The BMS Algorithm

Shojiro Sakata

Abstract We present a sketch of the n-dimensional (n-D) Berlekamp–Massey algorithm (alias Berlekamp–Massey–Sakata or BMS algorithm) w.r.t. n-D arrays. That is: (1) How is it related to Gröbner basis? (2) What problem can it solve? (3) How does it work? (4) Its variations. First we discuss another problem closely related to our main problem, and introduce some concepts about n-D linear recurrences and modules of n-D arrays as their general solutions. These two problems are just the inverse (or rather dual) to each other, which can be solved by the Buchberger algorithm (Buchberger in Ein Algorithmus zum Auffinden der Basiselemente des Restklassenringes nach einem nulldimensionalen Polynomideal, Ph.D. thesis, Innsbruck, 1965; J. Symb. Comput. 41(3–4):475–511, 2006; Multidimensional systems theory, Reidel, Dordrecht, pp. 184–232, 1985; Mora in Gröbner technology, this volume, pp. 11–25, 2009b), and the BMS algorithm, respectively. Furthermore, we discuss some properties of BMS algorithm and its outputs, including its computational complexity, as well as several variations of the BMS algorithm.

1 Introduction

In this paper, we present a sketch of the multidimensional Berlekamp–Massey algorithm (alias Berlekamp–Massey–Sakata algorithm or BMS algorithm) from (Sakata 1988, 1990). It is a generalization of the Berlekamp–Massey algorithm (Berlekamp 1968; Massey 1969) from one-dimensional (1-D) arrays to n-dimensional (n-D) arrays for $n \geq 1$. We discuss:

(1) How is it related to the Gröbner basis theory?
(2) What problem can it solve?
(3) How does it work?
(4) its several variations.

In another paper (Sakata 2009) of this issue we present its applications to decoding of algebraic error-correcting codes. In most part of this paper we restrict ourselves to treating finite fields although the contents remain valid in any field $\mathbb{F}$ provided that we have exact computations over $\mathbb{F}$.

Before we introduce our main theme, i.e. our main problem and its solution by the BMS algorithm, we discuss another problem closely related to it as well as some concepts which are important in this paper, where we call these problems

S. Sakata
The University of Electro-Communications, Chofu-shi, Tokyo 182-8585, Japan
e-mail: sakata@ice.uec.ac.jp

M. Sala et al. (eds.), *Gröbner Bases, Coding, and Cryptography*,
DOI 10.1007/978-3-540-93806-4_9, © Springer-Verlag Berlin Heidelberg 2009

primal and *dual*, respectively. The duality of the two problems are similar to the duality discussed by Mora (2009a). About these details we give several remarks in the following. Although the concept of "functional" given in *ibid.* is mathematically natural and more general, we use the terminology of "array" instead of that of functional for the convenience of our discussions. The descriptions of this chapter are rather elementary and intuitive, which are not necessarily refined mathematically as in *ibid.* The topic also turns to be a history of Gröbner basis in the world of Coding Theory.

Now we start to consider sequences (or 1-D arrays) and linear recurrences satisfied by them. The following is a linear recurrence over the real number field $\mathbb{R}$, which is satisfied by the famous Fibonacci sequence:

$$s_{j+2} - s_{j+1} - s_j = 0, \quad j \geq 0$$

When we start with the initial values $s_0 = 1, s_1 = 1$, we have not only the 1-D array $(s_j) = (1, 1, 2, 3, 5, 8, \ldots)$ but also an explicit form of the j-th element s_j for any $j \in \mathbf{N}$ (over a finite field we have another array, of course). Well, we generalize such 1-D arrays and 1-D linear recurrences to multidimensional arrays and multidimensional linear recurrences. For example, we consider the following system of two-dimensional (2-D) linear recurrences over $\mathbb{F}$:

$$\begin{cases} u_{i+2,j} + u_{i,j} = 0 \\ u_{i+1,j+1} + u_{i,j} = 0, \quad (i,j) \in \mathbf{N}^2 \\ u_{i,j+2} + u_{i,j} = 0 \end{cases}$$

In general such a condition as above is called a system of constant-coefficient *linear recurrence*s or (*partial*) *finite difference equation*s. Given a system of n-Dimensional (n-D) linear recurrences over any field $\mathbb{F}$, we want to find all n-D arrays satisfying them. It is just a *digital* version of finding the general solutions of a system of (homogeneous) constant-coefficient linear partial differential equations. We want to obtain not only a special solution but also the general solutions (the whole set of solutions). We treat multiple recurrences satisfied by n-D arrays. To discuss our problem in general we need some notation as follows.

An n-D array over a field $\mathbb{F}$ is a mapping u from $\mathbf{N}^n$ into $\mathbb{F}$. An array can be extended to a mapping (functional) from the n-variate polynomial ring $\mathcal{P} = \mathbb{F}[X_1, \ldots, X_n]$ into $\mathbb{F}$ so that a refinement of our discussions on duality can be given as in Mora (2009a). Since an n-D array u identifies a field element in $\mathbb{F}$ to an n-dimensional vector $a = (a_1, \ldots, a_n) \in \mathbf{N}^n$ of nonnegative integers (and hence to any term $X^a := X_1^{a_1} \cdots X_n^{a_n}$), we can associate to u the unique functional L taking the values of u on the corresponding term (and vice versa). If we know the values of a functional L on all terms, then by linearity we know all of its values. Moreover, the value of L on one term is completely independent from its value on any other term. We persist in using the terminology of *array*, which has been used in most literature on the (applications of) BMS algorithm. We denote an array u also as $u = (u_a)_{a \in \mathbf{N}^n}$, where $u_a := u(a) \in \mathbb{K}$ and we consider these elements to be arranged on the whole n-D integral lattice which is identified with $\mathbf{N}^n$. Let $\mathcal{A}$ be the set of all n-D arrays over $\mathbb{F}$ defined on $\mathbf{N}^n$, and introduce basic operations upon arrays $u \in \mathcal{A}$. Naturally, we have the sum of two arrays $u = (u_a), v = (v_a) \in \mathcal{A}$ as $u + v = (u_a + v_a) \in \mathcal{A}$, and the scalar product of u by an element c of the coefficient field $\mathbb{F}$ as $cu = (cu_a) \in \mathcal{A}$. Furthermore, we consider polynomials $f = \sum_{a \in \mathrm{supp}(f)} c(f,a)X^a \in \mathcal{P}$. In this paper we often refer to exponents (integer vectors) $a = (a_1, \dots, a_n) \in \mathbf{N}^n$ as basic entities instead of terms $\tau = X^a$ so that we denote the coefficient of $\tau = X^a$ as $c(f,a)$ instead of $c(f, X^a)$. We call the (finite) set of exponents a $(\in \mathbf{N}^n)$ of its nonzero monomials $c(f,a)X^a$ (having the nonzero coefficient $c(f,a) \in \mathbb{K} \setminus \{0\}$) by the name of the *support*[1] of f and denote it as $\mathrm{supp}(f)$ $(\subset \mathbf{N}^n)$. A polynomial $f \in \mathcal{P}$ is operated on an array $u \in \mathcal{A}$ (it is the same as multiplying a functional L by f) so that the following array v is obtained:

$$v = f \circ u := (v_b) \in \mathcal{A}, \qquad v_b := \sum_{a \in \mathrm{supp}(f)} c(f,a) u_{a+b}, \quad b \in \mathbf{N}^n$$

This *polynomial operation* '$f\circ$' is just a transformation of an array u to another array v. In particular, the operation by the monomial $f = X_j$, $1 \le j \le n$ is a unit shift along the X_j-axis (to the negative direction), where $v = X_j \circ u = (v_a)$, $a \in \mathbf{N}^n$ has the elements $v_{a_1,\dots,a_j,\dots,a_n} = u_{a_1,\dots,a_j+1,\dots,a_n}$ (the elements of u which are put out of the domain $\mathbf{N}^n$ are pruned away). For example, in case of $n = 1$, for $X := X_1$ and $u = (u_i)$, the unit-shifted array $v = X \circ u = (v_i)$ has the elements $v_i = u_{i+1}$, $i \in \mathbf{N}$, and the double-shifted array $w = X^2 \circ u = (w_i)$ has $w_i = u_{i+2}$, $i \in \mathbf{N}$, etc.[2] Consequently, the module $\mathcal{A}$ is a $\mathcal{P}$-module, i.e. a module with the ring $\mathcal{P}$ of operators. By using this notation, we can write any linear recurrences with the *characteristic polynomial*s $F = \{f^{(1)}, \dots, f^{(\mu)}\}$ $(\subset \mathcal{P})$ as follows,

$$f^{(i)} \circ u = 0, \quad 1 \le i \le \mu, \tag{1}$$

where 0 is the all-zero array. From now on, we do not distinguish between linear recurrences and the corresponding characteristic polynomials, identifying them. That is, for simplicity, provided that the formula (1) holds, we often say that the array u satisfies the polynomial $f^{(i)}$, and that 'the polynomial $f^{(i)}$ is *valid* for the array u,' etc. For a given $F \subset \mathcal{P}$, it is easy to see that the set $\mathcal{A}(F)$ of solutions u of (1) is a $\mathcal{P}$-submodule of the $\mathcal{P}$-module $\mathcal{A}$, since $f \circ (g \circ u) = (fg) \circ u$ for $f, g \in \mathcal{P}$. For example, for a univariate polynomial $f = x^2 - x - 1$ over $\mathbb{R}$, $\mathcal{A}(f)$ is the set of 1-D arrays (Fibonacci sequences) $u = (u_i)$ which are obtained by setting any initial values u_0, u_1 and then uniquely by determining the other values u_i, $i \ge 2$ iteratively with the linear recurrence $f \circ u = 0$. In general, for any polynomial set $F \subset \mathcal{P}$ and the ideal $\mathbf{I}(F) := \langle F \rangle_{\mathcal{P}}$ $(\subset \mathcal{P})$ generated by F, $\mathcal{A}(F) = \mathcal{A}(\mathbf{I}(F)) := \{u \in \mathcal{A} \mid f \circ u = 0,\ f \in \mathbf{I}(F)\}$.

[1] In Mora (2009a) the support is defined as a subset of the whole set $\mathcal{T}$ of terms X^a, $a \in \mathbf{N}^n$.

