H_{i+1}(f_{k-i}\varphi_{k-i})$, respectively, where H_1 and H_{i+1} are polynomials generated from $H(T)$ and are of T-degree less than or equal to s. It is clear that both of them require $O(kN)$ operations over $\mathbb{F}_q$. Step 4 calculates the roots in $\mathbb{F}_q$ of a polynomial over $\mathbb{F}_q$ of degree at most s. From [2], the roots in $\mathbb{F}_q$ of a polynomial over $\mathbb{F}_q$ of degree s can be found in expected time complexity $O((s\log^2 s)\cdot(\log\log s)\cdot\log q)$. From the discussion above, it is easy to see that the roots in $f \in R_\alpha$ of $H(T)$ can be found using $O(k(N + s\log^2 s\cdot\log(\log s)\cdot\log q))$ field operations over $\mathbb{F}_q$. Note that $s < N$. So the time complexity of Step 2 of Algorithm 2.1 is bounded from above by $O(kN^2)$, where $kN^2 \le N^3$, since $k \le N$. So, the time complexity of Algorithm 2.1 is $O(N^3)$.

From Lemma 2.6, we have $N = \sum_{j=0}^{s} \binom{v-ju+m}{m}$, where $s = \lfloor \frac{v}{u} \rfloor \le M$. Since the dimension of $\mathcal{RM}_q(u,m)$ is $k = \binom{u+m}{m}$, we have

$$\begin{aligned} N &= \sum_{j=0}^{s} \tbinom{v-ju+m}{m} < \sum_{i=1}^{s+1} \tbinom{iu+m}{m} < \sum_{i=1}^{s+1} k\cdot i^m \\ &= O((s+1)^{m+1}k) \\ &= O(M^{m+1}k) \\ &= O(m^{2(m+1)}u^{-m}n^{m+2}k). \end{aligned}$$

Hence Algorithm 2.1 can be implemented to run in time polynomials in the lengths and dimensions of the codes for all m.

3 Subfield Subcodes of RS Codes and Decoding

Alternant codes are subfield subcodes of Generalized Reed-Solomon codes, and it is shown [9, Theorem 15] that the Guruswami-Sudan algorithm can be applied to this situation. Following [12] in this section we will show that the q-ary Reed-Muller code $\mathcal{RM}_q(u,m)$ is a subfield subcode of a generalized Reed-Solomon code over $\mathbb{F}_{q^m}$, and then we give a list-decoding algorithm for the q-ary Reed-Muller code.

Let ζ be a primitive element of $\mathbb{F}_{q^m}$, then $\mathbb{F}_{q^m} = \{0, 1, \zeta, \zeta^2, \cdots, \zeta^{q^m-2}\}$. The field $\mathbb{F}_{q^m}$ can be viewed as an m-dimensional vector space over $\mathbb{F}_q$ with $1, \zeta, \cdots, \zeta^{m-1}$ as basis elements, every ζ^j can be expressed as a linear combination of $1, \zeta, \cdots, \zeta^{m-1}$

$$\zeta^j = \sum_{i=0}^{m-1} a_{ij}\zeta^i, \qquad 0 \le j \le q^m - 2,$$

where $a_{ij} \in \mathbb{F}_q$, $0 \le i \le m-1$, $0 \le j \le q^m - 2$. In other words, the elements of $\mathbb{F}_{q^m}$ can be written in the vector form as

$$\zeta^j = \begin{pmatrix} a_{0j} \\ a_{1j} \\ \vdots \\ a_{m-1,j} \end{pmatrix}, \qquad j = 0, 1, \cdots, q^m - 2.$$

Let $n = q^m$. The vector space $\mathbb{F}_q^m$ has n elements which are often called points. Let

$$P_0 := 0,\ P_j := (a_{0,j-1}, a_{1,j-1}, \ldots, a_{m-1,j-1}), \quad j = 1, \cdots, n-1. \tag{3.1}$$

Then $P_0, P_1, \cdots, P_{n-1}$ is an enumeration of the points of $\mathbb{F}_q^m$. Under this enumeration, a q-ary Reed-Muller code $\mathcal{RM}_q(u, m)$ of order u is defined as

$$\mathcal{RM}_q(u, m) = \{(f(P_0), f(P_1), \cdots, f(P_{n-1})) \mid f \in \mathbb{F}_q[X_1, \cdots, X_m],\ \deg(f) \le u\}.$$

For two vector $U = (a_1, \cdots, a_n)$ and $V = (b_1, \cdots, b_n)$, define the vector product as

$$UV = (a_1 b_1, a_2 b_2, \cdots, a_n b_n).$$

Now let

$$V_I = (1, 1, \cdots, 1) \in \mathbb{F}_q^n,$$
$$V_i = (0, a_{i0}, a_{i1}, \cdots, a_{i,n-2}), \quad i = 0, 1, \cdots, m-1.$$

With these vectors in $\mathbb{F}_q^n$, we construct a matrix G_u as

$$G_u = \begin{pmatrix} V_I \\ V_0^{k_0} V_1^{k_1} \cdots V_{m-1}^{k_{m-1}} \end{pmatrix},$$

with V_I and $V_0^{k_0} V_1^{k_1} \cdots V_{m-1}^{k_{m-1}}$ as rows for all nonnegative integers $k_0, k_1, \ldots, k_{m-1}$ such that $\sum_{t=0}^{m-1} k_t \le u$.

It is clear that the code over $\mathbb{F}_q$ generated by G_u is exactly the Reed-Muller code $\mathcal{RM}_q(u, m)$. From this representation of Reed-Muller codes, we can easily find a Reed-Solomon code over $\mathbb{F}_{q^m}$ such that $\mathcal{RM}_q(u, m)$ can be embedded into the Reed-Solomon code as a subfield subcode.

