Applications of the BMS Algorithm to decoding of algebraic codes

Shojiro Sakata
The University of Electro-Communications
Department of Information and Communication Engineering
Chofu-shi, Tokyo 182-8585, JAPAN

Abstract

I will discuss several applications of the BMS algorithm [1][2][3][4][5][6] to decoding of algebraic codes: (1) syndrome decoding of codes from curves up to the Feng-Rao bound by using the BMS algorithm with majority logic; (2) list decoding of RS codes and codes and Hermitian codes. The computational complexities of our methods are much less than the other decoding methods including Feng-Rao algorithm simply based on Gaussian elimination. These reductions in computational complexity are based on the special structures or properties of the given input data (syndrome arrays, etc.) which are used cleverly by the BMS algorithm. Furthermore, we will show that multi-variable interpolation problem can be solved by the BMS algorithm.

1 Introduction

We treat *algebraic codes* which are the most important class of error-correcting codes from both practical and theoretical viewpoints. They are a subclass of so-called *linear codes* which are defined as linear subspaces of the vector space $\mathbf{F}_q^n$ over a finite field $\mathbf{F}_q$. In the digital communication system, the transmitter sends via the noisy transmission channel one of the vectors (called *codewords*) which belong to a certain code. After the receiver gets a *received word*, which probably might be different from the *sent codeword* because of disturbances, he recovers (*decodes*) from it the sent codeword and obtain the information. A code which is capable of correcting t or less errors among the sent n symbols composing the codeword is called *t-error-correcting code*. For any two vectors $\underline{a} = (a_i)_{1\le i\le n}, \underline{b} = (b_i)_{1\le i\le n} \in \mathbf{F}_q^n$, their *Hamming distance* is defined as $d_H(\underline{a},\underline{b}) := \#\{i \mid a_i \neq b_i\}$, and further the *minimum distance* of a code C ($\subset \mathbf{F}_q^n$) is defined as $d(C) := \min\{d_H(\underline{a},\underline{b}) \mid \underline{a},\underline{b} \in C,\ \underline{a} \neq \underline{b}\}$, where $t = \lfloor \frac{d(C)-1}{2} \rfloor$ is the number of correctable errors. The value $d(C)$ or t is an important parameter representing the correction performance of the code.

Decoding, which is to recover the sent codeword as above-mentioned, is a kind of algebraic computation procedure over the finite field $\mathbf{F}_q$, and it is given basically in form of algorithm. If the received word contains more errors than the number t of correctable errors, the decoding algorithm might output a wrong codeword which is different from the sent codeword. But, the error event can be viewed as a probabilistic phenomenon, and more errors can occur with less probability, which usually is negligibly small.

Before discussing how to decode algebraic codes, we give the definitions of some important algebraic codes, particularly Reed-Solomon (RS) codes and *one-point* codes from algebraic curves (alias *algebraic geometry codes* or AG codes) [7][8].

Nowadays one of the most practically used algebraic error-correcting codes are RS codes, which are defined via the following three algebraic sets ($n = q-1$, $0 < h < n$, and α is a primitive element of $\mathbf{F}_q$):

(1) *Information symbol set* $\mathcal{K} = \mathbf{F}_q$;

(2) *Symbol locator set* $\mathcal{P} = \{P_i(:= \alpha^{i-1}) \mid 1 \le i \le n\}$;

(3) $\mathcal{C} = \{f \in \mathbf{F}_q[x] \mid \deg(f) \le h-1\}$,

where $\mathcal{K}$ is the set of symbols carrying information with it, and $\mathcal{P}$ is the set of *locators* (or labels) P_j denoting the position or index j of each component symbol c_j ($\in \mathbf{F}_q$) of a codeword $\underline{c} = (c_j)_{1\le j\le n} \in C$. A RS code C ($\subset \mathbf{F}_q^n$) is composed of the vectors $\mathrm{ev}(f) := (f(P_1,\cdots,f(P_n))$ ($\in \mathbf{F}_q^n$) corresponding to a function f in the space $\mathcal{C}$. For the subspace C of $\mathbf{F}_q^n$,

its orthogonal complement (*null space*)

$$C^{\perp} := \{\underline{c} = (c_j) \in \mathbf{F}_q^n \mid \underline{c} \cdot \mathrm{ev}(f) (:= \sum_{1 \le j \le n} c_j f(P_j)) = 0\}$$

is also called RS code, where we remark that the inner product $\underline{c} \cdot \mathrm{ev}(f)$ of these two vectors is an element of $\mathbf{F}_q$. C and $C^{\perp}$ sometimes are called *primal* and *dual* RS codes, respectively.

One-point codes from algebraic curves, which have better performance and great potentialities in near future, are defined based on the following three algebraic sets:

(1) *Information symbol set* $\mathcal{K} = \mathbf{F}_q$;

(2) *Symbol locator set* $\mathcal{P} = \{P_i \mid 1 \le i \le n\}$: the set of $\mathbf{F}_q$-rational points on an algebraic curve $\mathcal{X}$ over $\mathbf{F}_q$;

(3) $\mathcal{C} = L(mP_\infty)$: the set of algebraic functions on the curve $\mathcal{X}$ which has a single pole at the infinity point P_∞ with *pole order* less than or equal to m.

Similarly to RS codes, we have primal and dual codes:

$$C := \{\underline{c} = \mathrm{ev}(f) (= (f(P_i))_{1 \le i \le n}) \mid f \in \mathcal{C}\},$$
$$C^{\perp} := \{\underline{c} \in \mathbf{F}_q^n \mid \underline{c} \cdot \mathrm{ev}(f) = 0,\ f \in \mathcal{C}\}.$$

As a special case, if we take as $\mathcal{X}$ the projective line over $\mathbf{F}_q$ containing the infinity point P_∞ as well, and let $\mathcal{P}$ be the set of all affine points on $\mathcal{X}$ or equivalently $\mathbf{F}_q$, then we have the *extended* RS code with length $n = q$. By deleting 0 from $\mathcal{P}$, we have the ordinary RS code of length $n = q - 1$.