[2] Trivially, by multiplying a polynomial $g = \sum_{0 \le i \le d} g_i X^i$ with X, one gets the polynomial $\bar{g} = Xg = \sum_{0 \le i \le d} g_i X^{i+1} = \sum_{1 \le i \le d+1} g_{i-1} X^i$, where the array of coefficients of its monomials is obtained by shifting to the positive direction: $(g_i) \to (g_{i-1})$ in contrast with the above shift (to the negative direction) by operation X.

As is seen, in case of $n = 1$, we can easily obtain $\mathcal{P}$-submodules $\mathcal{A}(f)$ and $\mathcal{A}(F)$ of 1-D arrays. In particular, for $F = \{f^{(1)}, \ldots, f^{(\mu)}\}$ $(\subset \mathbb{F}[X])$, $\mathcal{A}(F) = \mathcal{A}(g)$, where $g = \mathrm{gcd}(F)$ (gcd = greatest common divisor). However, it is not so easy to obtain $\mathcal{A}(F)$ in case of $n \geq 2$ as in case of $n = 1$. In case of $n \geq 2$, it is difficult not only to give the solutions of a system of homogeneous linear recurrences (1) but also to specify even the positions of initial values. In fact, the Buchberger algorithm gives the solution of the present problem, which we will mention in the next section. For our discussions, we need some variants of basic notion from the Gröbner basis theory. In this paper we consider any term ordering $<$ over $\mathbf{N}^n$, although it usually is defined over the set $\mathcal{T} = \{X^a \mid a \in \mathbf{N}^n\}$ of terms in the Gröbner basis theory. Furthermore, we often consider the leading exponent $\mathrm{le}(f) := \max_< \mathrm{supp}(f)$ $(\in \mathbf{N}^n)$ of f instead of the corresponding leading term $\mathbf{T}(f) = X^{\mathrm{le}(f)} \in \mathcal{T}$.

2 Generating Arrays

We consider the problem of initial value positions via the following example in case of $n = 2$. Now we assume for a set of polynomials $F = \{f^{(1)}, \ldots, f^{(\mu)}\}$ $(\subset \mathbb{F}[X_1, X_2])$ that the leading exponents $\mathrm{le}(f^{(i)}) = d^{(i)} = (d_1^{(i)}, d_2^{(i)}) \in \mathbf{N}^2$, $1 \leq i \leq \mu$ of its elements satisfy

$$d_1^{(1)} > d_1^{(2)} > \cdots > d_1^{(\mu-1)} > d_1^{(\mu)} = 0, \qquad d_2^{(1)} = 0 < d_2^{(2)} < \cdots < d_2^{(\mu-1)} < d_2^{(\mu)}$$

Then, $\mathbf{N}^2$ can be split into two parts:

$$\Sigma(F) := \{a \in \mathbf{N}^2 \mid a \geq_P d^{(i)}, 1 \leq^{\exists} i \leq \mu\}, \qquad \Delta(F) := \mathbf{N}^2 \setminus \Sigma(F),$$

where $\geq_P$ is the natural partial ordering over $\mathbf{N}^n$. These subsets have the following properties and are called *stable* sets (sometimes the former and latter sets are called *upper* and *lower set*s, respectively),

$$\begin{aligned} &a \in \Sigma(F), \quad && b \in \mathbf{N}^2, \quad && a \leq_P b \quad \Rightarrow \quad b \in \Sigma(F); \\ &a \in \Delta(F), \quad && b \in \mathbf{N}^2, \quad && a \geq_P b \quad \Rightarrow \quad b \in \Delta(F). \end{aligned}$$

If F is a Gröbner basis (w.r.t. $<$) of an ideal $\mathbf{I}$, there are complementary subsets $\mathbf{T}(\mathbf{I})$ and $\mathbf{N}(\mathbf{I})$ $(\subset \mathcal{T})$ which are determined uniquely by $\mathbf{I}$ (Mora 2009a). Now we consider any stable subsets without assuming any knowledge of Gröbner basis. The latter set $\Delta(F)$ called *delta-set* or *footprint* seemingly can be used as the initial value positions. That is, after having specified any values $u_a \in \mathbb{F}$, $a \in \Delta(F)$ as the initial values, we proceed to find each of the remaining values u_b, $b \in \Sigma(F)$ iteratively by using the following (pseudo)algorithm of generating an array up to an prescribed position $r \in \mathbf{N}^2$. In the following we denote the next greater point (w.r.t. the term ordering $<$) of any point $a \in \mathbf{N}^2$ as $a \oplus 1$, and define $\overline{\mathrm{supp}}(f) := \mathrm{supp}(f) \setminus \{\mathrm{le}(f)\}$ $(\subset \mathbf{N}^2)$, $\Sigma^r := \{b \in \mathbf{N}^2 \mid b < r\}$; $\min_< \Sigma(F)$ is the minimum element of $\Sigma(F)$ w.r.t. $<$.

Table 1 Initial values u_a, $a \in \Delta(F)$ and partial array u_a, $a \in \Sigma^{(1,2)}$

$a_1 \setminus a_2$	0	1	2	3
0	1	0		
1	1			
2				
3				
4				

$a_1 \setminus a_2$	0	1	2	3
0	1	0	1	
1	1	1		
2	1	0*		
3	1			
4				

Algorithm 1 Generating an array u_b, $b \in \Sigma^r$;

```
Step 1 (initialization): $b := \min_< \Sigma(F)$;
Step 2 (computation): if $b \in \Sigma(F)$ then
        begin let $i$ be any $i$, $1 \le i \le \mu$ s.t. $d^{(i)} \le_P b$;
            $u_b := \frac{1}{\mathrm{lc}(f^{(i)})}\Big(-\sum_{a \in \overline{\mathrm{supp}}(f^{(i)})} c(f^{(i)}, a) u_{a+b-d^{(i)}}\Big)$
        end;
Step 3 (termination): $b := b \oplus 1$; if $b < r$ then go to Step 2 else stop.
```

Although it seems that we could find an array u before the terminal point r by using this algorithm, it will turn out naturally that we do not always succeed in getting a proper array having the specified initial values and satisfying all of the given linear recurrences. In Step 2, the value u_b is determined by using the polynomial $f^{(i)}$ so that one of the desired conditions

$$f^{(i)}[u]_b := \sum_{a \in \mathrm{supp}(f^{(i)})} c(f^{(i)}, a) u_{a+b-d^{(i)}} = 0$$

is satisfied, but some conditions corresponding to other polynomials $f^{(j)}$, $j \neq i$ with $d^{(j)} \le_P b$ might not always be satisfied. Consider the previous example (1) over $\mathbb{F}_2$. The linear recurrences are specified by $F = \{f^{(1)} := X_1^2 + 1, f^{(2)} := X_1 X_2 + 1, f^{(3)} := X_2^2 + 1\}$, and the delta-set is $\Delta(F) = \{(0,0), (1,0), (0,1)\}$. Starting with the initial values shown in the left half of Table 1 we proceed to find the other values of u iteratively w.r.t. the graded reverse lexicographic ordering $<$ (i.e. the term ordering with the weight $w = (1,1)$ associated with the reverse lexicographic ordering $X_1 <_L X_2$). Then, we have the intermediate result shown in the right half of Table 1.

The value $u_{2,1} = 0$ with signature * is found by using $f^{(1)}$, but it does not satisfy the linear recurrence of $f^{(2)}$. Thus, there exists no array u which has the initial values and is a solution of the linear recurrences (1) over $\mathbb{F}_2$. In other words, the above delta-set is not appropriate as a set of positions for initial values. On taking into consideration the properties of Gröbner basis, it might be hit upon that the polynomial set F is not a Gröbner basis and that the corresponding delta-set is too large to be a set of positions for initial values so that the algorithm[3] for generating arrays fails to find proper arrays. In fact, it is easy to see that the reduced Gröbner basis (w.r.t. the term ordering $<$ with $w = (1, 1)$) of the ideal $I = \langle F \rangle_{\mathcal{P}}$ is $\{X_1^2 + 1, X_2 + X_1\}$, and that the proper set of positions for initial values is $\{(0, 0), (1, 0)\}$. As we have seen, the problem of finding the proper set of positions for initial values, given a set of linear recurrences or its characteristic polynomials, is just identical with that of finding a Gröbner basis of the ideal $\mathbf{I}(F)$. Furthermore, iteratively from F, we can find a polynomial $f^{(b)} := X^b - \sum_{a \in \Delta(F)} c(f^{(b)}, a) X^a \in \mathbf{I}(F)$, by which we can get the value u_b for any $b \in \Sigma(F)$ directly from any given values u_a, $a \in \Delta(F)$, so that the so-called S-polynomial at any outer corner point outside $\Delta(F)$ can be obtained. By repeating reductions of such S-polynomials modulo F and consequent modifications of F and $\Delta(F)$, we finally get a Gröbner basis. Of course, it is just the Buchberger algorithm. Let us illustrate it with the previous example. We get $X_2 f^{(1)} = X_1^2 X_2 + X_2 \in \mathbf{I}(F)$ from $f^{(1)} = X_1^2 + 1$, and $X_1 f^{(2)} = X_1^2 X_2 + X_1$ from $f^{(2)} = X_1 X_2 + 1$. Then, we obtain the S-polynomial $f^{(3)} := X_2 f^{(1)} - X_1 f^{(2)} = X_2 + X_1 \in \mathbf{I}(F)$ at the corner point (2, 1), which is not reducible further modulo F. Finally, $\{f^{(1)}, f^{(3)}\}$ turns out to be the reduced Gröbner basis.

As a summary, we have that the problem of finding a module of linear recurrences is equivalent to that of finding a Gröbner basis of its characteristic polynomials, and thus these problems can be solved by the same algorithm. In the world of Coding Theory, it is a historical fact that the concept (Ikai et al. 1976) equivalent to Gröbner basis and an algorithm (Sakata 1981) equivalent to the Buchberger (1965, 1985, 2006) algorithm were introduced in the process of solving a certain problem of constructing a kind of multidimensional codes independently. In general, as it is shown in Lemma 4 of Mora (2009a), the general solution of the system of linear recurrences (1) is just the $\mathcal{P}$-module L which is dual (in the sense of Mora 2009a) to the ideal $\mathbf{I}$ generated by $F = \{f^{(1)}, \ldots, f^{(\mu)}\}$ and $\dim_{\mathbb{F}} L = \#\mathbf{N}(\mathbf{I})$,[4] where $\mathbf{N}(\mathbf{I})$ is the delta-set of the Gröbner basis of $\mathbf{I}$ which is called *Gröber escalier* in Mora (2009a), provided $\dim_{\mathbb{F}} L < \infty$, i.e. $\mathbf{I}$ is a zero-dimensional ideal.