Let

$$\mathbf{v}_I = (1, 1, \cdots, 1) \in \mathbb{F}_q^{n-1},$$

$$\mathbf{v}_i = (a_{i0}, a_{i1}, \cdots, a_{i,n-2}), \quad i = 0, 1, \cdots, m-1.$$

Note that $\mathbf{v}_I$ and $\mathbf{v}_i$ are punctured vectors of V_I and V_i, respectively, with the first digits dropped. Let G_u^* be the matrix with $\mathbf{v}_I$ and $\mathbf{v}_0^{k_0}\mathbf{v}_1^{k_1}\cdots\mathbf{v}_{m-1}^{k_{m-1}}$ as rows for all nonnegative integers $k_0, k_1, \ldots, k_{m-1}$ such that $\sum_{t=0}^{m-1} k_t \leq u$. We denote the code over $\mathbb{F}_q$ generated by G_u^* as $\mathcal{RM}_q^*(u,m)$. It is easy to see that $\mathcal{RM}_q^*(u,m)$ is the punctured code of $\mathcal{RM}_q(u,m)$ with the first digit dropped. Conversely $\mathcal{RM}_q(u,m)$ is the extended code of $\mathcal{RM}_q^*(u,m)$. Hence

$$(\xi, \mathbf{c}^*) \in \mathcal{RM}_q(u,m) \text{ if and only if } \mathbf{c}^* \in \mathcal{RM}_q^*(u,m) \text{ and } \xi + \sum c_i^* = 0.$$

It is well known that for given q, m and u, let $u^\perp = m(q-1) - u - 1$, then the dual code of $\mathcal{RM}_q(u,m)$ is equal to $\mathcal{RM}_q(u^\perp, m)$. Let ρ be the rest after division of $u^\perp + 1$ by $q-1$ with quotient σ, that is

$$u^\perp + 1 = \sigma(q-1) + \rho, \text{ where } \rho < q-1.$$

Define $d = (\rho+1)q^\sigma$. Then $d-1$ is the minimum distance of $\mathcal{RM}_q^*(u,m)$ and d is the minimum distance of $\mathcal{RM}_q(u,m)$.

Let h be an integer such that $0 \leq h \leq q^m - 1$. Express h in radix-q form

$$h = \delta_0 + \delta_1 q + \delta_2 q^2 + \cdots + \delta_{m-1} q^{m-1}.$$

Define the weight of h as

$$W(h) = \delta_0 + \delta_1 + \delta_2 + \cdots + \delta_{m-1}.$$

The following proposition is taken from Theorem 5 and Corollary 2 of [12].

Proposition 3.1 *Let $d-1$ be the minimum distance of $\mathcal{RM}_q^*(u,m)$. Then the q-ary code $\mathcal{RM}_q^*(u,m)$ is the subfield subcode of the BCH code over $\mathbb{F}_{q^m}$ whose generator polynomial has $\zeta, \zeta^2, \cdots, \zeta^{d-2}$ as all its roots. Furthermore $\mathcal{RM}_q^*(u,m)$ is the cyclic code over $\mathbb{F}_q$ with generator polynomial $g(X)$ such that ζ^h is a zero of $g(X)$ if and only if $0 < W(h) \leq u^\perp$.*

Denote by C_{BCH} the BCH code over $\mathbb{F}_{q^m}$ whose generator polynomial has $\zeta, \zeta^2, \cdots, \zeta^{d-2}$ as all its roots. Then

$$\mathcal{RM}_q^*(u,m) = C_{BCH} \cap \mathbb{F}_q^{n-1}.$$

The following matrix H is a parity-check matrix of C_{BCH}

$$H = \begin{pmatrix} 1 & \zeta & \zeta^2 & \cdots & \zeta^{n-2} \\ 1 & \zeta^2 & \zeta^4 & \cdots & \zeta^{2(n-2)} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & \zeta^{d-2} & \zeta^{2(d-2)} & \cdots & \zeta^{(d-2)(n-2)} \end{pmatrix}.$$

Let $\mathbf{a}^* = (1, \zeta, \zeta^2, \cdots, \zeta^{n-2})$. Then the matrix H is a generator matrix of the generalized Reed-Solomon code $GRS_{d-2}(\mathbf{a}^*, \mathbf{a}^*)$ over $\mathbb{F}_{q^m}$, see [17, ch.10 §8]. So C_{BCH} is the dual code of $GRS_{d-2}(\mathbf{a}^*, \mathbf{a}^*)$.

By [17, Theorem 4, ch.10 §8], the dual of $GRS_{d-2}(\mathbf{a}^*, \mathbf{a}^*)$ is a generalized Reed-Solomon code $GRS_{n-d+1}(\mathbf{a}^*, \mathbf{b}^*)$ for some $\mathbf{b}^*$. From the fact that

$$1 + \zeta^i + \zeta^{2i} + \cdots + \zeta^{(n-1)i} = 0, \quad \text{for } 1 \leq i \leq n-2,$$

we have that

$$G = \begin{pmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \zeta & \zeta^2 & \cdots & \zeta^{n-2} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & \zeta^{n-d} & \zeta^{2(n-d)} & \cdots & \zeta^{(n-2)(n-d)} \end{pmatrix}$$

is a generator matrix of $GRS_{n-d+1}(\mathbf{a}^*, \mathbf{b}^*)$ with $\mathbf{b}^* = \mathbf{1} = (1, \cdots, 1)$. And $GRS_{n-d+1}(\mathbf{a}^*, \mathbf{1})$ has a parity-check matrix H. Hence,

$$C_{BCH} = GRS_{n-d+1}(\mathbf{a}^*, \mathbf{1}).$$

Therefore, we have embedded $\mathcal{RM}_q^*(u, m)$ into $GRS_{n-d+1}(\mathbf{a}^*, \mathbf{1})$ as the subfield subcode, where $d-1$ is the minimum distance of $\mathcal{RM}_q^*(u, m)$.