Although we can take the defining curve $\mathcal{X}$ in the projective space of any dimension N, we restrict to a plane curve $\mathcal{X}$ (i.e. $N = 2$) or particularly the Hermitian curve over $\mathbf{F}_q$, where $q = q_1^2$:

$$\mathcal{X} : x^{q_1+1} = y^{q_1} + y.$$

We take as $\mathcal{P}$ all the $\mathbf{F}_q$-rational points on $\mathcal{X}$ excluding the infinity point P_∞. Letting $\Pi := \{\underline{j} = (j_1, j_2) \in \mathbf{Z}_0^2 \mid 0 \le j_2 \le q_1 - 1\}$, $\Pi(m) := \{\underline{j} = (j_1, j_2) \in \Pi \mid q_1 j_1 + (q_1 + 1) j_2 \le m\}$, and $\mathcal{C} = \langle \underline{x^j} = x^{j_1} y^{j_2} \mid \underline{j} = (j_1, j_2) \in \Pi(m) \rangle_{\mathbf{F}_q}$, we can have the primal code $C = C(m)$ and the dual code $C^{\perp} = C^{\perp}(m)$ with length $n := q_1^3$, whose dimensions and minimum distances are as follows in case of $n > m \ge 2g - 1$, where $g = \frac{q_1(q_1-1)}{2}$ is the genus of the curve $\mathcal{X}$:

$$k(C) = m - g + 1,\ d(C) \ge n - m;$$
$$k(C^{\perp}) = n - m + g - 1,\ d(C^{\perp}) \ge m - 2g + 2.$$

($d_G := m - 2g + 2$ is called *Goppa bound* of the dual code $C^{\perp}$). Actually, if $m + m' = q_1^3 + q_1^2 - q_1 - 2$, it is seen that the primal Hermitian code $C(m)$ and the dual Hermitian code $C^{\perp}(m')$ is equivalent.

2 Syndrome decoding of dual codes

First we will show that decoding of a dual RS code $C^{\perp}$ with minimum distance $d = h + 1$ is reduced to the problem of finding a polynomial in $\mathbf{F}_q[x]$ which is *valid* for a certain one-dimensional (1-D) array derived from the received word. Let $\underline{c} = (c_j)_{1 \le j \le n} \in C^{\perp}$ and $\underline{e} = (e_j)_{1 \le j \le n} \in \mathbf{F}_q^n$ be a sent codeword and an *error vector*, respectively. Then, the received word is $\underline{r} = \underline{c} + \underline{e} = (r_j)_{1 \le j \le n} \in \mathbf{F}_q^n$, where $r_j = c_j + e_j$, $1 \le j \le n$. We assume that the number of errors, or in other words the size of the set $\mathcal{E} := \{P_j \mid e_j \ne 0\}$ ($\subset \mathcal{P}$) of *error locators*, is $t' := \#\mathcal{E} \le t$, where t ($= \lfloor \frac{h}{2} \rfloor$) is the number of correctable errors. The receiver gets the received word $\underline{r} = (r_j)$, but he has no knowledge of both $\underline{c}$ and $\underline{e}$. How can he find either $\underline{c}$ or $\underline{e}$ from $\underline{r}$? Since no error, i.e. the case of $\underline{e} = 0$ is the most likely in actual channels, he begins with checking whether the received word $\underline{r}$ contains any error or not. For a dual RS code, it is very easy and he has only to check for some $f \in \mathcal{C}$ whether the inner product $\underline{r} \cdot \mathrm{ev}(f) = 0$ or not. More precisely, he calculates the syndromes $s_i := \underline{r} \cdot \mathrm{ev}(x^i)$ corresponding to the basis functions x^i, $0 \le i \le h - 1$ of the function space $\mathcal{C}$, and obtains the array $s = (s_i)_{0 \le i \le h-1}$. If $s = 0$, then he can suppose no error so that he does not need to decode. But, if $s \ne 0$, then he enters the procedure of decoding. A basic decoding method consists of two stages, finding the error locators, i.e. the unknown j_i or $\alpha^{j_i - 1}$, $1 \le i \le t'$ for $\mathcal{E} = \{\alpha^{j_i - 1} \mid 1 \le i \le t'\}$, and calculating the error values e_{j_i}, $1 \le i \le t'$. Provided the error locators $\mathcal{E}$ are found in the first stage, the second stage is easier and reduced to finding the unique solution e_{j_i}, $1 \le i \le t'$ of

the linear system of equations: $\sum_{1\le i\le t'} e_{j_i}\alpha^{(j_i-1)j} = s_j$, $0 \le j \le h-1$.

Now, our main concern is in the first stage. Assuming $t' \le t$ for $\mathcal{E} = \{\alpha^{j_i-1} \mid 1 \le i \le t'\}$, where t' and j_i, $1 \le i \le t'$ are unknown, we consider an infinite array $u = (u_j)$ defined by $u_j := \underline{e} \cdot \mathrm{ev}(x^j) = \sum_{1\le i\le t'} e_{j_i}\alpha^{(j_i-1)j}$, $j \in \mathbf{Z}_0$ instead of s, and further the ideal $I = I(u) := \{f \in \mathbf{F}_q[x] \mid f \circ u = 0\}$, which is called the *characteristic ideal* of u, as well as the zero manifold $V(I) := \{\gamma \in \mathbf{F}_q \mid f(\gamma) = 0, {}^\forall f \in I\}$ defined by it, where for $f = f(x) = \sum_{0\le l\le d} f_l x^l$, $v = f \circ u := (v_j)_{j\in\mathbf{Z}_0}$ is the array defined by $v_j := \sum_{0\le l\le d} f_l u_{l+j}$, $j \in \mathbf{Z}_0$. Actually, we have

Lemma 1 $\quad \mathcal{E} = V(I)$.

Proof: For $f = f(x) = \sum_{0\le l\le d} f_l x^l$, we have

$$\begin{aligned} & f(\alpha^{j_i-1}) = 0,\ 1 \le i \le t' \\ \Leftrightarrow \quad & \textstyle\sum_{0\le l\le d} f_l \alpha^{(j_i-1)l} = 0,\ 1 \le i \le t' \\ \Leftrightarrow \quad & \textstyle\sum_{1\le i\le t'} (\sum_{0\le l\le d} f_l \alpha^{(j_i-1)l}) e_{j_i}\alpha^{(j_i-1)j} \\ & \qquad = 0,\ {}^\forall j \in \mathbf{Z}_0 \\ \Leftrightarrow \quad & \textstyle\sum_{0\le l\le d} f_l \sum_{1\le i\le t'} e_{j_i}\alpha^{(j_i-1)(l+j)} = 0, \\ & \qquad {}^\forall j \in \mathbf{Z}_0, \end{aligned}$$

where

the last identity is equivalent to $\sum_{0\le l\le d} f_l u_{l+j} = 0$, ${}^\forall j \in \mathbf{Z}_0$, i.e. $f \circ u = 0$. By the way, the equivalence between the second and third identities comes from the fact that t' arrays $u^{(i)} := (u_j^{(i)})$, $1 \le i \le t'$ which are defined by $u_j^{(i)} := \alpha^{(j_i-1)j}$ are linearly independent of each other. ♡