3 BMS Algorithm

Our main problem is just the inverse (or rather dual from the viewpoint of duality in Mora 2009a) to the problem of finding the general solution of a given system of linear recurrences which we have discussed in the previous section. Now, for an integer μ, we are given a pair of sets $U := \{u^{(i)} \mid 1 \le i \le \mu\}$ and $V := \{v^{(i)} \mid 1 \le$

[3] As it is seen from these considerations, Algorithm 1 can be an actual algorithm if and only if F is a Gröbner basis.

[4] In Sakata (1978), this fact is described in the terminology of array, which is a little bit different from Mora (2009a).

$i \leq \mu\}$ of infinite (periodic) n-D arrays, i.e. $U, V \subset \mathcal{A}$, and consider the following linear recurrences corresponding to (unknown) polynomials $f \in \mathcal{P}$:

$$f \circ u^{(i)} = v^{(i)}, \quad 1 \leq i \leq \mu \tag{2}$$

We might be required to find a valid polynomial f having a (unknown) minimal leading exponent $\mathrm{le}(f)$. In most part of this paper we concentrate upon the homogeneous problem with the right-hand side arrays $v^{(i)} = 0$, $1 \leq i \leq \mu$, leaving the non-homogeneous problem (see Sect. 4).

It is easy to see that the following set of polynomials is an ideal of $\mathcal{P} = \mathbb{F}[X_1, \ldots, X_n]$, which we call the *characteristic ideal* of the given set U of arrays $u^{(i)}$, $1 \leq i \leq \mu$:

$$\mathbf{I}(U) := \{f \in \mathcal{P} \mid f \circ u^{(i)} = 0,\ 1 \leq i \leq \mu\}$$

This is the ideal dual to a $\mathcal{P}$-module generated by U in the terminology of Mora (2009a).

In this section, we treat only the case of a single array,[5] i.e. $U = \{u\} \subset \mathcal{A}$, and we will give a method of finding a Gröbner basis of the characteristic ideal $\mathbf{I}(u) := \{f \in \mathcal{P} \mid f \circ u = 0\}$. We want to find a set of polynomials $f = \sum_{a \in \mathrm{supp}(f)} c(f,a) X^a$ with a minimal leading exponent $d = \mathrm{le}(f)$ which satisfy $f \circ u = 0$, i.e.

$$\sum_{a \in \mathrm{supp}(f)} c(f,a) u_{a+b} = 0, \quad b \in \mathbf{N}^n \tag{3}$$

We try to find a set of polynomials with a minimal leading exponent satisfying a (partial) condition specified by a finite part of a given infinite array u.

To be more precise, we introduce some notations. According to a specified term ordering $<$ over $\mathbf{N}^n$, we arrange the points $a \in \mathbf{N}^n$ so that we have $\mathbf{N}^n = \{a^{(0)} = 0, a^{(1)}, a^{(2)}, \ldots, a^{(i)}, \ldots \mid a^{(i+1)} = a^{(i)} \oplus 1,\ i \in \mathbf{N}\}$, and a partial array $u^b := (u_a)$, $a < b$ for any point $b \in \mathbf{N}^n$. If a polynomial $f = \sum_{a \in \mathrm{supp}(f)} c(f,a) X^a$ $(\in \mathcal{P})$ satisfies for a certain $r \in \mathbf{N}^n$

$$f[u]_b := \sum_{a \in \mathrm{supp}(f)} c(f,a) u_{a+b-d} = 0, \quad d = \mathrm{le}(f) \leq_P b < r, \tag{4}$$

we say that f is *valid* (w.r.t. u) *before* r. From now on, having fixed an array u, we often omit the phrase "w.r.t. u." Furthermore, if the condition (4) holds and $f[u]_r \neq 0$, then we say that f is *not valid at the point r for the first time*. A monic (i.e. $\mathrm{lc}(f) = 1$) polynomial f which is valid before a and whose leading exponent $\mathrm{le}(f)$ is minimal w.r.t. the partial ordering $\leq_P$ is called a *minimal polynomial of the partial array u^a*. Since there exist in general plural minimal polynomials f with distinct leading exponents $\mathrm{le}(f)$ of a given partial array u^a, we can define a minimal polynomial set $F(a)$ (or simply, F) of u^a associated with a finite set of

[5] For the general case of multiple arrays, see Sect. 4.

points $D(a) = \{\mathrm{le}(f) \mid f \in F(a)\} \subset \mathbf{N}^n$ s.t. there is a single element $f \in F(a)$ with $\mathrm{le}(f) = d$ and $\mathrm{lc}(f) = 1$ for each $d \in D(a)$ and there exists no polynomial h with $\mathrm{le}(h) \in \Delta(a)$ which is valid (w.r.t. u) before a, where for $\Sigma_d := \{b \in \mathbf{N}^n \mid b \geq_P d\}$, we have a complementary pair of subsets $\subset \mathbf{N}^n$:

$$\Sigma(a) := \bigcup_{d \in D(a)} \Sigma_d, \qquad \Delta(a) := \mathbf{N}^n \setminus \Sigma(a)$$

In addition to $D(a)$, letting $\Gamma_c := \{b \in \mathbf{N}^n \mid b \leq_P c\}$ for $c \in \mathbf{N}^n$, we have a finite subset $C(a)(\subset \mathbf{N}^n)$ s.t. $\Delta(a) = \bigcup_{c \in C(a)} \Gamma_c$. We call $\Delta(a)$ the *delta-set* of $F(a)$, which is, roughly speaking, in form of a stack of multidimensional building blocks and whose apices (corner points) are $c \in C(a)$. As above-mentioned, there exists no polynomial h with $\mathrm{le}(h) \in \Delta(a)$ which is valid before a. These subsets $D(a)$, $C(a)$, $\Sigma(a)$ and $\Delta(a)$ are unique for the given array u^a, but a minimal polynomial set $F(a)$ is not necessarily unique for u^a. In view of the definition of minimal polynomial set $F(a)$, $\Delta(a) \subseteq \Delta(a \oplus 1)$. Similar notations can be used for an infinite array u, e.g. a minimal polynomial set $F(\subset \mathcal{P})$ of u, the delta-set $\Delta(\subset \mathbf{N}^n)$ of u, etc. if they exist.

The BMS algorithm is just to find a minimal polynomial set $F(a)$ of a given partial array u^a for a fixed point $a \in \mathbf{N}^n$. Starting with the origin $0 \in \mathbf{N}^n$, we proceed to find a minimal polynomial set $F(b)$ of the partial array u^b iteratively at each point $b \leq a$ accordingly to the term ordering $<$. If $f \in F(b)$ is valid still at $b \oplus 1$, then $f \in F(b \oplus 1)$. However, if some $f \in F(b)$ is not valid at b, then we must update these invalid f. Whether $\Delta(a \oplus 1) = \Delta(a)$ or not depends on certain relations among a, $D(a)$ and $C(a)$. The following basic lemma (Sakata 1990) describing this fact stipulates the main procedure of the BMS algorithm.

Lemma 1 *If a polynomial f is not valid* (*w.r.t. u*) *for the first time at a, i.e.*

$$f[u]_b = 0, \quad d = \mathrm{le}(f) \leq_P b < a; \qquad f[u]_a \neq 0,$$

then there exists no polynomial g with $\mathrm{le}(g) = d' \leq_P a - d$ *satisfying the following condition*:

$$g[u]_b = 0, \quad d' \leq_P b \leq a$$

Lemma 1 is very important because it determines the delta-set $\Delta(a \oplus 1)$ and its complement $\Sigma(a \oplus 1)$, where the minimal (w.r.t. the partial ordering $<_P$) points of $\Sigma(a \oplus 1)$ are just identical with the set $\{\mathrm{le}(f) \mid f \in F(a \oplus 1)\}$ of leading exponents of all elements of a minimal polynomial set $F(a \oplus 1)$.

Based on Lemma 1 we define the *discrepancy*, *fail* and *span*[6] of f, respectively, as

$$\mathrm{dis}(f) := f[u]_a (\neq 0), \qquad \mathrm{fail}(f) := a, \qquad \mathrm{span}(f) := a - d\,(= \mathrm{fail}(f) - \mathrm{le}(f))$$

[6] This is not the usual ‘linear span.’

In addition we introduce the following notation for later convenience.

$$\begin{aligned} \mathrm{Val}(u^b) &:= \{f \in \mathcal{P} \mid f[u]_a = 0,\ \mathrm{le}(f) \leq_P a < b\} \\ \mathrm{mVal}(u^b) &:= \{f \in \mathrm{Val}(u^b) \mid \mathrm{le}(f) : \text{minimal}\} \\ \mathrm{Aux}(u; c) &:= \{g \in \mathcal{P} \mid \mathrm{span}(g) = c\} \\ \mathrm{mAux}(u; c) &:= \{g \in \mathrm{Aux}(u; c) \mid \mathrm{le}(g) : \text{minimal}\} \end{aligned}$$

Associated with the finite subset $C(a)$ related to the delta-set $\Delta(a)$, we have a finite set of polynomials $G(a) := \{g \mid \mathrm{span}(g) \in C(a)\}$, which we call an *auxiliary polynomial set* of u^a. An auxiliary polynomial $g \in G(a)$ is characterized by the property that it has a maximal (w.r.t. the partial ordering $\leq_P$) $\mathrm{span}(g)$ among the polynomials s.t. $\mathrm{fail}(g) < a$. If a minimal polynomial $f \in F(a)$ fails to be valid at a, minimal polynomial(s) $f' \in F(a \oplus 1)$ at $a \oplus 1$ can be obtained by using appropriate auxiliary polynomial(s) $g \in G(a)$ (if it exists) as shown in Lemma 2 or without any $g \in G(a)$. First we have in view of Lemma 1 that, if there exists a polynomial $f \in F(a)$ with $d = \mathrm{le}(f)$ which is not valid at a and $a - d \notin \Delta(a)$, then $\Delta(a \oplus 1) \neq \Delta(a)$. Thus, we define $F_{\mathrm{fail}} := \{f \in F(a) \mid \mathrm{fail}(f) = a\}$, $F_{\mathrm{fall}} := \{f \in F_{\mathrm{fail}} \mid a - d \notin \Delta(a)\}$, $D_{\mathrm{fall}} := \{\mathrm{le}(f) \mid f \in F_{\mathrm{fall}}\}$. Furthermore, for $c = (c_i)_{1 \leq i \leq n} (\in C(a))$, let $\max(d, a - c) := (\max\{d_i, a_i - c_i\})_{1 \leq i \leq n}$ $(\in \mathbf{N}^n)$, and let D' be the set of minimal elements d' in $D'' := \{d' := \max(d, a - c) \mid d \in D_{\mathrm{fall}}, c \in C(a)\}$ $(\subset \mathbf{N}^n)$, and let $\hat{D}$ be the set of minimal elements in $\Sigma(a) \setminus \Gamma_a$. Now we have the following lemma about how to update F (Sakata 1990).