Let $\mathbf{a} = (0, \mathbf{a}^*) = (0, 1, \zeta, \cdots, \zeta^{n-2})$. Then $GRS_{n-d+1}(\mathbf{a}, \mathbf{1})$ is the extended code of $GRS_{n-d+1}(\mathbf{a}^*, \mathbf{1})$. Hence $\mathcal{RM}_q(u, m)$ is the subfield subcode of $GRS_{n-d+1}(\mathbf{a}, \mathbf{1})$, where d is the minimum distance of $\mathcal{RM}_q(u, m)$, i.e.,

$$\mathcal{RM}_q(u, m) = GRS_{n-d+1}(\mathbf{a}, \mathbf{1}) \cap \mathbb{F}_q^n.$$

$GRS_{n-d+1}(\mathbf{a}, \mathbf{1})$ is a $[n, n-d+1, d]$ Reed-Solomon code over $\mathbb{F}_{q^m}$. We can decode $GRS_{n-d+1}(\mathbf{a}, \mathbf{1})$ using the list-decoding algorithm of Guruswami-Sudan in [9]. We give a list-decoding algorithm for $\mathcal{RM}_q(u, m)$ as follows.

Algorithm 3.1 (*List Decoding of RM Codes as Alternant Codes*)

Input: $n = q^m$, $\mathbf{y} = (y_0, y_1, \cdots, y_{n-1}) \in \mathbb{F}_q^n$.

Step 0: (1) Compute the minimum distance d of $\mathcal{RM}_q(u, m)$ and a parameter $E = \lceil n - \sqrt{n(n-d)} - 1 \rceil$.

(2) Construct the extension field $\mathbb{F}_{q^m}$ using an irreducible polynomial of degree m over $\mathbb{F}_q$.

(3) Find a primitive element ζ of $\mathbb{F}_{q^m}$. Generate the code $GRS_{n-d+1}(\mathbf{a}, \mathbf{1})$ with $\mathbf{a} = (0, 1, \zeta, \cdots, \zeta^{n-2})$.

Step 1: Using Guruswami-Sudan algorithm find the set $\mathcal{L}^{(1)}$ of all the codewords $\mathbf{c}$ of $GRS_{n-d+1}(\mathbf{a}, \mathbf{1})$ satisfying

$$d(\mathbf{c}, \mathbf{y}) \leq E.$$

Step 2: For every $\mathbf{c} \in \mathcal{L}^{(1)}$, check if $\mathbf{c} \in \mathbb{F}_q^n$, if so, forward $\mathbf{c}$ into $\mathcal{L}$. Output $\mathcal{L}$.

From [9, Theorem 8 and Proposition 9] we have the following theorem.

Theorem 3.2 *Assume d is the minimum distance of the q-ary Reed-Muller code $\mathcal{RM}_q(u, m)$. Then $\mathcal{RM}_q(u, m)$ is (E, M)-decodable, provided that*

$$E < n - \sqrt{n(n-d)} \qquad \text{and} \quad M = O(\sqrt{(n-d)n^3}).$$

The algorithm above correctly finds all the E-consistent codewords for any received vector $\mathbf{y} \in \mathbb{F}_q^n$.

Remark 3.1: Note that Algorithm 3.1 outputs a set of E-consistent codewords of the q-ary Reed-Muller code defined by the enumeration of points of $\mathbb{F}_q^m$, say $P_0, P_1, \cdots, P_{n-1}$, given by (3.1). If $\mathcal{RM}_q(u, m)$ is defined by another enumeration of the points of $\mathbb{F}_q^m$, namely $P'_0, P'_1, \cdots, P'_{n-1}$, we can get the correct E-consistent codewords by the following steps: (1) Find the permutation π such that $P_i = P'_{\pi(i)}$, $i = 0, 1, \cdots, n-1$, and the inverse permutation π^{-1}. (2) Let $\mathbf{y}^* = (y_{\pi(0)}, y_{\pi(1)}, \cdots, y_{\pi(n-1)})$. Then, go to Steps 0-2 of Algorithm 3.1 with $\mathbf{y}^*$. (3) For every codeword $\mathbf{c} = (c_0, c_1, \cdots, c_{n-1}) \in \mathcal{L}$, let

$\pi^{-1}(\mathbf{c}) = (c_{\pi^{-1}(0)}, c_{\pi^{-1}(1)}, \cdots, c_{\pi^{-1}(n-1)})$. Then, $\pi^{-1}(\mathcal{L}) = \{\pi^{-1}(\mathbf{c}) \mid \mathbf{c} \in \mathcal{L}\}$ is the set of E-consistent codewords of $\mathcal{RM}_q(u, m)$. □

Remark 3.2: It is clear that Algorithm 3.1 works for $\mathcal{RM}_q(u, m)$ with any $u > 0$. So the algorithm is applicable to q-ary Reed-Muller codes of any rates. Theorem 3.2 gives an error-correction bound of Algorithm 3.1, $E < n - \sqrt{n(n-d)}$. When $u < q$, the minimum distance of $\mathcal{RM}_q(u, m)$ is given by $d = (q-u)q^{m-1}$. In this case, $n - \sqrt{n(n-d)} = n(1 - \sqrt{u/q}) > n\left(1 - \sqrt[m+1]{u/q}\right)^m$. □

In Step 0 of Algorithm 3.1, to construct the extension field $\mathbb{F}_{q^m}$, it requires to find an irreducible polynomial $g(x)$ of degree m over $\mathbb{F}_q$. It is well known that there are efficient algorithms for finding irreducible polynomials over finite fields [25]. In [23], a probabilistic algorithm is given for finding an irreducible polynomial of degree m over $\mathbb{F}_q$ with expected number of $O((m^2 \log m + m \log q) \log m \log\log m)$ field operations in $\mathbb{F}_q$.

On the other hand, to generate the Reed-Solomon code $GRS_{n-d+1}(\mathbf{a}, \mathbf{1})$ over $\mathbb{F}_{q^m}$, we need to find a primitive element of $\mathbb{F}_{q^m}$. From [25], a primitive element of $\mathbb{F}_{q^m}$ can be found in deterministic time $O((q^m)^{1/4+\varepsilon}) = O(n^{1/4+\varepsilon})$, where $n = q^m$ is the length of the code, ε denotes an arbitrary positive number.