Since we have that $s_i = \underline{r} \cdot \mathrm{ev}(x^i) = (\underline{c} + \underline{e}) \cdot \mathrm{ev}(x^i) = \underline{e} \cdot \mathrm{ev}(x^i)$, $0 \le i \le h-1$, the subarray $u^h := (u_j)_{0\le j\le h-1}$ of the above infinite array u coincides with the syndrome array $s = (s_j)_{0\le j\le h-1}$, although we cannot obtain the whole infinite array u. Particularly, the values u_j, $j \ge h$ sometimes are called *unknown syndromes*. However, if $\deg(f) = t' \le t$, in view of $h-1-t' \ge t'-1$, for $1 \le i \le t'$, we have t' finite arrays $u_j^{(i)} := \alpha^{(j_i-1)j}$, $0 \le j \le h-1-t'$, which also are linearly independent of each other. Consequently, we have: $\mathcal{E} = V(f) \Leftrightarrow$

$$\sum_{0\le l\le t'} f_l u_{l+j} = 0,\ 0 \le j \le h-1-t', \qquad (1)$$

which implies that we can find the error locators $\mathcal{E}$ as the roots of a polynomial f which is valid for the *known syndromes* $u_i(= s_i)$, $0 \le i \le h-1$ obtained from the received word $\underline{r}$ and has the minimum degree, provided that the actual number t' of errors contained in $\underline{r}$ does not exceed the number t of correctable errors.

As we have seen, the problem of decoding dual RS codes is reduced to finding a valid polynomial for a certain finite (1-D) array. Naturally this fact can be extended to the problem of decoding more general codes including *codes from algebraic curves*. Particularly, in the multidimensional case, it also implies that we must find a Gröbner basis of the characteristic ideal of the array. Below we will show that the decoding of a dual Hermitian code $C^\perp$ is reduced to the problem of finding a minimal polynomial set ($\in \mathbf{F}_q[x,y]$) of a certain 2-D array derived from a received word.

Let $\underline{c} = (c_j)(\in C^\perp)$, $\underline{e} = (e_j)(\in \mathbf{F}_q^n)$, $\underline{r} = \underline{c} + \underline{e} = (v_j)(\in \mathbf{F}_q^n)$ be the sent codeword, the error vector, and the received word, respectively. We assume that the size of the error locators $\mathcal{E} := \{P_j \mid e_j \ne 0\} = \{P_{l_i} \mid 1 \le i \le t'\}(\subset \mathcal{P})$ is $t' := \#\mathcal{E} \le t := \lfloor \frac{d_G-1}{2} \rfloor$. As each point of the curve can be represented as $P_l = (\alpha_l, \beta_l)(\in \mathbf{F}_q^2)$, the syndrome $s = (s_{\underline{j}})$, $\underline{j} \in \Pi(m)$ obtained by $s_{\underline{j}} := \underline{r} \cdot \mathrm{ev}(\underline{x}^{\underline{j}})$ from the received word $\underline{r}$ is a finite subarray of the infinite 2-D array $u = (u_{\underline{j}})$, $\underline{j} \in \mathbf{Z}_0^2$, defined by

$$\begin{aligned} u_{\underline{j}} \quad &:= \quad \underline{e} \cdot \mathrm{ev}(\underline{x}^{\underline{j}}) = \sum_{1\le i\le t'} e_{l_i} \alpha_{l_i}^{j_1} \beta_{l_i}^{j_2}, \\ & \qquad \underline{j} = (j_1, j_2) \in \mathbf{Z}_0^2, \end{aligned}$$

which we call *error locator array*. About the *characteristic ideal* (*submodule*) $I = I(u) := \{f \in \mathbf{F}_q[\Pi] \mid f \circ u = 0\}$ of a 2-D array $u = (u_{\underline{j}})$, $\underline{j} \in \mathbf{Z}_0^2$ and its zero manifold $V(I) := \{P \in \mathcal{P} \mid f(P) = 0, {}^\forall f \in I\}$, we have the following lemma similar to Lemma 1. Thus, we call I also the *error locator ideal* (or *submodule*), and sometimes denote it as $\tilde{I}_e$ (or $\tilde{M}_e$).

Lemma 2 $\quad \mathcal{E} = V(I)$.

Proof: For $f = f(x,y) = \sum_{\underline{i}\in \mathrm{Supp}(f)} f_{\underline{i}} \underline{x}^{\underline{l}}$, we have

$$\begin{aligned} & f(\alpha_{l_k}, \beta_{l_k}) = 0,\ 1 \le k \le t' \\ \Leftrightarrow \quad & \textstyle\sum_{\underline{i}=(i_1,i_2)\in \mathrm{Supp}(f)} f_{\underline{i}} \alpha_{l_k}^{i_1} \beta_{l_k}^{i_2} = 0,\ 1 \le k \le t' \\ \Leftrightarrow \quad & \textstyle\sum_{1\le k\le t'} (\sum_{\underline{i}\in \mathrm{Supp}(f)} f_{\underline{i}} \alpha_{l_k}^{i_1} \beta_{l_k}^{i_2}) e_{l_k} \alpha_{l_k}^{j_1} \beta_{l_k}^{j_2} \\ & \qquad = 0,\ {}^\forall \underline{j} = (j_1, j_2) \in \mathbf{Z}_0^2 \end{aligned}$$

$$\Leftrightarrow \quad \sum_{\underline{i}\in \mathrm{Supp}(f)} f_{\underline{i}} \sum_{1\le k\le t'} e_{l_k} \alpha_{l_k}^{i_1+j_1} \beta_{l_k}^{i_2+j_2} = 0, \quad {}^{\forall}\underline{j} = (j_1, j_2) \in \mathbf{Z}_0^2,$$

where the last identity is equivalent to $\sum_{\underline{i}\in \mathrm{Supp}(f)} f_{\underline{i}} u_{\underline{i}+\underline{j}} = 0$, ${}^{\forall}\underline{j} \in \mathbf{Z}_0^2$, i.e. $f \circ u = 0$. The equivalence between the second and third identities comes from the fact that t' arrays $u^{(l)} := (u_{\underline{j}}^{(l)})$, $1 \le l \le t'$ deflned by $u_{\underline{j}}^{(l)} := \alpha_l^{j_1} \beta_l^{j_2}$, $\underline{j} \in \mathbf{Z}_0^2$, are linearly independent from each other. ♡