Lemma 2 (1) *For* $f \in F_{\mathrm{fail}} \setminus F_{\mathrm{fall}}$, *there exists* $c \in C(a)$ *s.t.* $d \geq_P a - c$. *In this case, by using an auxiliary polynomial* $g \in G(a)$ *s.t.* $\mathrm{span}(g) = c$, *we obtain*

$$h := f - \frac{\mathrm{dis}(f)}{\mathrm{dis}(g)} X^{d-(a-c)} g \in F(a \oplus 1)$$

(2) *For a pair* $(f, g) \in F_{\mathrm{fall}} \times G(a)$ *with* $d = \mathrm{le}(f)$, $c = \mathrm{span}(g)$, *respectively, if it holds that* $d' := \max(d, a - c) \in D'$, *then we obtain*

$$h := X^{(a-c)-dd'-d} f - \frac{\mathrm{dis}(f)}{\mathrm{dis}(g)} g \in F(a \oplus 1)$$

(3) *For* $\hat{d} \in \hat{D}$, *if there exists no* $d' \in D'$ *s.t.* $\hat{d} \geq_P d'$, *then, by using* $f \in F_{\mathrm{fail}}$ *s.t.* $\hat{d} \geq_P d = \mathrm{le}(f)$, *we obtain*

$$h := X^{\hat{d}-d} f \in F(a \oplus 1)$$

Based on the above observations we have the following form of the BMS algorithm, whose validity can be proven based on Lemmas 1, 2, where some notational simplicities are used, i.e. minimal polynomial set $F(b)$ and *auxiliary polynomial set* $G(b)$ at each point b are denoted simply as F and G, respectively. For $f \in F$,

$g \in G$, let $d := \mathrm{le}(f)$, $c := \mathrm{span}(g)$, $d_f := \mathrm{dis}(f)$, $d_g := \mathrm{dis}(g)$,

$$D = \{d = \mathrm{le}(f) \mid f \in F\}, \quad C = \{c = \mathrm{span}(g) \mid g \in G\},$$

and $\varSigma = \varSigma(b)$, $\varDelta = \varDelta(b)$ at the beginning of Step 2. A simple example of its computation is shown in Appendix A.

Algorithm 2 (BMS algorithm) Finding a minimal polynomial set of a finite n-D array u^r over $\mathbb{F}$ (Sakata 1988, 1990);[7]

```
Step 1  (initialization): $b := 0$; $F := \{1\}$; $D := \{0\}$; $\varSigma := \mathbf{N}^n$;
        $G := \emptyset$; $C := \emptyset$; $(\varDelta := \emptyset;)$
Step 2  (discrepancy): for each $f \in F$, $d_f := f[u]_b$;
        $F_{\mathrm{fail}} := \{f \in F \mid d_f \neq 0\}$;
        $F_{\mathrm{fall}} := \{f \in F_{\mathrm{fail}} \mid \nexists c \in C \text{ s.t. } d \geq_P b - c\}$;
        $D_{\mathrm{fall}} := \{d = \mathrm{le}(f) \in D \mid f \in F_{\mathrm{fall}}\}$; $\hat{D} := \{\text{minimal } \hat{d} \in \varSigma \setminus \Gamma_b\}$;
        $D'' := \{\max(d, b-c) \mid d \in D_{\mathrm{fall}},\ c \in C\}$; $D' := \{\text{minimal } d' \in D''\}$;
Step 3  (updating): (1) for each $f \in F_{\mathrm{fail}} \setminus F_{\mathrm{fall}}$
                          begin $h := f - d_f X^{d-(b-c)} g$ (for $g \in G$ s.t. $d \geq_P b - c$);
                            $F' := F \cup \{h\}$ end;
                        for each $(f, g) \in F_{\mathrm{fall}} \times G$ s.t. $d' := \max(d, b-c) \in D'$
                          begin $h := X^{\hat{d}-dd'-d} f - d_f g$; $F' := F \cup \{h\}$ end;
                        for each $\hat{d} \in \hat{D}$ if $\nexists d' \in D'$ s.t. $\hat{d} \geq_P d'$ then
                          for $f \in F_{\mathrm{fall}}$ s.t. $d \leq_P \hat{d}$
                              begin $h := X^{\hat{d}-d} f$; $F' := F \cup \{h\}$ end;
                    (2) $F := F' \setminus F_{\mathrm{fail}}$; $G'' := \{g \in G \mid \exists f \in F_{\mathrm{fall}} \text{ s.t. } c <_P b - d\}$;
                        $G := (G \cup \{\frac{1}{d_f} f \mid f \in F_{\mathrm{fall}}\}) \setminus G''$;
                        $D := \{\mathrm{le}(f) \mid f \in F' \setminus F_{\mathrm{fall}}\}$; $\varSigma := \bigcup_{d \in D} \varSigma_d$; $(\varDelta := \bigcup_{c \in C} \Gamma_c;)$
                        $C := (C \cup \{b - d \mid \exists f \in F_{\mathrm{fall}} \text{ s.t. } b - d >_P c\})$
                              $\setminus \{c \in C \mid \exists f \in F_{\mathrm{fall}} \text{ s.t. } b - d >_P c\}$;
Step 4  (termination): $b := b \oplus 1$; if $b < r$ then go to Step 2 else stop.
```

A minimal polynomial set $F(b)$ is not necessarily unique for the given array u. Let $\mathcal{F}$ be the class of all reduced minimal polynomial sets $F = F(b)$ of u^b, where $F \in \mathcal{F}$ is said to be *reduced* iff any $f \in F$ has support $\mathrm{supp}(f)$ s.t. $\overline{\mathrm{supp}}(f) := \mathrm{supp}(f) \setminus \{\mathrm{le}(f)\}$ is contained in the delta-set $\varDelta := \varDelta(a)$. Let $\mathrm{le}(f) + \varDelta := \{\mathrm{le}(f) + a \mid a \in \varDelta\}$. Now, we have the following theorems (Sakata 1990) about the complete class $\mathcal{F} = \mathcal{F}(b)$ which consists of all minimal polynomial sets $F = F(b)$ of u^b and the condition of uniqueness of F, i.e. $\#\mathcal{F} = 1$.

Theorem 1 (Complete class) *Let $F \in \mathcal{F}$ $(= \mathcal{F}(b))$ and $G = G(b)$ be a minimal polynomial set and an auxiliary polynomial set of u^b with $D = D(b)$, $C = C(b)$*

[7] The 1-D case of this algorithm is reduced to a refined version of the well-known Berlekamp–Massey (BM) algorithm (Berlekamp 1968; Massey 1969).

s.t. $\Sigma = \Sigma(b) = \bigcup_{d \in D} \Sigma_d$ *and* $\Delta = \Delta(b) = \bigcup_{c \in C} \Gamma_c$. *Take a minimal polynomial* $f \in F$ *and an* $F' \in \mathcal{F}$. *Then, any* $f' \in F'$ *with* $\mathrm{le}(f') = \mathrm{le}(f) = d \in D$ *is of the form*:

$$f' = f + \sum_{g \in G_d} h_g g,$$

where $h_g \in \mathcal{P}$, $\mathrm{le}(h_g) \le d + \mathrm{span}(g) - b$, *and* $G_d := \{g \in G \mid d + \mathrm{span}(g) \ge_P b\}$.

Theorem 2 (Uniqueness) *Let* $F \in \mathcal{F}$. *Then, we have that* $\#\mathcal{F} = 1$ *iff*

$$\bigcup_{f \in F} (\mathrm{le}(f) + \Delta) \subseteq \Sigma^b,$$

or in other words,

$$\max_{<}\{\mathrm{le}(f) + \mathrm{fail}(g) - \mathrm{le}(g) \mid f \in F, g \in G\} < b,$$

where G *is an auxiliary polynomial set of* u^b, *and* $\max_{<}\{\cdots\}$ *is the maximum* (*w.r.t.* $<$) *element of the set* $\{\cdots\}$.

Theorem 3 (Gröbner basis of $\mathbf{I}(u)$) *Let* $P = \{\sum_{1 \le i \le n} c_i a^{(i)} \in \mathbf{N}^n \mid 0 \le c_i \le 1, c_i \in \mathbf{Q}, 1 \le i \le n\}$ *be a fundamental period parallelotope* $P \subset \mathbf{N}^n$ *of an infinite n-D periodic array* u *s.t.* $u_{b+a^{(i)}} = u_b$ *for any* $b \in \mathbf{N}^n$, *and let* $2P := \{a + c \mid a, c \in P\}$. *Then, if the subset* $\Sigma^b = \{a \in \mathbf{N}^n \mid a < b\}$ *contains* $2P$, *a minimal polynomial set* F *of the partial array* u^b *is a Gröbner basis of* $\mathbf{I}(u)$.