Step 1 of Algorithm 3.1 can be implemented using Guruswami-Sudan algorithm in [9] for Reed-Solomon code $GRS_{n-d+1}(\mathbf{a}, \mathbf{1})$ over $\mathbb{F}_{q^m}$. We can also use directly Algorithm 2.1 for $\mathcal{RM}_{q^m}(n-d, 1) = GRS_{n-d+1}(\mathbf{a}, \mathbf{1})$. From [9, Themorem 12 and Corollary 13], if $t^2 > (1+\delta)n(n-d)$ and $E = n - t < n - \sqrt{n(n-d)}$, then Guruswami-Sudan algorithm can be implemented to run in $O(n^3\delta^{-6})$ field operations in $\mathbb{F}_{q^m}$. If furthermore the rate of Reed-Solomon codes is fixed, the complexity of this algorithm is $O(n^3)$.

So, the implementation of Algorithm 3.1 requires $O(n)$ field operations in $\mathbb{F}_q$ and $O(n^3)$ field operations in $\mathbb{F}_{q^m}$.

4 One-Point AG Codes and Decoding

It is known that every linear code is a weakly algebraic-geometric code (AG) by [18]. In this section we prove that for appropriate u the Reed-Muller code $\mathcal{RM}_q(u, m)$ can be described as a one-point algebraic-geometric code. Then using Guruswami-Sudan algorithm for algebraic-geometric codes [9] we give a list-decoding algorithm for the Reed-Muller codes.

Let $\mathcal{X}$ be a projective, absolutely irreducible, reduced and non-singular curve defined over the finite field $\mathbb{F}_q$. Let $\mathcal{P} = (P_1, \ldots, P_n)$ be an enumeration of n distinct $\mathbb{F}_q$-rational points of $\mathcal{X}$. Let G be a $\mathbb{F}_q$-rational divisor of $\mathcal{X}$. Let $L(G)$ be the vector space defined as

$$L(G) = \{f \text{ is a rational function on } \mathcal{X},\ (f) + G \geq 0 \ \text{ or } f = 0\}.$$

Suppose that P_i is not in the support of G for $i = 1, \cdots, n$. Then the evaluation map

$$ev_{\mathcal{P}} : L(G) \longrightarrow \mathbb{F}_q^n$$

with $ev_{\mathcal{P}}(f) = (f(P_1), \ldots, f(P_n))$ is well defined. Its image is denoted by $C_L(\mathcal{X}, \mathcal{P}, G)$ or $C_L(\mathcal{P}, G)$ for short. If C is a linear code and $C = C_L(\mathcal{X}, \mathcal{P}, G)$ for some $\mathcal{X}$, $\mathcal{P}$ and G as above, then this is called a *weakly algebraic-geometric* (WAG) *representation* of C. See [18]. If moreover $\deg(G) < n$, then the code is called *algebraic geometric* (AG) and the parameters $[n, k, d]$ of this code satisfy

$$k \geq \deg(G) + 1 - g \quad \text{ and } \quad d \geq n - \deg(G).$$

This code is called *strongly algebraic geometric* (SAG) if

$$2g - 2 < \deg(G) < n.$$

Then we have equality for the lower bound of the dimension, i.e.,

$$k = \deg(G) + 1 - g.$$

The number $d' := n - \deg(G)$ is called the *designed minimum distance* of the code. If P is an $\mathbb{F}_q$-rational point that is distinct form the P_i and $G = lP$, then the code is called a *one point* (weakly or strongly) algebraic-geometric code.

Sudan's decoding algorithm has been applied to AG codes, see [9, Theorem 27] and [22] with the following result.

Proposition 4.1 *Let C be a one-point algebraic-geometric code of length n and designed distance d'. Then there is a list-decoding algorithm of polynomial time complexity that corrects up to E errors if*

$$E < n - \sqrt{n(n - d')}.$$

Let p be a prime and q a power of p. Let $\mathcal{X}(m,q)$ be the scheme over $\mathbb{F}_p$ in the m dimensional projective space $\mathbb{P}^m$ defined by the homogeneous ideal

$$I(m,q) = (X_i^{q+1} - X_i^2 X_0^{q-1} + X_{i+1} X_0^q - X_{i+1}^q X_0, i = 1, \ldots, m-1)$$

in $\mathbb{F}_q[X_0, \ldots, X_m]$. According to Propositions 3 and 4 of [18] the following holds.

Proposition 4.2 *The scheme $\mathcal{X}(m,q)$ is a projective, absolutely irreducible, reduced curve over $\mathbb{F}_p$. It has exactly one point $P_\infty = (0 : 0 : \cdots : 0 : 1)$ at the hyperplane H with equation $X_0 = 0$, the curve is nonsingular outside P_∞ and goes through all the q^m rational points of $\mathbb{P}^m$ over $\mathbb{F}_q$ outside the hyperplane H. The normalization of $\mathcal{X}(m,q)$ has genus $g(m,q)$, where*

$$g(m,q) = \tfrac{1}{2}\left((q^2-1)(q+1)^{m-1} - q^{m+1} + 1\right).$$

Let

$$\mathcal{N} : \mathcal{Y}(m,q) \longrightarrow \mathcal{X}(m,q)$$

be the normalization of $\mathcal{X}(m,q)$. Then there is exactly one point Q_∞ on $\mathcal{Y}(m,q)$ above P_∞. Let $K_\infty(Q_\infty)$ be the ring of all rational functions on $\mathcal{Y}(m,q)$ that have no poles outside Q_∞. Let $z_i = (X_i/X_0) \circ \mathcal{N}$. Then $K_\infty(Q_\infty) = \mathbb{F}_q[z_1, \ldots, z_m]$, see [19, Example 3]. Let v_∞ be the valuation at Q_∞ counting the pole order at Q_∞ of a rational function on $\mathcal{Y}(m,q)$. Then z_i is a rational function on $\mathcal{Y}(m,q)$ and

$$v_\infty(z_i) = -q^{m-i}(q+1)^{i-1}.$$

The vector space $L(lQ_\infty)$ consists of all rational functions on $\mathcal{Y}(m,q)$ that have no poles outside Q_∞ and with pole order at most l at Q_∞. And