In the above, for the ring $R := \mathbf{F}_q[x, y]$, the function space $\mathbf{F}_q[\Pi] := \langle \underline{x}^{\underline{j}} = x^{j_1} y^{j_2} \mid \underline{j} = (j_1, j_2) \in \Pi \rangle_{\mathbf{F}_q}$ is a R-submodule, which coincides with the R-module $\mathbf{F}_q[\mathbf{Z}_0^2]$ $(:= \langle \underline{x}^{\underline{j}} = x^{j_1} y^{j_2} \mid \underline{j} = (j_1, j_2) \in \mathbf{Z}_0^2 \rangle_{\mathbf{F}_q})$ modulo the submodule $M_{\mathcal{X}}$ $(:= \langle x^{q_1+1} - y^{q_1} - y \rangle_{\mathbf{F}_q})$. The known syndromes $s_{\underline{j}} = \underline{r} \cdot \mathrm{ev}(\underline{x}^{\underline{j}})$, $\underline{j} \in \Pi(m)$, which are obtained from the received word, are identical with the subarray $u_{\underline{j}}$, $\underline{j} \in \Pi(m)$, but the part $u_{\underline{j}}$, $\underline{j} \in \Pi \setminus \Pi(m)$ are unknown syndromes. On the other hand, among the functions defined on the curve, since $\underline{x}^{\underline{j}}$, $\underline{j} \in \mathbf{Z}_0^2 \setminus \Pi$ are linearly dependent on $\{\underline{x}^{\underline{j}} \mid \underline{j} \in \Pi\}$, the subarray $u_{\underline{j}}$, $\underline{j} \in 2\Pi(m)$ also is known, where $2\Pi(m) := \{\underline{i}+\underline{j} \mid \underline{i}, \underline{j} \in \Pi(m)\}$. In the linear recurrence $f \circ u = 0$, i.e.

$$\sum_{\underline{i}\in \mathrm{Supp}(f)} f_{\underline{i}} u_{\underline{i}+\underline{j}} = 0, \ \underline{j} \in \Pi,$$

not only the components $u_{\underline{j}}$, $\underline{j} \in \Pi(m)$ but also the components $u_{\underline{j}}$, $\underline{j} \in 2\Pi(m) \setminus \Pi(m)$ are concerned. Therefore, all the components $u_{\underline{j}}$, $\underline{j} \in 2\Pi(m)$ are necessary for decoding by using the BMS algorithm. Furthermore, treating only the known syndrome is not enough for decoding of this kind of codes up to half of the designed distance, which we will discuss below.

There have been several investigations on *designed distances* or *lower bounds for minimum distances* of codes from curves. We consider the Feng-Rao bound of dual Hermitian codes, which is equal to the so-called order bound as well as to the Goppa bound d_G in case of $n > m \ge 2g-1$ for these codes. Although the Feng-Rao decoding algorithm based on Gaussian elimination and majority logic can decode up to $t_G := \lfloor \frac{d_G-1}{2} \rfloor$ errors, it will turn out that the BMS algorithm with majority logic can do the same more efficiently [3]. By using the BMS algorithm we can determine the unknown syndromes based on majority logic in its unique (basically, similar to the Feng-Rao algorithm) fashion.

Let $\mathcal{O}$ be the set of pole orders $o(f)$ of functions f on the algebraic curve $\mathcal{X}$ over the closed extension (closure) $\tilde{\mathbf{F}}_{q_1} := \cup_{i\ge 1} \mathbf{F}_{q_1^i}$ of $\mathbf{F}_{q_1}$, and $\mathcal{O}(m) := \{l \in \mathcal{O} \mid l \le m\}$. Particularly, we denote the pole order $o(\underline{x}^{\underline{i}})$ of the coordinate function $\underline{x}^{\underline{i}}$ simply as $o(\underline{i})$, $\underline{i} \in \mathbf{Z}_0^2$. Then, via $o(\underline{i})$, $\underline{i} \in \mathbf{Z}_0^2$, $\mathcal{O}$ and $\mathcal{O}(m)$ one-to-one correspond to Π and $\Pi(m)$, respectively. For $l \in \mathcal{O}$,

$$\nu(l) := \#\{(i, j) \in \mathcal{O}^2 \mid i + j = l\}$$

is introduced and the order bound of the code $C^{\perp}(m)$ is defined as

$$d(m) := \min\{\nu(l) \mid l \ge m + 1\}.$$

On the other hand we sometimes have a couple of points $\underline{r} \in \Pi$ and $\underline{r}' \in 2\Pi \setminus \Pi$ s.t. $o(\underline{r}) = o(\underline{r}')$ (thus, $\underline{x}^{\underline{r}} = \underline{x}^{\underline{r}'} \bmod \mathbf{F}_q[\Pi]$), which are called *conjugate* to each other. In our terminology, we have that if $o(\underline{r}) = o(\underline{r}') = l \in \mathcal{O}$,

$$\nu(l) = \#(\Gamma_{\underline{r}} \cap \Gamma_{\underline{r}'}) \cap \Pi,$$

where if such a couple does not exist, $\Gamma_{\underline{r}} \cap \Gamma_{\underline{r}'}$ should be regarded simply as $\Gamma_{\underline{r}}$ for $\underline{r}$ s.t. $o(\underline{r}) = l$.

As we show below, in case of t_G or less errors, we can find iteratively at each $\underline{j} \in 2\Pi \setminus 2\Pi(m)$ the value of the unknown syndrome $u_{\underline{j}}$ and update a pair of minimal polynomial set F and auxiliary polynomial set G by using the modified BMS algorithm with majority voting among the candidate syndrome values, where a pair of conjugate points are treated simultaneously at each BMS iteration, i.e. F and G are updated at each pole order l s.t. $o(\underline{x}^{\underline{r}}) = o(\underline{x}^{\underline{r}'}) = l$. Thus, we consider the syndrome subarray $u^l := u^{\underline{r}}$ s.t. $o(\underline{x}^{\underline{r}}) = o(\underline{x}^{\underline{r}'}) = l$ for each $l > m$. First we remark that $\nu(l) > 2t_G$, $l \ge m + 1$. From the known syndromes, we can get a minimal polynomial set F for the subarray u^m. Now, assume that we have got already the syndrome subarray u^l for some $l \ge m$ together with F and G of u^l, which is accompanied with the stable subsets Σ_F, Δ_F, and Δ_G. We stipulate the following number as the *total number of votes* at l: $v(l) :=$

$$\#((\Gamma_{\underline{r}} \bigcup \Gamma_{\underline{r}'}) \bigcap \Pi \bigcap \Sigma_F) \setminus ((\underline{r} - \Delta_G) \bigcup (\underline{r}' - \Delta_G)),$$

where $\underline{r} - \Delta_G := \{\underline{r} - \underline{j} \in \Pi \mid \underline{j} \in \Delta_G\}$. Furthermore, for a subset $\bar{F} \subset F$ at l, we stipulate the following

number as the *number of votes for* $\bar{F}$ or *for the candidate values of the unknown syndromes determined by using* $f \in \bar{F}$ at l: $v(\bar{F}) :=$

$$\#((\Gamma_{\underline{r}} \bigcup \Gamma_{\underline{r}'}) \bigcap \Pi \bigcap \Sigma_{\bar{F}}) \setminus ((\underline{r} - \Delta_G) \bigcup (\underline{r}' - \Delta_G)).$$