In real applications of the BMS algorithm, we usually know in advance the approximate size $\#\Delta(F)$ for the delta-set $\Delta(F)$ of the Gröbner basis F (without any other knowledge of F itself). In such cases, in view of Theorem 2, we can terminate the iterations of the BMS algorithm much earlier than described in Theorem 3. We assume the term ordering $<$ with the weight $w = (w_i)_{1 \le i \le n}$ whose elements w_i are almost equal to each other, i.e. $w_1 \sim w_2 \sim \cdots \sim w_n$. This is just the case that the Buchberger algorithm has the least complexity. Let $m := \#\Delta$ for the delta-set Δ of the Gröbner basis which is the minimal polynomial set F at the termination point, and let $\mu := \#F$ ($\sim \#G$). Then, if the computational complexity[8] of the BMS algorithm is measured as the total number of arithmetic operations over the finite field $\mathbb{F}$, in view of $\mu \sim m^{1-\frac{1}{n}}$, it is $\mathcal{O}(\mu m^2) \sim m^{3-\frac{1}{n}}$ when n is fixed. This complexity is somewhat better than $\mathcal{O}(m^3)$ of any relevant algorithm based on the usual Gaussian elimination if n is not large. We should remark that we can have various modifications of the original BMS algorithm and that the computational complexities of these versions are reduced considerably when they are applied to various practical

[8] To determine the set of $D' \cup \hat{D}$ of $\mathrm{le}(f')$, $f' \in F(b \oplus 1)$ we need to have some combinatorial manipulation, particularly finding minimal (w.r.t. $<_P$) elements d' of D''. We omit complexity of such integer operations which are independent from finite field arithmetic.

problems including decoding of algebraic error-correcting codes because they cleverly can make use of the structures or properties of the given input data (i.e. arrays) which depend on each individual problem (Sakata 2009, 1989; Sakata et al. 1995).

4 Variations

In this section we present several versions of the BMS algorithm, each of which solves a distinct extension or generalization of the original BMS problem, respectively. The following are a list of these problems.

(1) **Multiarray BMS problem** (Sakata 1989; Feng and Tzeng 1989, 1991)

Given a finite set $U = \{u^{(i)} \mid 1 \le i \le \mu\}$ of finite n-D arrays over $\mathbb{F}$, where all component arrays $u^{(i)}$, $1 \le i \le \mu$ are defined over $\Sigma^r \subset \mathbf{N}^n$ (w.r.t. a fixed term ordering $<$) for a certain point $r \in \mathbf{N}^n$;

Find a minimal polynomial set F composed of polynomials f which are valid (w.r.t. every $u^{(i)}$, $1 \le i \le \mu$), i.e. satisfy the following conditions

$$f[u^{(i)}]_b := \sum_{a \in \mathrm{supp}(f)} c(f,a) u^{(i)}_{a+b-d} = 0,$$
$$d = \mathrm{le}(f) \le_P b < r, \ 1 \le i \le \mu \tag{5}$$

and have a distinct minimal leading exponent $d = \mathrm{le}(f)$ among the valid polynomials s.t. $(\Sigma(F) = \bigcup_{f \in F} \Sigma_{\mathrm{le}(f)}, \Delta(F))$ is a separation of $\mathbf{N}^n$ and there exists no valid polynomial $g \in \mathcal{P}$ with $\mathrm{le}(g) \in \Delta(F)$.

This is a multidimensional extension of the multisequence shift-register synthesis problem treated by Feng and Tzeng (1989, 1991).

(2) **Vectorial BMS problem** (Sakata 1991)

Similarly to the Gröbner basis theory of modules we define the leading exponent, leading position and leading coefficient of a polynomial vector $\mathbf{f} = (f^{(1)}, \ldots, f^{(m)}) \in \mathcal{P}^m (:= (\mathbb{F}[X_1, \ldots, X_n])^m)$ as follows:

$$\mathrm{le}(\mathbf{f}) := \max_<\{\mathrm{le}(f^{(i)}) \in \mathbf{N}^n \mid 1 \le i \le m\}$$
$$\mathrm{lp}(\mathbf{f}) := \max\{i \in [1,m] \mid \mathrm{le}(f^{(i)}) = \mathrm{le}(\mathbf{f})\}$$
$$\mathrm{lc}(\mathbf{f}) := c(f^{(i)}, \mathrm{le}(f^{(i)})) \in \mathbb{K} \quad \text{for } i = \mathrm{lp}(\mathbf{f}),$$

where $[1,m] := \{1, \ldots, m\} \subset \mathbf{N}$, and we use the pair $\mathrm{le}(\mathbf{f}) \in \mathbf{N}^n$, $\mathrm{lp}(f) \in \mathbf{N}$ instead of $\mathbf{T}(f) \in \mathcal{T}^m$ in Mora (2009a).

Given an array vector $\mathbf{u} = (u^{(1)}, \ldots, u^{(m)})$ whose components are finite n-D arrays $u^{(i)}$ over $\mathbb{F}$, $1 \le i \le m$, defined over $\Sigma^r \subset \mathbf{N}^n$ (w.r.t. a fixed term ordering $<$) for a certain point $r \in \mathbf{N}^n$;

Find a minimal polynomial vector set $\mathbf{F}$ of $\mathbf{u}$ which is a union of m subsets $\mathbf{F}^{(i)} (\subset \mathcal{P}^m)$, $1 \le i \le m$, where each $\mathbf{F}^{(i)}$ is composed of polynomial vectors

$\mathbf{f} = (f^{(1)}, \ldots, f^{(m)}) \in \mathcal{P}^m$ with leading position $\mathrm{lp}(\mathbf{f}) = i$, $1 \le i \le m$, which are valid w.r.t. $\mathbf{u}$, i.e. satisfy the following condition:

$$\mathbf{f}[\mathbf{u}]_b := \sum_{i=1}^{m} \sum_{a \in \mathrm{supp}(f^{(i)})} c(f^{(i)}, a) u^{(i)}_{a+b-d} = 0, \quad d = \mathrm{le}(\mathbf{f}) \le_P b < r \tag{6}$$

s.t. there exists no valid polynomial vector $\mathbf{g}$ with $\mathrm{lp}(\mathbf{g}) = i$ and $\mathrm{le}(\mathbf{g}) <_P \mathrm{le}(\mathbf{f})$ for any $\mathbf{f} \in \mathbf{F}^{(i)}$, $1 \le i \le m$.

Naturally this problem can be generalized to finding a minimal polynomial vector set of a given finite set $\mathbf{U} = \{\mathbf{u}^{(1)}, \ldots, \mathbf{u}^{(L)}\}$ of array vectors. For the set $\mathbf{U}$ composed of infinite array vectors, it amounts to find a Gröbner basis of the *characteristic module* of $\mathbf{U}$ which is defined as

$$\mathbf{M}(\mathbf{U}) := \{\mathbf{f} \in \mathcal{P}^m \mid \mathbf{f}[\mathbf{u}^{(l)}]_b = 0, b \in \mathbf{N}^n, 1 \le l \le L\}$$

similarly to the characteristic ideal $\mathbf{I}(U) \subset \mathcal{P}$ of a set U of infinite n-D arrays introduced in Sect. 3 (if it exists).

As far as we know, neither any 1-D version of this problem nor its solution had been published before (Sakata 1991).

(3) **Non-homogeneous BMS problem** (Sakata 2003)

At the beginning of Sect. 3, we introduced the nonhomogeneous BMS problem (2) for a given pair of sets $U := \{u^{(i)} \mid 1 \le i \le \mu\}$ and $V := \{v^{(i)} \mid 1 \le i \le \mu\}$. Now we consider the simplest case of $\mu = 1$ as follows.

Given a pair of finite n-D arrays u, v over $\mathbb{F}$, where u and v are defined over $2\Sigma^r := \{a + b \mid a, b \in \Sigma^r\}$ and Σ^r ($\subset \mathbf{N}^n$), respectively, w.r.t. a fixed term ordering $<$ for a certain point $r \in \mathbf{N}^n$.

Find a set F of polynomials f which are valid w.r.t. the given pair (u, v^r), i.e. satisfy the following condition

$$f\langle u\rangle_b := \sum_{a \in \mathrm{supp}(f)} c(f, a) u_{a+b} = v_b, \quad 0 \le b < r, \tag{7}$$

and have a distinct minimal leading exponent $d = \mathrm{le}(f)$ among the valid polynomials s.t. $(\Sigma(F) = \bigcup_{f \in F} \Sigma_{\mathrm{le}(f)}, \Delta(F))$ is a separation of $\mathbf{N}^n$ and there exists no valid polynomial $g \in \mathcal{P}$ with $\mathrm{le}(g) \in \Delta(F)$.

In addition to $\mathrm{Val}(u^b)$, $\mathrm{mVal}(u^b)$, $\mathrm{Aux}(u; c)$, and $\mathrm{mAux}(u; c)$ introduced before, we define

$$\mathrm{Val}(u; v^b) := \{f \in \mathcal{P} \mid f\langle u\rangle_a = v_a, 0 \le a < b\}$$

$$\mathrm{mVal}(u; v^b) := \{f \in \mathrm{Val}(u; v^b) \mid \mathrm{le}(f) : \text{minimal}\}$$

For the 1-D case Sugiyama (1986) gave a solution based on the Euclidean algorithm.

(4) **Submodule BMS problem** (Sakata 2007)

For a fixed pair $(\bar{\Sigma}, \bar{\Delta})$ of stable subsets of $\mathbf{N}^n$ s.t. $\bar{\Sigma} = \bigcup_{d \in \bar{D}} \Sigma_d$ and $\bar{\Delta} =$

$\mathbf{N}^n \setminus \bar{\Sigma} = \bigcup_{c\in\bar{C}} \Gamma_c$, where $\bar{D}$ and $\bar{C}$ are given a priori consistently, we consider a module $\mathcal{P}(\bar{\Sigma})$ over $\mathcal{P}$ which is defined to be the set of all polynomials f with $\mathrm{supp}(f) \subset \bar{\Sigma}$. In such a specified module $\mathcal{P}(\bar{\Sigma})$ we have $\mathcal{P}$-submodules and their Gröbner bases, and correspondingly several similar concepts extended from the original BMS problem to the present case. Particularly, given a finite array $u^r = (u_a)$, $a < r$ (w.r.t. a term ordering $<$) defined over the subset $\bar{\Sigma}^r = \{a \in \bar{\Sigma} \mid a < r\}$, a polynomial $f \in \mathcal{P}(\bar{\Sigma})$ is said to be *valid* for the array u^r iff the identity of the same form holds as (4). And, a minimal polynomial set $F \subset \mathcal{P}(\bar{\Sigma})$ of u^b, $b \in \bar{\Sigma}$ is defined similarly together with $\Sigma(b) = \bigcup_{d\in D} \Sigma_d \subset \bar{\Sigma}$ and $\Delta(b) = \bar{\Sigma} \setminus \Sigma(b)$, where there exists no valid polynomial g (for u^b) with $\mathrm{le}(g) \in \Delta(b)$ and there exists a valid polynomial f with $\mathrm{le}(f) = d$ for each $d \in D$, etc. In this case we have the following problem:

Given a finite array u^r (defined over $\bar{\Sigma}^r \subset \bar{\Sigma}$);
Find a minimal polynomial set $F \subset \mathcal{P}(\bar{\Sigma})$ of u^r.