$$K_\infty(Q_\infty) = \cup_{l=1}^\infty L(lQ_\infty).$$

Proposition 6 of [18] gives

Proposition 4.3 *The vector space $L(lQ_\infty)$ over $\mathbb{F}_q$ is generated by the set of all*

$$z_1^{k_1} \cdots z_m^{k_m}$$

such that

$$\sum_{i=1}^m k_i q^{m-i}(q+1)^{i-1} \leq l.$$

Let $n = q^m$. Let b be a positive integer. Let $\mathbf{P} = (P_1, \ldots, P_n)$ be an enumeration of all the points of $\mathbb{F}_q^m$. Now $\mathbb{F}_q^m$ is the set of all $\mathbb{F}_q$-rational points of the affine space $\mathbb{A}^m$ which we embed in the projective space $\mathbb{P}^m$ by the map $(a_1, \ldots, a_m) \mapsto (1 : a_1 : \ldots : a_m)$. Let Q_i be the unique point of $\mathcal{Y}(m, q^b)$ above P_i. Let $\mathbf{Q} = (Q_1, \ldots, Q_n)$. Consider the evaluation maps

$$ev_{\mathbf{P}} : \mathbb{F}_q[Y_1, \ldots, Y_m] \longrightarrow \mathbb{F}_q^n$$

and

$$ev_{\mathbf{Q}} : K_\infty(Q_\infty) \longrightarrow \mathbb{F}_q^n.$$

Consider the ring morphism

$$\theta : \mathbb{F}_q[Y_1, \ldots, Y_m] \longrightarrow K_\infty(Q_\infty)$$

defined by $\theta(Y_i) := z_i$. Then $ev_{\mathbf{P}} = ev_{\mathbf{Q}} \circ \theta$.

The image under $ev_{\mathbf{P}}$ of all polynomials of degree at most u is $\mathcal{RM}_q(u, m)$. The image under $ev_{\mathbf{Q}}$ of $L(lQ_\infty)$ is the weakly algebraic geometry code on the curve $\mathcal{Y}(m, q^b)$ with respect to the points $\mathbf{Q}$ and the divisor lQ_∞ and is denoted by $C_L(\mathcal{Y}(m, q^b), \mathbf{Q}, lQ_\infty)$. By Proposition 4.3, we have that for any $b \geq 1$,

$$\mathcal{RM}_q(u, m) \subseteq C_L(\mathcal{Y}(m, q^b), \mathbf{Q}, u(q^b + 1)^{m-1} Q_\infty).$$

Especially, $\mathcal{RM}_q(u, m) \subseteq C_L(\mathcal{Y}(m, q), \mathbf{Q}, u(q + 1)^{m-1} Q_\infty)$.

We will have a close look at b in $l = u(q^b + 1)^{m-1}$. Choose the positive integer b such that

$$q^b > (\sqrt[m-1]{1 + 1/u} - 1)^{-1}.$$

The inequality above is $u(q^b + 1)^{m-1} < (u + 1)q^{b(m-1)}$, so by this choice of b we have that

$$u(q^b + 1)^i < (u + 1)q^{bi} \quad \text{for } i = 0, 1, \ldots, m - 1.$$

From these inequalities above, we can prove that

$$\sum_{i=1}^{m} k_i q^{b(m-i)} (q^b + 1)^{i-1} \leq u(q^b + 1)^{m-1} \quad \text{if and only if} \quad k_1 + \cdots + k_m \leq u.$$

Therefore the monomials

$$z_1^{k_1} \cdots z_m^{k_m} \quad \text{with} \quad k_1 + \cdots + k_m \leq u$$

generate the space $L(u(q^b + 1)^{m-1} Q_\infty)$ by Proposition 4.3. In this way we have shown the following proposition.

Proposition 4.4 *Let*

$$q^b > (\sqrt[m-1]{1+1/u} - 1)^{-1}.$$

Then

$$\mathcal{RM}_q(u,m) = C_L(\mathcal{Y}(m,q^b), \mathbf{Q}, u(q^b+1)^{m-1}Q_\infty).$$

This representation of $\mathcal{RM}_q(u,m)$ is algebraic geometric if and only if $u(q^b+1)^{m-1} < q^m$, which is equivalent to $b=1$ and $u < q^m/(q+1)^{m-1}$.

Now by Propositions 4.3 and 4.4, we the following corollary

Corollary 4.5 *Denote* $C_L := C_L(\mathcal{Y}(m,q), \mathbf{Q}, u(q+1)^{m-1}Q_\infty)$. *Then* $\mathcal{RM}_q(u,m) \subseteq C_L$. *Moreover*

(1) If $u < q^m/(q+1)^{m-1}$, *then* C_L *is a one-point algebraic-geometric code.*

(2) If $u < \min\left\{\frac{q^m}{(q+1)^{m-1}}, \frac{q^{m-1}}{(q+1)^{m-1}-q^{m-1}}\right\}$, *then* C_L *is a one-point algebraic-geometric code and* $\mathcal{RM}_q(u,m) = C_L$.

The following is a list-decoding algorithm for $\mathcal{RM}_q(u,m)$ with $u < q^m/(q+1)^{m-1}$.

Algorithm 4.1 (*List Decoding of RM Codes as AG Codes*)

Input: $n = q^m$, $\mathbf{y} = (y_1, y_2, \cdots, y_n) \in \mathbb{F}_q^n$.

Step 0: Compute a parameter $E = \left\lceil n\left(1 - \sqrt{\frac{u(q+1)^{m-1}}{n}}\right) - 1\right\rceil$.

Step 1: Using Guruswami-Sudan algorithm find the set $\mathcal{L}^{(1)}$ of all the codewords $\mathbf{c}$ of C_L satisfying

$$d(\mathbf{c}, \mathbf{y}) \le E.$$

Step 2: For every $\mathbf{c} \in \mathcal{L}^{(1)}$, check if $\mathbf{c} \in \mathcal{RM}_q(u,m)$, if so, forward $\mathbf{c}$ into $\mathcal{L}$. Output $\mathcal{L}$.