From the nature of iteration of BMS algorithm, we have the following:

Lemma 3 *If we have a minimal polynomial set* $F^{\oplus}$ *of* u^{l+1} *by updating* F *at the iteration at* l*, the difference* $\#\Delta_{F^{\oplus}} - \#\Delta_F$ *is identical with the number of votes for* $F_N := \{f \in F \mid f[u]_{\underline{r}} \neq 0 \vee f[u]_{\underline{r}'} \neq 0\}$ *for the pair of conjugate points* $\underline{r}$ *and* $\underline{r}'$ *at* l*.*

Then, we have the following conclusion, which assures the validity of the BMS algorithm with majority voting for finding the correct values of the unknown syndrome in case of correctable number of errors.

Theorem 1 *Provided the number of errors is* $t' \leq t$*, the polynomials* f *in* F *which give the correct syndrome values* $u^{\underline{r}}$ *or* $u^{\underline{r}'}$ *have the majority of votes among* F*.*

Proof: It is shown that $\#((\underline{r} - \Delta_G) \bigcup (\underline{r}' - \Delta_G)) \bigcap \Pi = \#\Delta_G$, and thus if the subset F_N of f which does not give the true syndrome values $u_{\underline{r}}$ or $u_{\underline{r}'}$ at l has the majority of votes, in view of Lemma 3 and $\#\Delta_F \setminus \Delta = \#\Delta_G$, we should have $\#\Delta_{F^{\oplus}} \setminus \Delta > \#\Delta_F \setminus \Delta + \frac{1}{2}v(l) = \#\Delta_F \setminus \Delta + \frac{1}{2}(2t - \#\Delta_F \setminus \Delta - \#\Delta_G) = t$, which contradicts the fact that for the eventual minimal polynomial set $\tilde{F}$ and auxiliary polynomial set $\tilde{G}$, we have $\#\Delta_{\tilde{F}} \setminus \Delta (= \#\Delta_{\tilde{G}}) = t'$, where $t' = \#\mathcal{E}$ for the zero manifold $V(\tilde{M}_e) = \mathcal{E}$ of the error locator submodule $\tilde{M}_e$. ♡

Our syndrome decoding method for Hermitian codes of codelength n has computational complexity $\mathcal{O}(n^{\frac{7}{3}})$ compared with $\mathcal{O}(n^3)$ of the method based on Gaussian elimination.

3 Multivariate polynomial interpolation and list decoding of algebraic codes

A univariate polynomial interpolation is given by the well-known *Lagrange interpolating polynomial*, i.e. given a set of M points $\{(x_l, y_l) \in \mathbf{F}_q^2 \mid 1 \leq l \leq M\}$ in the 2-D space $\mathbf{F}_q^2$, a polynomial with minimum degree satisfying the interpolation condition $f(x_l) = y_l$, $1 \leq l \leq M$ is

$$f(x) = \sum_{l=1}^{M} y_l \frac{\prod_{k \neq l}(x - x_k)}{\prod_{k \neq l}(x_l - x_k)},$$

where $x_k \neq x_l$, $k \neq l$, $1 \leq k, l \leq M$. We can consider any field, provided that exact computation without numerical errors is assumed. However, we restrict to the finite field $\mathbf{F}_q$ with sufficiently large q to concern ourselves in decoding of algebraic codes and to make our discussions simpler.

In the general case of multivariate interpolation, we cannot have such an explicit interpolating polynomial as above. This is the following problem. Given a set of M points $\{(\underline{x}^{(l)}, y^{(l)}) \in \mathbf{F}_q^N \times \mathbf{F}_q \mid 1 \leq l \leq M\}$ in the $N+1$-dimensional space $\mathbf{F}_q^{N+1}$ over $\mathbf{F}_q$, we want to find a N-variate polynomial f, which is *simplest* in some sense, satisfying the following condition:

$$f(\underline{x}^{(l)}) = y^{(l)},\ 1 \leq l \leq M \tag{2}$$

where $\underline{x}^{(l)} = (x_1^{(l)}, \cdots, x_N^{(l)}) \in \mathbf{F}_q^N$, $1 \leq l \leq M$, and we assume $\underline{x}^{(k)} \neq \underline{x}^{(l)}$, $k \neq l$, $1 \leq k, l \leq M$. Since this is a system of linear equations for the unknown coefficients of f, its solution is not always unique (if exists), which is given as a sum of a (special) solution of (2) and a general solution f of the following homogeneous system which is derived from (2) by putting $y^{(l)} = 0$, $1 \leq l \leq M$:

$$f(\underline{x}^{(l)}) = 0,\ \underline{x}^{(l)} \in V, \tag{3}$$

where $V := \{\underline{x}^{(l)} \mid 1 \leq l \leq M\} \subset \mathbf{F}_q^N$. The set of solutions f of (3)

$$I(V) := \{f \in \mathbf{F}_q[\underline{x}] \mid f(\underline{x}^{(l)}) = 0,\ \underline{x}^{(l)} \in V\}$$

is an ideal of the ring $\mathbf{F}_q[\underline{x}]$. Thus, provided that 'simplicity' is interpreted as 'minimality' as in Gröbner basis theory, the interpolation problem (2) can be divided into two subproblems, i.e. finding a Gröbner basis of the ideal corresponding to the homogeneous system (3) and obtaining a special (*minimal*) solution of the non-homogeneous system (2).