We have such a problem in decoding two-point codes from curves (Sakata 2007).

(5) **Semigroup BMS problem** (Sakata 1995)

Instead of the usual integral lattice $\mathbf{N}^n$ and polynomial ring $\mathcal{P}$ we consider a semigroup $\bar{\Sigma}$ of the additive group $\mathbf{Z}^n$ (or an n-D *convex cone* in geometrical terms) and the corresponding ring $\bar{\mathcal{P}}$ which are define by a given unimodular matrix $W = (w_{ij}) \in \mathbf{Z}^{n\times n}$ as follows:

$$\bar{\Sigma} := \{a \in \mathbf{Z}^n \mid aW \in \mathbf{N}^n\}$$

$$\bar{\mathcal{P}} := \left\{ f = \sum_{a\in\mathrm{supp}(f)} c(f,a)X^a \in \mathbb{F}[X_1,\ldots,X_n,X_1^{-1},\ldots,X_n^{-1}] \mid \right.$$
$$\left. \mathrm{supp}(f) \subset \bar{\Sigma} \right\}$$

Over $\bar{\Sigma}$ we have a special partial ordering $<_{\bar{P}}$ as follows:

$$a \le_{\bar{P}} b \quad \Leftrightarrow \quad b - a \in \bar{\Sigma}.$$

We can consider not only ideals of this special ring $\bar{\mathcal{P}}$ and their Gröbner bases (w.r.t. a specified term ordering $<$ over $\bar{\Sigma}$) but also a minimal polynomial set $F(\subset \bar{\mathcal{P}})$ of a (finite or infinite) array u defined over $\bar{\Sigma}$, where any $f \in F$ satisfies the condition:

$$f[u]_b := \sum_{a\in\mathrm{supp}(f)} c(f,a)u_{a+b-d} = 0, \quad d = (f) \le_{\bar{P}} b. \tag{8}$$

Thus we have the present problem:

Given a finite array u^r (defined over $\bar{\Sigma}^r$ $(:= \{a \in \bar{\Sigma} \mid a < r\})$;
Find a minimal polynomial set $F(\subset \bar{\mathcal{P}})$ of u^r.

We have such a problem in decoding codes from Klein curves (Sakata 1995).

In the following subsections we present a series of extended BMS algorithms for these problems, where the validity of these algorithms can be proven similarly to the original BMS algorithm. The theorems about complete class, uniqueness condition, and relevant Gröbner bases also can be given. All of them can be applied to fast decoding of certain algebraic codes (see Sakata 2009).

4.1 Multiarray BMS Algorithm

In this subsection we present the extended BMS algorithm (Sakata 1989) for solving the multiarray BMS problem (5), which is a multidimensional extension of the Feng-Tzeng algorithms (Feng and Tzeng 1989, 1991). As above-mentioned, given a finite set $U = \{u^{(i)} \mid 1 \le i \le \mu\}$ of (finite or infinite) n-D arrays over $\mathbb{F}$, we want to find a minimal polynomial set of U. To discuss this problem, we introduce some additional notation.

For a given pair $(j, r) \in [1, \mu] \times \mathbf{N}^n$, where $[1, \mu] := \{1, \dots, \mu\} \subset \mathbf{N}$, let

$$\begin{aligned}\Sigma^{(j,r)} := &\left(\bigcup_{i<j}\{(i,a) \in [1,\mu] \times \mathbf{N}^n \mid a \in \Sigma^{r\oplus 1}\}\right)\\ &\cup \left(\bigcup_{i\ge j}\{(i,a) \in [1,\mu] \times \mathbf{N}^n \mid a \in \Sigma^{r}\}\right)\end{aligned}$$

and consider the corresponding set of partial arrays defined over $\Sigma^{(j,r)}$

$$U^{(j,r)} := \left(\bigcup_{i<j}\{u^{(i,r\oplus 1)} := (u^{(i)}_a), a \le r\}\right) \cup \left(\bigcup_{i\ge j}\{u^{(i,r)} := (u^{(i)}_a), a < r\}\right)$$

If a polynomial $f = \sum_{a\in \mathrm{supp}(f)} c(f,a)X^a$ satisfies

$$f[u^{(i)}]_b := \sum_{a\in\mathrm{supp}(f)} c(f,a)u^{(i)}_{a+b-d} = 0 \tag{9}$$

for any $(i, b) \in \Sigma^{(j,r)}$ s.t. $d = \mathrm{le}(f) \le_P b$, we say that f is valid (w.r.t. U) before (j, r). As in case of a single array, i.e. $\mu = 1$, we can define a minimal polynomial set $F(j,r)$ (or simply F) $\subset \mathcal{P}$ of $U^{(j,r)}$ for a pair $(j, r) \in [1, \mu] \times \mathbf{N}^n$ associated with a finite set of points $D(j,r) := \{\mathrm{le}(f) \mid f \in F(j,r)\} \subset \mathbf{N}^n$ s.t. there is a single element $f \in F(j,r)$ with $\mathrm{le}(f) = d$ and $\mathrm{lc}(f) = 1$ for each $d \in D(j,r)$ and there exists no polynomial h with $\mathrm{le}(h) \in \Delta(j,r)$ which is valid (w.r.t. U) before (j, r), where we have a complementary pair of subsets $\subset \mathbf{N}^n$:

$$\Sigma(j,r) := \bigcup_{d\in D(j,r)} \Sigma_d, \qquad \Delta(j,r) := \mathbf{N}^n \setminus \Sigma(j,r)$$

We call $\Delta(j,r)$ the *delta-set* of $F(j,r)$. These subsets $D(j,r)$, $\Sigma(j,r)$ and $\Delta(j,r)$ are unique for the given set $U^{(j,r)}$ of partial arrays, but a minimal polynomial set $F(j,r)$ is not necessarily unique for $U^{(j,r)}$. In view of the definition of minimal polynomial set $F(j,r)$, $\Delta(j,r) \subseteq \Delta(j, r \oplus 1)$, $\Delta(j,r) \subseteq \Delta(j+1, r)$, $1 \leq j < \mu$ and $\Delta(\mu, r) \subseteq \Delta(1, r \oplus 1)$. Similar notations can be used for any set of infinite arrays U, e.g. a minimal polynomial set $F(\subset \mathcal{P})$ of U, the delta-set $\Delta(\subset \mathbf{N}^n)$ of U, etc. if they exist. In the present problem we do not have a single auxiliary polynomial set of U, which is associated directly to the minimal polynomial set F of U and its delta-set Δ, but μ distinct auxiliary polynomial sets $G^{(i)} = G^{(i)}(j,r)$ with $C^{(i)} = C^{(i)}(j,r) = \{\text{span}(g) \mid g \in G^{(i)}\} \subset \mathbf{N}^n$ and $\Delta^{(i)} = \Delta^{(i)}(j,r) = \bigcup_{c \in C^{(i)}} \Gamma_c$, $1 \leq i \leq \mu$ s.t. $\#\Delta(j,r) = \sum_{1 \leq i \leq \mu} \#\Delta^{(i)}$. Below we show the multiarray BMS algorithm, which is almost the same as the original BMS algorithm (Algorithm 2) except for appearance of the additional loop w.r.t. j and $C^{(i)}$ (and $G^{(i)}$) $1 \leq i \leq \mu$ instead of C.

Algorithm 3 (Multiarray BMS algorithm) Finding a minimal polynomial set of $U^{(\mu,r)}$ for $U = \{u^{(i)} \mid 1 \leq i \leq \mu\}$ of finite n-D arrays over $\mathbb{F}$ (Sakata 1989);

```
Step 1 (initialization): $j := 1$; $b := 0$; $F := \{1\}$; $D := \{0\}$; $\Sigma := \mathbf{N}^n$;
       $G^{(i)} := \emptyset, 1 \leq i \leq \mu$; $C^{(i)} := \emptyset, 1 \leq i \leq \mu$;
Step 2 (discrepancy): for each $f \in F \quad d_f := f[u]_b$;
       $F_{\text{fail}} := \{f \in F \mid d_f \neq 0\}$;
       $F_{\text{fall}} := \{f \in F_{\text{fail}} \mid \nexists c \in C^{(j)} \text{ s.t. } d \geq_P b - c\}$;
       $D_{\text{fall}} := \{d \in D \mid f \in F_{\text{fall}}\}$; $\hat{D} := \{\text{minimal}\, \hat{d} \in \Sigma \setminus \Gamma_b\}$;
       $D'' := \{\max(d, b-c) \mid d \in D_{\text{fall}}, c \in C^{(j)}\}$;
       $D' := \{\text{minimal}\, d' \in D''\}$;
Step 3 (updating): (1) for each $f \in F_{\text{fail}} \setminus F_{\text{fall}}$
                           begin $h := f - d_f X^{d-(b-c)} g$ (for $g \in G^{(j)}$ s.t. $d \geq_P b - c$);
                              $F' := F \cup \{h\}$ end;
                        for each $(f, g) \in F_{\text{fall}} \times G^{(j)}$ s.t. $d' := \max(d, b-c) \in D'$
                           begin $h := X^{\hat{d}-dd'-d} f - d_f g$; $F' := F \cup \{h\}$ end;
                        for each $\hat{d} \in \hat{D}$
                              if $\nexists d' \in D'$ s.t. $\hat{d} \geq_P d'$ then for $f \in F_{\text{fall}}$ s.t. $d \leq_P \hat{d}$
                                 begin $h := X^{\hat{d}-d} f$; $F' := F \cup \{h\}$ end;
                   (2) $F := F' \setminus F_{\text{fail}}$; $G'' := \{g \in G^{(j)} \mid \exists f \in F_{\text{fall}} \text{ s.t. } c <_P b - d\}$;
                        $G^{(j)} := (G^{(j)} \cup \{\frac{1}{d_f} f \mid f \in F_{\text{fall}}\}) \setminus G''$;
                        $D := \{\text{le}(f) \mid f \in F' \setminus F_{\text{fall}}\}$; $\Sigma := \bigcup_{d \in D} \Sigma_d$;
                        $C^{(j)} := (C^{(j)} \cup \{b - d \mid \exists f \in F_{\text{fall}} \text{ s.t. } b - d >_P c\})$
                                    $\setminus \{c \in C^{(j)} \mid \exists f \in F_{\text{fall}} \text{ s.t. } b - d >_P c\}$;
Step 4 (termination): $j := j + 1$: if $j \leq \mu$ then go to Step 2
                        else begin $j := 1$; $b := b \oplus 1$;
                   if $b < r$ then go to Step 2 else stop.
```

4.2 Vectorial BMS Algorithm

In this subsection we present the vectorial BMS algorithm (Sakata 1991) for solving the vectorial BMS problem (6), for which the special 1D case of the vectorial BMS problem had not been treated and the vectorial BM algorithm had not given before.