Theorem 4.6 *Assume that $u < q^m/(q+1)^{m-1}$. The algorithm above is a list-decoding algorithm for $\mathcal{RM}_q(u,m)$, which works for up to E errors, provided that*

$$E < n\left(1 - \sqrt{\frac{u(q+1)^{m-1}}{n}}\right).$$

Remark 4.1: From the theorem above, Algorithm 4.1 works only for $\mathcal{RM}_q(u,m)$ with $u < q^m/(q+1)^{m-1} < q$. Since $n = q^m$, when q is large, $u(q+1)^{m-1}/n$ is very close to u/q. So, for large q, $n(1-\sqrt{u(q+1)^{m-1}/n}) \approx n(1-\sqrt{u/q}) \geq n(1-\sqrt[m+1]{u/q})^m$. Therefore, when $u < q^m/(q+1)^{m-1}$ and q is large, the error-correction bound of Algorithm 4.1 is better than that of Algorithm 2.1. □

Now let us consider the complexity of Algorithm 4.1. Denote by $\mathcal{K}$ the function field of $\mathcal{Y}(m,q)$. From [9, Proposition 22], Step 1 of Algorithm 4.1 can be reduced to a root-finding problem over the function field $\mathcal{K}$ of a univariate polynomial $H(y)$ of degree $s \leq \frac{l}{n-d'}$ with at most $O\left(\frac{l^6}{(n-d')^3}\right)$ operations over $\mathbb{F}_q$ and $O(nl^2)$ operations over $\mathcal{K}$, where $d' = n - \deg(u(q+1)^{m-1}Q_\infty) = n - u(q+1)^{m-1}$, and l is a parameter given by

$$l = O\left(\max\left\{\frac{g(n-E)+n(n-d')}{(n-E)^2-n(n-d')}, n-E\right\}\right),$$

$g = g(m,q)$ is the genus of $\mathcal{Y}(m,q)$.

Using the algorithm in [31] the root-finding problem of $H(y)$ can be solved with $O(ks(n^2+s^2+\log^2 s\cdot \text{loglog}s\cdot \log q))$ operations over $\mathbb{F}_q$ and $O(ks^2)$ operations over $\mathcal{K}$. So the implementation of Algorithm 4.1 requires $O\left(\frac{l^6}{(n-d')^3} + ks(n^2+s^2+\log^2 s\cdot \text{loglog}s\cdot \log q)\right)$ operations over $\mathbb{F}_q$ and $O(nl^2+ks^2)$ operations over $\mathcal{K}$.

Assume $\delta = n-\sqrt{n(n-d')}-E$ is fixed such that $0 < \delta \leq 1$. Then $n-E = \sqrt{n(n-d')}+\delta = O(n)$. On the other hand, $g = \frac{1}{2}((q^2-1)(q+1)^{m-1} - q^{m+1}+1) = O(n)$. We have

$$l = O\left(\max\left\{\frac{g(n-E)+n(n-d')}{(n-E)^2-n(n-d')}, n-E\right\}\right) = O(n^{\frac{3}{2}}),$$

since $(n-E)^2 - n(n-d') = 2\delta\sqrt{n(n-d')} + \delta^2 \geq 2\delta\sqrt{n}$. Then Algorithm 4.1 can be implemented in $O(n^9)$ operations over $\mathbb{F}_q$ and $O(n^4)$ operations over $\mathcal{K}$.

5 Analysis and Comparison

In this section, we will compare the proposed list-decoding algorithms with each other and with the known algorithms of [1] and [27].

In [1] and [27], randomized algorithms for $\mathcal{RM}_q(u,m)$ were given, for any codewords $\mathbf{c}$ within relative Hamming distance $(1-\varepsilon)$ to the received word $\mathbf{y}$, i.e., $d(\mathbf{c},\mathbf{y})/n \leq 1-\varepsilon$, those algorithms correctly find $\mathbf{c}$. In [1] the parameter ε (which is called fraction of agreement) is an unspecified polynomial in u and $1/q$. The result in [27] is a strengthening of the result in [1], and the algorithm in [27] works for a smaller fraction of agreement given by $\varepsilon > c\sqrt{u/q}$, where the constant c can be pushed down to any constant greater than $\sqrt{2}$ assuming u/q is sufficiently small. This means that the algorithm in [27] works for at most $E < n(1-\varepsilon)$ errors where $\varepsilon > \sqrt{2}\cdot\sqrt{u/q}$, and the algorithm is applicable only to those $\mathcal{RM}_q(u,m)$ with $u < q/2$.

Consider the error-correction capabilities of our algorithms. By Remark 2.1, Algorithm 2.1 works for up to $E < n(1-\sqrt[m+1]{u/q})^m$ errors. It is clear that $(1-\sqrt[m+1]{u/q})^m$ is often smaller than $1-\sqrt{2}\cdot\sqrt{u/q}$. But Algorithm 2.1 is applicable to $\mathcal{RM}_q(u,m)$ with $u < q$. From Theorem 3.2, Algorithm 3.1 works for up to $E < n(1-\sqrt{(n-d)/n})$, where d is the minimum distance of the code. Since when $u < q$, for $\mathcal{RM}_q(u,m)$ we have $n = q^m$ and $d = (q-u)q^{m-1}$. So, $1-\sqrt{(n-d)/n} = 1-\sqrt{u/q} > 1-\sqrt{2}\cdot\sqrt{u/q}$. Therefore, Algorithm 3.1 has a better error-correction capability than the list decoding algorithm in [27]. More importantly, Algorithm 3.1 is applicable to any q-ary Reed-Muller code. Next consider Algorithm 4.1. From Theorem 4.6, this algorithm is applicable to Reed-Muller codes $\mathcal{RM}_q(u,m)$ with $u < q^m/(q+1)^{m-1}$, and it works for up to $E < n\left(1-\sqrt{\frac{u(q+1)^{m-1}}{n}}\right)$ errors. Since $n = q^m$, when m is fixed and q is sufficiently large, $1-\sqrt{\frac{u(q+1)^{m-1}}{n}}$ is very close to $1-\sqrt{u/q}$ and greater than $1-\sqrt{2}\cdot\sqrt{u/q}$. Thus, Algorithm 4.1 can be also better than the algorithm in [27] in the sense of error-correction capability.