Now, for the arrays $u^{(l)} = (u_{\underline{j}}^{(l)})$, $v^{(l)} = (v_{\underline{j}}^{(l)})$, $\underline{j} \in \mathbf{Z}_0^N$, $1 \leq l \leq M$ and $u = (u_{\underline{j}})$, $v = (v_{\underline{j}})$, $\underline{j} \in \mathbf{Z}_0^N$ defined

by

$$u_{\underline{j}}^{(l)} := (\underline{x}^{(l)})^{\underline{j}},\ v_{\underline{j}}^{(l)} := y^{(l)}(\underline{x}^{(l)})^{\underline{j}},\ \underline{j} \in \mathbf{Z}_0^N, 1 \le l \le M;$$
$$u_{\underline{j}} := \textstyle\sum_{1\le l\le M} u_{\underline{j}}^{(l)},\ v_{\underline{j}} := \sum_{1\le l\le M} v_{\underline{j}}^{(l)},\ \underline{j} \in \mathbf{Z}_0^N,$$

it holds that

Lemma 4 *$f = \sum_{\underline{i}\in \mathrm{Supp}(f)} f_{\underline{i}} \underline{x}^{\underline{i}}$ satisfies the interpolation condition (2) iff $f \circ u = v$, i.e.*

$$\sum_{\underline{i}\in \mathrm{Supp}(f)} f_{\underline{i}} u_{\underline{i}+\underline{j}} = v_{\underline{j}},\ \underline{j} \in \mathbf{Z}_0^N \tag{4}$$

Proof:

$$\begin{aligned}
&\textstyle\sum_{\underline{i}\in \mathrm{Supp}(f)} f_{\underline{i}} (\underline{x}^{(l)})^{\underline{i}} = y^{(l)},\ 1 \le l \le M \\
\Leftrightarrow\quad &\textstyle\sum_{\underline{i}\in \mathrm{Supp}(f)} f_{\underline{i}} (\underline{x}^{(l)})^{\underline{i}+\underline{j}} = y^{(l)}(\underline{x}^{(l)})^{\underline{j}},\ \underline{j} \in \mathbf{Z}_0^N, \\
&\qquad\qquad 1 \le l \le M \\
\Leftrightarrow\quad &\textstyle\sum_{\underline{i}\in \mathrm{Supp}(f)} f_{\underline{i}} u_{\underline{i}+\underline{j}}^{(l)} = v_{\underline{j}}^{(l)},\ \underline{j} \in \mathbf{Z}_0^N,\ 1 \le l \le M \\
\Leftrightarrow\quad &\textstyle\sum_{\underline{i}\in \mathrm{Supp}(f)} f_{\underline{i}} u_{\underline{i}+\underline{j}} = v_{\underline{j}},\ \underline{j} \in \mathbf{Z}_0^N,
\end{aligned}$$

where the equivalence between the third and fourth conditions comes from the linear independence of the arrays $u^{(l)}$, $1 \le l \le M$ (Remark: we assume that q is sufficiently large). ♡

The linear recurrence corresponding to the homogeneous system (3) is just the homogeneous linear recurrence which is derived from (4) by letting the right-hand array $v := 0$, and it is easy to see that the characteristic ideal $I(u)$ of the left-hand array u is identical with $I(V)$.

Such a multivariate interpolation problem as above appears in the context of *list decoding* [9][10][11], which is a generalization of conventional *bounded-distance decoding* (including syndrome decoding) of algebraic codes. First, we give a simple sketch of list decoding of (primal) RS codes. We take a primal RS code $C = \{\underline{c} = (f(\alpha^{j-1}))_{1\le j\le n} \mid f \in \mathbf{F}_q[x], \deg(f) \le k-1\}$ of length $n = q-1$ and an integer $\tau (< n)$ which is more than the number of correctable errors $t = \lfloor \frac{n-k}{2} \rfloor$. Given a received word $\underline{r} = (r_j)_{1\le j\le n} \in \mathbf{F}_q^n$, we want to find all the codewords $\underline{c} = (c_j)_{1\le j\le n} \in C$ whose components differ from $\underline{r}$ by at most τ components, i.e. for $\underline{r} = \underline{c} + \underline{e}$ with $\underline{e} = (e_j)_{1\le j\le n} \in \mathbf{F}_q^n$, we assume that the size $t' := \#\mathcal{E}$ of the error locators $\mathcal{E} = \{\alpha^{j-1} \mid e_j \ne 0, 1 \le j \le n\}$ is less than or equal to τ. Then, it is shown below that list decoding is reduced to an interpolation problem, where the *degree* $\deg(Q) (\in \mathbf{Z}_0^2)$ of a bi-variate polynomial Q is introduced according to the total order $\le_T$ defined by the weight $\underline{w} = (1, k-1)$ (and the lexical order $\le_L$ s.t. $x <_L y$).

Lemma 5 *Assume that a nonzero bi-variate polynomial $Q(x,y) = \sum_{(i,j)\in \mathrm{Supp}(Q)} Q_{ij} x^i y^i$ $(\in \mathbf{F}_q[x,y])$ satisfies the condition*

$$Q(\alpha^{j-1}, r_j) = 0,\ 1 \le j \le n \tag{5}$$

and that it has $\deg(Q) <_T (n-\tau, 0)$. Then, a polynomial f corresponding to a codeword satisfies $y - f(x) \mid Q(x,y)$.

Therefore, by finding $Q(x,y)$ satisfying the interpolation condition (5) and furthermore finding its factors in the form of $y - f(x)$, we can obtain f which gives a candidate codeword. The 2-D linear recurrence derived from (5) is a special case of the homogeneous linear recurrence (4), where the right-hand side is 0. As a conclusion, we can obtain Q among a Gröbner basis of the characteristic ideal of the 2-D array u defined by $\underline{x}^{(j)} = (\alpha^{j-1}, r_j)$, $1 \le j \le n$. Our method of finding the interpolation polynomial for list decoding of RS codes of codelength n and coding rate $\frac{k}{n} = R$ has computational complexity $\mathcal{O}(R^{-\frac{1}{2}} n^2)$ compared with $\mathcal{O}(n^3)$ of the method based on Gaussian elimination.

We do not discuss the existence condition of such an interpolation polynomial as above, although it is related with a practically important problem of how much list decoding can contribute to improvement of reliability in transmission. If exists, it is the most convenient to have an interpolation polynomial Q with minimal degree.