As above-mentioned, we are given an array vector $\mathbf{u} \in \mathcal{A}^m$, $\mathbf{u} = (u^{(1)}, \ldots, u^{(m)})$, whose components are n-D arrays $u^{(j)}$ over $\mathbb{F}$, $1 \le j \le m$, defined over $\Sigma^r \subset \mathbf{N}^n$ (w.r.t. a certain term ordering $<$) for a fixed point $r \in \mathbf{N}^n$;

The following is the vectorial BMS algorithm which finds m minimal polynomial vector sets $\mathbf{F}^{(i)}(b)$, $1 \le i \le m$ of the partial array vector $\mathbf{u}^b$ as well as an auxiliary polynomial vector set $\mathbf{G}(b)$ at each point $b \in \Sigma^r$ iteratively w.r.t. the term ordering $<$, where all the elements $\mathbf{f}$ of $\mathbf{F}^{(i)}(b)$ have $\mathrm{lp}(\mathbf{f}) = i$, $1 \le i \le m$. $\mathbf{F}^{(i)}(b)$, $1 \le i \le m$ and $\mathbf{G}(b)$ are denoted simply as $\mathbf{F}^{(i)}$, $1 \le i \le m$ and $\mathbf{G}$, respectively. During the execution of the algorithm, we have m delta-set $\Delta^{(i)} = \Sigma \setminus \Sigma^{(i)}$, $1 \le i \le m$ associated to $\mathbf{F}^{(i)}$ with $D^{(i)} = \{d = \mathrm{le}(\mathbf{f}) \mid \mathbf{f} \in \mathbf{F}^{(i)}\}$ s.t. $\Sigma^{(i)} = \bigcup_{d \in D^{(i)}} \Sigma_d$, $1 \le i \le m$, and an auxiliary polynomial vector set $\mathbf{G} \in \mathcal{P}^m$, to which a subset $C = \{c = \mathrm{span}(\mathbf{g}) \mid \mathbf{g} \in \mathbf{G}\} (\subset \mathbf{N}^n)$ is associated. Every initial $\mathbf{F}^{(i)}$ is composed of a singleton $\mathbf{e}_i = (0, \ldots, 0, 1, 0, \ldots, 0) \in \mathcal{P}^m$, i.e. the i-th unit vector, $1 \le i \le m$.

Algorithm 4 (Vectorial BMS algorithm) Finding minimal polynomial vector sets of a finite n-D array vector $\mathbf{u}^b \in \mathcal{A}^m$ over $\mathbb{F}$ (Sakata 1991);

```
Step 1  (initialization): $j := 1$; $b := 0$; $\mathbf{F}^{(i)} := \{\mathbf{e}_i\}$, $1 \le i \le m$; $\mathbf{G} := \emptyset$;
        $D^{(i)} := \{0\}\ (\subset \mathbf{N}^n)$, $1 \le i \le m$; $\Sigma^{(i)} := \mathbf{N}^n$, $1 \le i \le m$; $C := \emptyset$;
Step 2  (discrepancy): for each $\mathbf{f} \in \mathbf{F}^{(j)}$ $d_{\mathbf{f}} := \mathbf{f}[\mathbf{u}]_b$;
        $\mathbf{F}^{(j)}_{\mathrm{fail}} := \{\mathbf{f} \in \mathbf{F}^{(j)} \mid d_{\mathbf{f}} \neq 0\}$;
        $\mathbf{F}^{(j)}_{\mathrm{fall}} := \{\mathbf{f} \in \mathbf{F}^{(j)}_{\mathrm{fail}} \mid \nexists c \in C \text{ s.t. } d \ge_P b - c\ \}$;
        $D^{(j)}_{\mathrm{fall}} := \{d = \mathrm{le}(\mathbf{f}) \in D^{(j)} \mid \mathbf{f} \in \mathbf{F}^{(j)}_{\mathrm{fall}}\}$; $\hat{D} := \{\text{minimal}\, \hat{d} \in \Sigma \setminus \Gamma_b\}$;
        $D''^{(j)} := \{\max(d, b - c) \mid d \in D^{(j)}_{\mathrm{fall}},\ c \in C\}$;
        $D'^{(j)} := \{\text{minimal}\, d' \in D''^{(j)}\}$;
Step 3  (updating): (1) for each $\mathbf{f} \in \mathbf{F}^{(j)}_{\mathrm{fail}} \setminus \mathbf{F}^{(j)}_{\mathrm{fall}}$
                        begin $\mathbf{h} := \mathbf{f} - d_{\mathbf{f}} X^{d-(b-c)} \mathbf{g}$ (for $\mathbf{g} \in \mathbf{G}$ s.t. $d \ge_P b - c$);
                          $\mathbf{F}'^{(j)} := \mathbf{F}^{(j)} \cup \{\mathbf{h}\}$ end;
                    for each $(\mathbf{f}, \mathbf{g}) \in \mathbf{F}^{(j)}_{\mathrm{fall}} \times \mathbf{G}$ s.t. $d' := \max(d, b - c) \in D'^{(j)}$
                        begin $\mathbf{h} := X^{d'-d}\mathbf{f} - d_{\mathbf{f}}\mathbf{g}$; $\mathbf{F}'^{(j)} := \mathbf{F}^{(j)} \cup \{\mathbf{h}\}$ end;
                    for each $\hat{d} \in \hat{D}$ if $\nexists d' \in D'^{(j)}$ s.t. $\hat{d} \ge_P d'$ then
                        for $\mathbf{f} \in \mathbf{F}^{(j)}_{\mathrm{fall}}$ s.t. $d \le_P \hat{d}$
                          begin $\mathbf{h} := X^{\hat{d}-d}\mathbf{f}$; $\mathbf{F}'^{(j)} := \mathbf{F}^{(j)} \cup \{\mathbf{h}\}$ end;
                (2) $\mathbf{F}^{(j)} := \mathbf{F}'^{(j)} \setminus \mathbf{F}^{(j)}_{\mathrm{fail}}$; $\mathbf{G}'' := \{\mathbf{g} \in \mathbf{G} \mid \exists \mathbf{f} \in \mathbf{F}^{(j)}_{\mathrm{fall}} \text{ s.t. } c <_P b - d\}$;
                    $\mathbf{G} := (\mathbf{G} \cup \{\frac{1}{d_{\mathbf{f}}}\mathbf{f} \mid \mathbf{f} \in \mathbf{F}^{(j)}_{\mathrm{fall}}\}) \setminus \mathbf{G}''$;
                    $D^{(j)} := \{\mathrm{le}(\mathbf{f}) \mid \mathbf{f} \in \mathbf{F}'^{(j)} \setminus \mathbf{F}^{(j)}_{\mathrm{fall}}\}$; $\Sigma^{(j)} := \bigcup_{d \in D^{(j)}} \Sigma_d$;
                    $C := (C \cup \{b - d \mid \exists \mathbf{f} \in \mathbf{F}^{(j)}_{\mathrm{fall}} \text{ s.t. } b - d >_P c\})$
                            $\setminus \{c \in C \mid \exists \mathbf{f} \in \mathbf{F}^{(j)}_{\mathrm{fall}} \text{ s.t. } b - d >_P c\}$;
Step 4  (termination): $j := j + 1$: if $j \le m$ then go to Step 2
                    else begin $j := 1$; $b := b \oplus 1$;
                if $b < r$ then go to Step 2 else stop.
```

4.3 Non-Homogeneous BMS Algorithm

This is a two-path algorithm, the first path of which is just the (homogeneous) BMS algorithm for the given array u^{2r}. As a byproduct of the BMS algorithm, we get a series of minimal auxiliary polynomial sets $\mathrm{mAux}(u; c)$ for each $c \in \Sigma^r$ based on the following two lemmas.

Lemma 3 *Let* $h \in \mathrm{Val}(u^b)$ *and* $h[u]_b \neq 0$ (*thus*, $\mathrm{span}(h) = b - \mathrm{le}(h)$). *If* $\mathrm{span}(h) = c \notin \Delta(u^b)$, *then* $h \in \mathrm{mAux}(u; c)$.

Lemma 4 *Let* $\Delta(u^b) = \Delta(u^{b\oplus 1}) = \cdots = \Delta(u^{b\oplus l}) \subset \Delta(u^{b\oplus(l+1)})$ *and* $h \in \mathrm{mVal}\Delta(u^{b\oplus l})$, $h[u]_{b\oplus l} \neq 0$, *where* $b \oplus (l+1) := (b \oplus l) \oplus 1$, $1 \leq l$ $(\in Z_0)$. *Then*, $h \in \mathrm{mAux}(u; c)$ *for* $c := \mathrm{span}(h)$ $(= b \oplus l - \deg(h))$, *and furthermore*, $X^a h \in \mathrm{mAux}(u; c - a)$ *for* $0 \leq a <_P c - c'$, *provided that there exists* $h' \in \mathrm{mVal}(u^{b\oplus l})$ *with* $\deg(h') = c' \leq_P c$.

First, by applying the BMS algorithm, we get a series of auxiliary polynomials g^c for several intermittent points $c = c_1, c_2, \ldots, c_\lambda$ s.t. $0 \leq c_1 < c_2 < \cdots < c_\lambda < r$ and $c_i = \mathrm{span}(g^{(c_i)})$, $1 \leq i \leq \lambda$, where λ is an integer determined by execution of BM algorithm. Furthermore, by Lemmas 3, 4, we can have a certain delta set $\bar{\Delta}(= \bigcup_{1\leq i\leq\lambda} \Gamma_{c_i})$ and an auxiliary polynomial $h^{(c)}$ for each point $c \in \bar{\Delta}$ s.t. $\mathrm{span}(h^{(c)}) = c$, $c \in \bar{\Delta}$. The following lemma leads us to have a fast algorithm of solving the nonhomogeneous BMS problem.