To compare the proposed algorithms with each other, we define the following error-correction-capability (ECC) functions:

$$F(u) = \frac{d}{2}, \qquad u > 0,$$

$$G(u) = n(1-\sqrt[m+1]{u/q})^m, \qquad 0 < u < q,$$

$$H(u) = n(1-\sqrt{(n-d)/n}), \qquad u > 0,$$

$$V(u) = n(1 - \sqrt{u(q+1)^{m-1}/n}), \qquad 0 < u < q^m/(q+1)^{m-1}.$$

The traditional decoding algorithms can correct at most $E \leq \lceil F(u) - 1 \rceil$ errors; $E \leq \lfloor G(u) - 1 \rfloor$ gives a upper bound for the error-correction capability of Algorithm 2.1; Algorithm 3.1 works for up to $E \leq \lceil H(u) - 1 \rceil$ errors; and Algorithm 4.1 works for up to $E \leq \lceil V(u) - 1 \rceil$ errors. We have following properties:

(5.1) *For sufficiently small u, $G(u) > F(u)$, and Algorithm 2.1 can correct more errors than traditional decoding algorithms.*

(5.2) *For any positive integer u, $H(u) > F(u)$. Algorithm 3.1 can correct more errors than any traditional decoding algorithm.*

(5.3) *When $u < q$, $H(u) \geq G(u)$, so, Algorithm 3.1 can correct more errors than Algorithm 2.1.*

(5.4) *For $0 < u < q^m/(q+1)^{m-1}$, $H(u) > V(u)$. So, Algorithm 3.1 can correct more errors than Algorithm 4.1. But when q is large, $V(u) \approx H(u)$, the error-correction capability of Algorithm 4.1 is close to that of Algorithm 3.1.*

The following figure illustrates the error-correction capabilities of the proposed algorithms and traditional decoding algorithms. In the figure, $q = 64$ and $m = 2$, the values of functions $F(u)$, $G(u)$, $H(u)$ and $V(u)$ are shown for $0 < u \leq 63$.

Figure 1: Comparison of the ECC functions

Regarding the time complexity of the proposed algorithms, we have proved in Section 2.4 that the implementation of Algorithm 2.1 requires $O(N^3)$ operations in $\mathbb{F}_q$, where N is bounded from above by a polynomial in m, the code length n and the dimension k. Let $t > \sqrt{(1+\delta)n(n-d)}$. Assume that the number of errors is at most $E < n - t$. Then Step 1 of Algorithm 3.1 requires $O(n^3\delta^{-6})$ operations in the extension field $\mathbb{F}_{q^m}$. And we need to construct the extension field $\mathbb{F}_{q^m}$ and to generate the Reed-Solomon code $GRS_{n-d+1}(\mathbf{a}, \mathbf{1})$ over $\mathbb{F}_{q^m}$ before decoding. The implementation of Algorithm 4.1 requires $O\left(\frac{l^6}{(n-d')^3} + ks(n^2 + s^2 + \log^2 s \cdot \mathrm{loglog} s \cdot \log q)\right)$ operations over $\mathbb{F}_q$ and $O(nl^2 + ks^2)$ operations over the function field $\mathcal{K}$, where $d' = n - \deg(u(q+1)^{m-1}Q_\infty) = n - u(q+1)^{m-1}$, and l is a parameter given by

$$l = O\left(\max\left\{\frac{g(n-E) + n(n-d')}{(n-E)^2 - n(n-d')}, n - E\right\}\right),$$

$g = g(m, q)$ is the genus of $\mathcal{Y}(m, q)$. Under some assumption, Algorithm 4.1 can be implemented in $O(n^9)$ operations over $\mathbb{F}_q$ and $O(n^4)$ operations over $\mathcal{K}$.

We mention that for fixed and small m Algorithm 2.1 runs over the field $\mathbb{F}_q$ with low complexity, and when u/q is sufficiently small, the error-correction capability of Algorithm 2.1 can be close to that of Algorithms 3.1 and 4.1 (one can also see this from Figure 1). So in some practical applications, Algorithm 2.1 can be preferable for the decodings of Reed-Muller codes of small rates.

6 Conclusions

In this paper, viewing q-ary Reed-Muller codes as codes from order domains, we present a list-decoding algorithm for q-ary Reed-Muller codes, which is a straightforward generalization of the list-decoding algorithm of Reed-Solomon codes by Guruswami and Sudan in [9]. The algorithm works for up to $\lfloor n(1 - \sqrt[m+1]{u/q})^m - 1 \rfloor$ errors, and it is applicable to codes $\mathcal{RM}_q(u, m)$ with $u < q$. This algorithm can be implemented to run in time polynomial in the length of the codes. We show that q-ary Reed-Muller codes are subfield subcodes of Reed-Solomon codes, and then present a second list-decoding algorithm for q-ary Reed-Muller codes. This algorithm works for codes with any rates, and achieves an error-correction bound $\lceil n(1 - \sqrt{(n-d)/n}) - 1 \rceil$. The second algorithm achieves a better error-correction bound than the algorithm in [27]. The implementation of the second algorithm requires $O(n)$ field operations in $\mathbb{F}_q$ and $O(n^3)$ field operations in $\mathbb{F}_{q^m}$ under some assumption. Also, we prove that q-ary Reed-Muller codes can be described as one-point AG codes. And using the algorithm of AG codes in [9], we give a third list-decoding algorithm for Reed-Muller codes. The third algorithm achieves an error-correction bound $\lceil n(1 - \sqrt{u(q+1)^{m-1}/n}) - 1 \rceil$. The time complexity of the third algorithm is also bounded from above by polynomials in the length of the codes.

Acknowledgement We like to thank Tom Høholdt and Diego Ruano for noticing a mistake in the proof of Lemma 2.9 in a previous version of this paper.