List decoding of codes from curves also is reduced to an interpolation problem. For simplicity, we consider only primal Hermitian codes $C := \{\underline{c} = (f(P_j))_{1\le j\le n} \mid f \in L(mP_\infty) (= \mathbf{F}_q[\Pi(m)])\}$. In this case, the *degree* of a tri-variate polynomial $Q(x,y,z)$ with support $\mathrm{Supp}(Q)$ $(\subset \Pi(m) \times \mathbf{Z}_0)$ is introduced over $\Pi(m) \times \mathbf{Z}_0$ according to the total order $\le_T$ defined by the weight $\underline{w} = (q_1, q_1+1, m)$ (and the lexical order $<_L$ s.t. $x <_L y <_L z$). Then, we have:

Lemma 6 *We assume that a nonzero polynomial (or rather function)* $Q(P,z) = Q(x,y,z) = \sum_{(\underline{i},l)\in \mathrm{Supp}(Q)} q_{\underline{i},l}\underline{x}^{\underline{i}}z^l$ $(\in \mathbf{F}_q[\Pi(m)][z])$ *satisfies the condition*

$$Q(P_j, r_j) = 0,\ 1 \le j \le n \tag{6}$$

and has degree $\deg(Q) <_T (n-\tau, 0, 0)$*, where the components of* $\underline{x} = (x,y)$ *are viewed not only as the coordinates of* P *but also as functions on the curve* $\mathcal{X}$*. Then, a function* $f(x,y) \in \Pi(m)$ *corresponding to a codeword satisfies* $z - f(x,y) \mid Q(x,y,z)$*.*

Also in this situation, the interpolation condition (6) is reduced to a homogeneous linear recurrence. Consequently, we can obtain Q among a Gröbner basis of the characteristic ideal of a 3-D array defined by $\underline{x}^{(j)} = (P_j, r_j)$, $1 \le j \le n$.

From the viewpoint of linear algebra, the linear recurrence (4) is nothing but a system of linear equations for unknowns $f_{\underline{i}}$, $\underline{i} \in \mathrm{Supp}(f)$. Particularly, in the 2-D case, it is just a 2-D block-Hankel or 2-D block-Toeplitz system of linear equations, where the extent $\mathrm{Supp}(f)$ of a solution f is also unknown in our situation, distinctly from solving the ordinary system of linear equations. For the purposes of multivariate interpolation or decoding of codes, our method is unique and different from the known fast methods of solving block-Hankel systems or other interpolation methods.

Soon after Sudan [9] proposed list decoding method, Guruswami and Sudan [11] gave its improvement called *GS list decoding* method, which can be effective even for higher coding rate, while the original Sudan list decoding works only for coding rate $\le \frac{1}{3}$. It is based on the notion of *zeros with multiplicity* defined as follows. A point $\underline{x}^{(l)} = (x^{(l)}, y^{(l)}) \in \mathbf{F}_q^2$ is called a *zero with multiplicity* s *or more* of $Q(x,y) = \sum_{(i,j)\in \mathrm{Supp}(Q)} Q_{ij}x^i y^i = \sum_{\underline{k}\in \mathrm{Supp}(Q)} Q_{\underline{k}}\underline{x}^{\underline{k}} \in \mathbf{F}_q[x,y]$ iff in the expansion

$$\tilde{Q}^{(l)}(x,y) = \sum_{\underline{k}\in \mathbf{Z}_0^2} \tilde{Q}^{(l)}_{\underline{k}}\underline{x}^{\underline{k}} \tag{7}$$

of the polynomial $\tilde{Q}^{(l)}(x,y) := Q(x + x^{(l)}, y + y^{(l)})$, all the terms $\tilde{Q}^{(l)}_{\underline{k}}\underline{x}^{\underline{k}}$ vanish, i.e. $\tilde{Q}^{(l)}_{\underline{k}} = 0$, for $^\forall\underline{k} = (k_1, k_2) \in \mathbf{Z}_0^2$ s.t. $k_1 + k_2 < s$. Then, we have a modification of Lemma 5

Lemma 7 *Assume that a nonzero bi-variate polynomial* $Q(x,y) = \sum_{(i,j)\in \mathrm{Supp}(Q)} Q_{ij}x^i y^i$ $(\in \mathbf{F}_q[x,y])$ *has zeros* (α^{j-1}, r_j), $1 \le j \le n$*, each with multiplicity* s *or more and that it has* $\deg(Q) <_T (s(n-\tau), 0)$*. Then, a polynomial* f *corresponding to a codeword satisfies* $y - f(x) \mid Q(x,y)$*.*

We do not discuss the error correction performance of GS list decoding. We will show that one can apply the BMS algorithm to find such an interpolation polynomial with minimal degree. Generalizing the notion of zero with multiplicity only a bit, we consider a subset $\Lambda \subset \mathbf{Z}_0^2$, which sometimes is called a *stable* set, like delta set Δ s.t. if $\underline{i} \in \Lambda$, then any $\underline{j} \in \mathbf{Z}_0^2$ s.t. $\underline{j} \le_P \underline{i}$ satisfies $\underline{j} \in \Lambda$. In this context we call such a subset Λ a *multiplicity region*, and we have

Lemma 8 *For a finite subset* $V = \{\underline{x}^{(l)} \mid 1 \le l \le n\} \subset \mathbf{F}_q^2$ *and a certain multiplicity region* $\Lambda \subset \mathbf{Z}_0^2$*, the set* $I(V;\Lambda) :=$

$$\{Q(x,y) \in \mathbf{F}_q[x,y] \mid \tilde{Q}^{(l)}_{\underline{k}} = 0,\ \underline{k} \in \Lambda,\ 1 \le l \le n\}$$

is an ideal of $\mathbf{F}_q^2$*, which we call the ideal of the zero manifold* V *with multiplicity region* Λ*.*

Next, for two points $\underline{p} = (p_1, p_2)$, $\underline{q} = (q_1, q_2)$ $(\in \mathbf{Z}_0^2)$ we introduce the 2-D binomial coefficients

$$\binom{\underline{p}}{\underline{q}} := \binom{p_1}{q_1}\binom{p_2}{q_2},$$

where if it does not hold that $\underline{p} \ge_P \underline{q}$, $\binom{\underline{p}}{\underline{q}} = 0$. Then, the coefficients $\tilde{Q}^{(l)}_{\underline{k}}$ of the expansion of (7) are written as

$$\tilde{Q}^{(l)}_{\underline{k}} = \sum_{\underline{j}\ge_P \underline{k},\ \underline{j}\in \mathrm{Supp}(Q)} \binom{\underline{j}}{\underline{k}} Q_{\underline{j}}(\underline{x}^{(l)})^{\underline{j}-\underline{k}}.$$

Therefore,

Lemma 9 $Q = \sum_{\underline{k}\in \mathrm{Supp}(Q)} Q_{\underline{k}}\underline{x}^{\underline{k}} \in I(V,\Lambda) \Leftrightarrow$

$$\sum_{\underline{j}\ge_P \underline{k},\ \underline{j}\in \mathrm{Supp}(Q)} \binom{\underline{j}}{\underline{k}} Q_{\underline{j}}(\underline{x}^{(l)})^{\underline{j}-\underline{k}} = 0,\ \underline{k} \in \Lambda, 1 \le l \le n.$$