Lemma 5 *Let* $h \in \mathrm{mAux}(u; c)$, $f \in \mathrm{mVal}(u; v^c)$ *and* $\mathrm{le}(f) < \mathrm{le}(h)$. *If* $f \notin \mathrm{Val}(u; v^{c\oplus 1})$, *then there is no polynomial* $f' \in \mathrm{Val}(u; v^{c\oplus 1})$ *s.t.* $\mathrm{le}(f') < \mathrm{le}(h)$.

Algorithm 5 (Nonhomogeneous BMS algorithm) Finding $f \in \mathrm{mVal}(u; v^r)$ for $u = (u_a)$, $a \in \Sigma^{2r}$ and $v = (v_a)$, $a \in \Sigma^r$ (Sakata 2003);

```
Step 1: $b := 0$; $f := 1$; $d := 0$;
Step 2: If $f\langle u\rangle_b \neq v_b$ then
            begin $h := h^{(b)}$; $f := f + \frac{1}{d_b}(v_b - f\langle u\rangle_b)h$; $d := \mathrm{le}(h)$ end;
Step 3: $b := b \oplus 1$; if $b < r$ then go to Step 2 else stop.
```

If $b \notin \bar{\Delta}$ in Step 2, then Algorithm 5 halts, which implies that (7) has no solution.

4.4 Submodule BMS Algorithm

To solve this problem, we can have a modification (Sakata 2007) of BMS algorithm which is obtained by modifying Step 1 (initialization) as follows:

Step 1 (initialization): $b := \min_T \bar{D}$; $F := \{X^d \mid d \in \bar{D}\}$; $G := \emptyset : C := \emptyset$;

The validity of the algorithm can be proven similarly to the original BMS algorithm. Particularly, it holds during the whole iterations of the algorithm that for $\Delta(F) := \bar{\Sigma} \setminus \Sigma(F)$, where $\Sigma(F) = \Sigma(b)$, $\Delta(F) = \Delta(b)$ $(\subset \bar{\Sigma})$ for a minimal polynomial set F and an auxiliary polynomial set G of u^b, and $\Delta(G) := \{a \in \mathbf{N}^n \mid a \leq_P \mathrm{span}(g),\ g \in G\}$ $(\subset \mathbf{N}^n)$, it holds that $\#\Delta(F) = \#\Delta(G)$, although $\Delta(F) \neq \Delta(G)$.

4.5 Semigroup BMS Algorithm

In this case we also have a specific term ordering over $\bar{\Sigma}$, and we have a version (Sakata 1995) of the BMS algorithm, which is obtained by replacing every partial ordering $\leq_P$ by $\leq_{\bar{P}}$ in the descriptions of the original BMS algorithm.

5 Conclusion

First we have discussed that the BMS algorithm (Sakata 1988, 1990) is related to Gröbner basis via multidimensional arrays and multidimensional linear recurrences satisfied by them, and that it can solve just the inverse problem of that of the Buchberger algorithm. Second, we have presented the essence of the BMS algorithm which outputs a minimal polynomial set of a given finite n-D array. Then, we have given theorems about the complete class of minimal polynomial sets, uniqueness condition and a Gröbner basis of the ideal $\mathbf{I}(u)$ defined by an infinite array u, and discussed its computational complexity. Furthermore, we presented various extensions of the original BMS problem and their algorithms (Sakata 1989, 1991, 1995, 2003, 2007) which solve these problems. Those extended algorithms are useful for decoding several algebraic codes efficiently (Sakata 2009).

Acknowledgements Part of these results have been presented at Linz D1 2006, which was a workshop within the Special Semester on Gröbner Bases, February–July 2006, organized by RICAM, Austrian Academy of Sciences, and RISC, Johannes Kepler University, Linz, Austria.

Appendix A: Computation of BMS Algorithm

Example of Computation

In Table 3 is shown a result of computations by BMS algorithm applied to the 2-D array shown in Table 2, where we take the graded reverse lexicographic ordering as the term ordering $<$. The symbol $\star$ implies an updated polynomial. We can make sure by the Buchberger criterion that the minimal polynomial set obtained at the final iteration is a Gröbner basis. (*Remark*: A minimal polynomial set is not necessarily a Gröbner basis.)

Table 2 2-D array over $\mathbf{F}_2$: $u = (u_{ij})$

$i \setminus j$	0	1	2	3	4
0	0	1	0	0	1
1	1	1	1	1	
2	1	0	0		
3	1	0			
4	0	*			
5	1				

Table 3 Computations by BMS algorithm

b	F	D	G	C
(0, 0)	1	(0, 0)	–	–
(1, 0)	·		·	
(0, 1)	$\star x^2$	(2, 0)	$\star 1$	(1, 0)
	y	(0, 1)		
(2, 0)	x^2	(2, 0)	1	(1, 0)
	$\star y + x$	(0, 1)		
(1, 1)	$\star x^2 + x$	(2, 0)	1	(1, 0)
	$y + x$	(0, 1)		
(0, 2)	·		·	
(3, 0)	$x^2 + x$	(2, 0)	1	(1, 0)
	$\star xy + x^2$	(1, 1)	$\star y + x$	(0, 1)
	$\star y^2 + xy + x$	(0, 2)		
(2, 1)	·		·	
(1, 2)	$x^2 + y$	(2, 0)	1	(1, 0)
	$\star xy + x^2 + 1$	(1, 1)	$y + x$	(0, 1)
	$y^2 + xy + x$	(0, 2)		
(0, 3)	·		·	
(4, 0)	·		·	
(3, 1)	·		·	
(2, 2)	$\star x^3 + xy + y + x$	(3, 0)	$\star xy + x^2 + 1$	(2, 0)
	$\star x^2y + x^2 + x + 1$	(2, 1)	$\star x^2 + y$	(1, 1)
	$y^2 + xy + x$	(0, 2)	$y + x$	(0, 1)
(1, 3)	·		·	
(0, 4)	$x^3 + xy + y + x$	(3, 0)	$xy + x^2 + 1$	(2, 0)
	$x^2y + x^2 + x + 1$	(2, 1)	$x^2 + y$	(1, 1)
	$\star y^2 + x^2 + x + 1$	(0, 2)	$y + x$	(0, 1)
(5, 0)	·		·	
(4, 1)	*			

References

E. R. Berlekamp, *Algebraic coding theory*, McGraw–Hill, New York, 1968.

B. Buchberger, *Ein Algorithmus zum Auffinden der Basiselemente des Restklassenringes nach einem nulldimensionalen Polynomideal*, Ph.D. thesis, Innsbruck, 1965.

B. Buchberger, *Gröbner-bases: An algorithmic method in polynomial ideal theory*, Multidimensional systems theory, Reidel, Dordrecht, 1985, pp. 184–232.

B. Buchberger, *Bruno Buchberger's PhD thesis 1965: An algorithm for finding the basis elements of the residue class ring of a zero dimensional polynomial ideal*, J. Symb. Comput. **41** (2006), nos. 3–4, 475–511.

G. L. Feng and K. K. Tzeng, *A generalized Euclidean algorithm for multisequence shift-register synthesis*, IEEE Trans. on Inf. Th. **35** (1989), no. 3, 584–594.

G. L. Feng and K. K. Tzeng, *Decoding cyclic and BCH codes up to actual minimum distance using nonrecurrent syndrome dependence relations*, IEEE Trans. on Inf. Th. **37** (1991), no. 6, 1716–1723.

T. Ikai, H. Kosako and Y. Kojima, *Basic theory of two-dimensional cyclic codes – generator polynomials and the position of check symbols*, IEICE Trans. Fundamentals **J59-A** (1976), 311–318.

J. L. Massey, *Shift-register synthesis and BCH decoding*, IEEE Trans. on Inf. Th. **15** (1969), 122–127.

T. Mora, *The FGLM problem and Möller's algorithm on zero-dimensional ideals*, this volume, 2009a, pp. 27–45.

T. Mora, *Gröbner technology*, this volume, 2009b, pp. 11–25.

S. Sakata, *General theory of doubly periodic arrays over an arbitrary finite field and its applications*, IEEE Trans. on Inf. Th. **24** (1978), no. 6, 719–730.

S. Sakata, *On determining the independent point set for doubly periodic arrays and encoding two-dimensional cyclic codes and their duals*, IEEE Trans. on Inf. Th. **27** (1981), no. 5, 556–565.

S. Sakata, *Finding a minimal set of linear recurring relations capable of generating a given finite two-dimensional array*, J. Symbolic Comput. **5** (1988), no. 3, 321–337.

S. Sakata, *n-dimensional Berlekamp–Massey algorithm for multiple arrays and construction of multivariate polynomials with preassigned zeros*, LNCS, vol. **357**, 1989, Springer, Berlin, pp. 356–376.

S. Sakata, *Extension of the Berlekamp–Massey algorithm to N dimensions*, Inform. and Comput. **84** (1990), no. 2, 207–239.

S. Sakata, *Finding a minimal polynomial vector set of a vector of nD arrays*, LNCS, vol. **539**, Springer, Berlin, 1991, pp. 414–425.

S. Sakata, *Shift register synthesis on convex cones and cylinders and fast decoding of general one-point AG codes*, Bull. Univ. Electro-Comm. **8** (1995), no. 2, 187–203.

S. Sakata, *Efficient factorization methods for list decoding of code from curves*, Proc. of ISIT 2003, 2003, pp. 363–363.

S. Sakata, *Fast decoding of two-point Hermitian codes*, preprint, 2007.

S. Sakata, *The BMS algorithm and decoding of AG codes*, this volume, 2009, pp. 165–185.

S. Sakata, H. E. Jensen and T. Høholdt, *Generalized Berlekamp–Massey decoding of algebraic-geometric codes up to half the Feng-Rao bound*, IEEE Trans. on Inf. Th. **41** (1995), no. 6, 1762–1768, part 1.

Y. Sugiyama, *An algorithm for solving discrete-time Wiener-Hopf equations based upon Euclid's algorithm*, IEEE Trans. on Inf. Th. **32** (1986), no. 3, 394–409.