References

[1] S. Arora and M. Sudan, "Improved low-degree testing and its applications," *Proceedings of the 29th Annual ACM Symposium on Theory of Computing*, pp.485-495, 1997.

[2] M. Ben-Or, "Probabilistic algorithms in finite fields," in *Proceedings of the 22nd Annual IEEE Symposium on Foundations of Computer Science*, 1981, pp. 394–398.

[3] G.F. Clements and B. Lindström, "A generalization of a combinatorial theorem of Macaulay," *Journ. Combinatorial Theory*, vol. 7, pp. 230-238, 1969.

[4] P. Delsarte, "On subfield subcodes of Reed-Solomon codes," *IEEE Trans. Inform. Thoery,* vol. 21, pp. 575-576, 1975.

[5] P. Elias, "List decoding for noisy channel," *Tech. Rep. 335, Res. Lab. Electron.*, MIT, Cambridge, MA, 1957.

[6] O. Geil and T. Høholdt, "Footprints or Generalized Bezout's Theorem," *IEEE Trans. Inform. Theory*, vol. 46, no. 2, Mar. 2000, pp. 635-641.

[7] O. Geil and R. Pellikaan, "On the Structure of Order Domains," to apear in *Finite Fields and their Applications* 2002.

[8] O. Goidreich, R. Rubinfeld, and M. Sudan, "Learning polynomials with queries: The high noise case," in *Proc. 36th Annu. IEEE Symp. Foundations of Computer Science*, 1995, pp. 294-303.

[9] V. Guruswami and M. Sudan,"Improved Decoding of Reed-Solomon and Algebraic-Geometry Codes," *IEEE Trans. Inform. Theory*, vol. 45, no. 6, Nov. 1999, pp. 1757–1767.

[10] P. Heijnen and R. Pellikaan, "Generalized Hamming weights of q-ary Reed-Muller codes," *IEEE Trans. Inform. Theory,* vol. 44, pp. 181-196, 1998.

[11] T. Høholdt and R. Nielsen, "Decoding Hermitian codes with Sudan's algorithm," in *Applied Algebra, Algebraic Algorithms and Error-Correcting Codes*, N. Fossorier, H. Imei, S. Lin, A. Pole (Eds), Lecture Notes in Computer Science, Vol. 1719, Springer, 1999, pp. 260-270.

[12] T. Kasami, S. Lin and W. Peterson, "New Generalization of Reed-Muller Codes, Part I: Primitive Codes," *IEEE Trans. Inform. Theory*, vol. 14, no. 2, 1968, pp. 189–199.

[13] G.O.H. Katona, "A theorem of finite sets," in *Theory of Graphs, Proceedings of the Colloquium held at Tihany*, Hungary, September 1966, ed. by P. Erdös and G. Katona, Academic Press, New York; Akademia Kiado, Budapest 1968, pp. 187-207.

[14] J.B. Kruskal, "The optimal number of simplices in a complex," in *Math. Optimization Techniques,* Univ. Calif. Press, Berkeley, California, 1963, pp. 251-268.

[15] S. Lang, *Algebra* , Addison-Wesley Publ. Comp., Reading Massachusetts, 1965.

[16] F.S. Macaulay, "Some properties of enumeration in the theory of modular systems," *Proc. London Math. Soc.*, vol. 26, pp. 531-555, 1927.

[17] F.J. MacWilliams and N.J.A. Sloane, *The Theory of Error-Correcting Codes,* North-Holland Math. Library, vol. 16, Elsevier Sc. Publ., New York 1972.

[18] R. Pellikaan, B.-Z. Shen and G.J.M. van Wee, "Which linear codes are algebraic-geometric ?," *IEEE Trans. Inform. Theory,* vol. 37, pp. 583-602, 1991.

[19] S.C. Porter, B.-Z. Shen and R. Pellikaan, "Decoding geometric Goppa codes using an extra place," *IEEE Trans. Inform. Theory,* vol. 38, pp. 1663-1676, 1992.

[20] J. Rifa and J. Borrell, "A fast algorithm to compute irreducible and primitive polynomials in finite fields," *Math. Systems Theory*, vol. 28, 1995, pp. 13-20.

[21] R. Roth and G. Ruckenstein, "Efficient decoding of Reed-Solomon codes beyond half the minimum distance," *IEEE Trans. Inform. Theory*, vol. 46, no. 1, January 2000, pp. 246-257.

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

[23] V. Shoup, "Fast construction of irreducible polynomials over finite fields," *J. Symb. Comp.*, vol. 17, 1994, pp.371-391.

[24] I.E. Shparlinski, "Finding irreducible and primitive polynomials," *Appl. Alg. Engin. Commun. Comp.* vol. 4, pp. 263-268, 1993.

[25] I.E. Shparlinski, *Finite Fields: Theory and Computation,* Mathematics and its Applications vol 477, Kluwer Acad. Publ., Dordrecht 1999.

[26] M. Sudan, "Decoding of Reed Solomon codes beyond the error-correction bound," *Journal of Complexity*, 13, 1997, pp. 180-193.

[27] M. Sudan, L. Trevisan, and S. Vadhan, "Pseudorandom generators without the XOR lemma," *Proceedings of the 31st Annual ACM Symposium on Theory of Computing*, pp.537-546, 1999.

[28] V.K. Wei, "Generalized Hamming weights for linear codes," *IEEE Trans. Inform. Theory*, vol. IT-37, pp. 1412-1418, Sept. 1991.

[29] J.M. Wozencraft, "List decoding," *Quarterly Progress Report*, vol. 48, Research Laboratory of Electronics, MIT, Jan. 15, 1958, 90–95.

[30] Xin-Wen Wu, "An Algorithm for Finding the Roots of the Polynomials over Order Domains," in *Proc. of 2002 IEEE International Symposium on Information Theory*, Lausanne, Switzerland, June 2002.

[31] Xin-Wen Wu and Paul H. Siegel, "Efficient Root-Finding Algorithm with Applications to List Decoding of Algebraic-Geometric Codes," *IEEE Trans. Inform. Theory*, vol. 47, no. 6, Sept. 2001, pp. 2579-2587.