In case of $\Lambda = \Gamma_{\underline{m}}$ for a point $\underline{m} \in \mathbf{Z}_0^2$, we introduce a 2-D array $u = (u_{\underline{j}})$ as follows:

$$u_{\underline{j}} := \sum_{1\le l\le n} \binom{\underline{j}}{\underline{m}} (\underline{x}^{(l)})^{\underline{j}-\underline{m}},\ \underline{j} \in \mathbf{Z}_0^2.$$

Then,

Lemma 10 $Q = \sum_{\underline{k} \in \mathrm{Supp}(Q)} Q_{\underline{k}} \underline{x}^{\underline{k}} \in I(V, \Lambda)$ $\Leftrightarrow Q \circ u = 0$, *i.e.*

$$\sum_{\underline{k} \in \mathrm{Supp}(Q)} Q_{\underline{k}} u_{\underline{k}+\underline{j}} = 0,\ \underline{j} \in \mathbf{Z}_0^2.$$

More generally, in case of $\Lambda = \cup_{1 \le i \le L} \Gamma_{\underline{m}^{[i]}}$ for $\underline{m}^{[i]} \in \mathbf{Z}_0^2$ s.t. $\underline{m}^{[i]} \not\le_P \underline{m}^{[j]}$, for $i \ne j$, $1 \le i, j \le L$, we introduce 2-D arrays $u^{[i]} = (u^{[i]}_{\underline{j}})$, $1 \le i \le L$ as follows:

$$u^{[i]}_{\underline{j}} := \sum_{1 \le l \le n} \binom{\underline{j}}{\underline{m}^{[i]}} (\underline{x}^{(l)})^{\underline{j}-\underline{m}^{[i]}},\ \underline{j} \in \mathbf{Z}_0^2.$$

Then, we have

Corollary 1 $Q \in I(V, \Lambda) \Leftrightarrow Q \circ u^{[i]} = 0$, $1 \le i \le L$, *i.e.*

$$\sum_{\underline{k} \in \mathrm{Supp}(Q)} Q_{\underline{k}} u^{[i]}_{\underline{k}+\underline{j}} = 0,\ \underline{j} \in \mathbf{Z}_0^2,\ 1 \le i \le L.$$

Consequently, it turns out that GS list decoding can be solved by the multiple-array BMS algorithm [4], which is a modification of the BMS algorithm for finding a minimal polynomial set of a given finite set of 2-D arrays $u^{[i]}$, $1 \le i \le L$. Particularly, in case of GS list decoding of primal RS codes, we have to consider s 2-D arrays $u^{[i]} := (u^{[i]}_{\underline{j}})$, $1 \le i \le s$ defined by

$$u^{[i]}_{\underline{j}} := \sum_{1 \le l \le n} \binom{\underline{j}}{\underline{m}^{[i]}} (\underline{x}^{(l)})^{\underline{j}-\underline{m}^{[i]}},\ \underline{j} \in \mathbf{Z}_0^2.$$

for $\underline{x}^{(l)} = (\alpha^{l-1}, r_l) \in \mathbf{F}_q^2$, $1 \le l \le n$ and $\underline{m}^{[i]} := (i-1, s-i) \in \mathbf{Z}_0^2$, $1 \le i \le s$. Our method [5] of finding the interpolation function for GS list decoding with multiplicity s of RS codes of codelength n and coding rate R has computational complexity $\mathcal{O}(R^{-\frac{1}{2}} n^2 s^5)$ compared with $\mathcal{O}(n^3 s^6)$ of the method based on Gaussian elimination.

4 Conclusion

We have discussed how the BMS algorithm [1][2][6] is applied to several decoding methods of algebraic codes and multivariate interpolation, and how these decoding methods are related to Gröbner bases via multidimensional arrays and linear recurrences. In the sequel, we have clarified that these problems are reduced to finding a set of minimal polynomials, which corresponds to a Gröbner basis, of a given (set of) multidimensional array(s). We have given a basic set of algorithms for solving these problems, which constitute a unified system of unique methods in comparison with other various relevant methods related to Gröbner bases.

References

[1] S. Sakata, "Finding a minimal set of linear recurring relations capable of generating a given finite two-dimensional array," *J. Symbol. Comp.*, Vol.5, pp.321–337, 1988.

[2] S. Sakata, "Extension of the Berlekamp-Massey algorithm to N dimensions," *Inform. & Comp.*, Vol.84, pp.207–239, 1990.

[3] S. Sakata, H.E. Jensen, T. Høholdt, "Generalized Berlekamp-Massey decoding of algebraic geometric codes up to half the Feng-Rao bound," *IEEE Trans. Inform. Theory*, Vol.41, pp.1762–1768, 1995.

[4] S. Sakata, "N-dimensional Berlekamp-Massey algorithm for multiple arrays and construction of multivariate polynomials with preassigned zeros," in *Applied Algebra, Algebraic Algorithms and Error-Correcting Codes: Proc. AAECC-6 (Ed. T. Mora)*, Springer: Berlin, pp.356–376, 1989.

[5] Y. Numakami, M. Fujisawa, S. Sakata, "Fast interpolation methods for list decoding of RS codes" (in Japanese), *Trans. IEICE*, Vol.J83, pp.1309–1317, 2000.

[6] S. Sakata, "On the BMS Algorithm," preprint for Workshop on Gröbner bases in Cryptography, Coding Theory and Algebraic combinatorics, Linz, April 30 – May 5, 2006.

[7] V.D. Goppa, "Codes from curves," *Soviet Math. Dokl*, Vol.24, pp.170–172, 1981.

[8] T. Høholdt, J.H. van Lint, R. Pellikaan, *Handbook of Coding Theory*, §10: "Algebraic Geometry Codes," pp.871–961, 1998.

[9] M. Sudan, "Decoding of RS codes beyond the error-correction bound," *J. Complexity*, Vol.13, pp.180–193,1997.

[10] M.A. Shokrollahi, H. Wassermann, "List decoding of algebraic-geometric codes," *IEEE Trans. Inform. Theory*, Vol.45, pp.432–437, 1999.

[11] V. Guruswami, M. Sudan, "Improved decoding of RS codes and algebraic-geometric codes," *IEEE Trans. Inform. Theory*, Vol.45, pp.1757–1767, 1999.