Algorithms for Computing Sparse Shifts for Multivariate Polynomials

Dima Yu. Grigoriev

IRMAR Université Rennes-1
Campus Beaulieu 35042 Rennes France

Lakshman Y. N.[1]

Computing Science Research
Bell Labs., 600 Mountain Hill
Murray Hill, NJ 07974, USA

e-mail: `dima@maths.univ-rennes1.fr`, `ynl@research.bell-labs.com`

Abstract

In this paper, we investigate the problem of finding t-sparse shifts for multivariate polynomials. Given a polynomial $f \in \mathcal{F}[x_1, x_2, \ldots, x_n]$ of degree d, and a positive integer t, we consider the problem of representing $f(x)$ as a $\mathcal{K}$-linear combination of the power products of u_i where $u_i = x_i - b_i$ for some $b_i \in \mathcal{K}$, an extension of $\mathcal{F}$, for $i = 1, \ldots, n$, i.e., $f = \sum_j \mathrm{F}_j u^{\alpha_j}$, in which *at most* t of the F_j are non-zero. We provide sufficient conditions for uniqueness of sparse shifts for multivariate polynomials, prove tight bounds on the degree of the polynomial being interpolated in terms of the sparsity bound t and a bound on the size of the coefficients of the polynomial in the standard representation, and describe two new efficient algorithms for computing sparse shifts for a multivariate polynomial.

[1]Work supported by NSF grant CCR–9203062

Introduction

In this paper, we consider the problem of computing t-sparse shifts for multivariate polynomials. Given a polynomial $f \in \mathcal{F}[x_1, x_2, \ldots, x_n]$ of degree d (where $\mathcal{F}$ is a field of characteristic 0), consider the representation of $f(x)$ as a $\mathcal{K}$-linear combination of the power products of u_i where $u_i = x_i - b_i$ for some $b_i \in \mathcal{K}$, an extension of $\mathcal{F}$, for $i = 1, \ldots, n$, i.e., $f = \sum_j \mathrm{F}_j u^{\alpha_j}$ where α_j denotes the multi-index $(\alpha_{j1}, \alpha_{j2}, \ldots, \alpha_{jn})$, and u^{α_j} indicates the power product $u_1^{\alpha_{j1}} u_2^{\alpha_{j2}} \ldots u_n^{\alpha_{jn}}$. Let t be a positive integer $\leq \binom{d+n}{n}$. We say that $\vec{b} = (b_1, b_2, \ldots, b_n)$ is a t-sparse shift for f (or, f is t-sparse in the shifted basis consisting of the power products of the u_i) if *at most* t of the F_j in the above representation are non-zero (the term "basis" refers to the fact that the power products of the u_i form a $\mathcal{F}$-basis for the polynomial ring $\mathcal{F}[x_1, x_2, \ldots, x_n]$).

The main problem that we address is: given an f and t as above, can we efficiently compute a t-sparse shift for f if one exists? We are particularly interested of in the case polynomials that have rational shifts (each $b_i \in \mathcal{F}$) and the case of polynomials that have finitely many t-sparse shifts.

Recently, there has been much interest in the design of efficient algorithms for computing sparse representations for various classes of functions such as polynomials, rational functions, and algebraic functions (Grigoriev-Karpinski 1987, Clausen et al 1988, Ben-Or&Tiwari 1988, Kaltofen-Lakshman 1988, Borodin-Tiwari 1990, Grigoriev-Karpinski-Singer 1990,1991,1992, 1993, 1994, Mansour 1992, Lakshman-Saunders 1993, 1994). The problem of finding sparsifying invertible linear tranformations for polynomials in $\mathcal{F}[x_1, x_2, \ldots, x_n]$ was first addressed in a recent paper by Grigoriev and Karpinski (Grigoriev-Karpinski 1992) where they provide an algebraic criterion to be satisfied by any sparsifying linear tranformation for a polynomial in $\mathcal{F}[x_1, x_2, \ldots, x_n]$ and an algorithm based on the algebraic criterion for computing sparsifying linear transformations. However, the algorithm requires solving systems of polynomial equations and inequalities involving the parameters $a_{i,j}, c_i$ of the sparsifying transformations over the algebraic closure of $\mathcal{F}$. While this is possible in principle, it is known to be very hard. For the important special case considered in this paper, we show that one can compute a system of polynomial equations involving the parameters of the unknown transformation which is already "solved" in a sense (the parameters are separated) in time that is polynomially bounded by t. The dependence of the algorithm on d, n is sensitive to how the polynomial f is presented. We will state the precise complexity result a little later. In this paper, we build on the results of two earlier papers, (Grigoriev-Karpinski 1993, Lakshman-Saunders 1994), and we make use of techniques used to deal with zero-dimensional Gröbner bases.

We assume that we are given a straight line program for computing f. Consequently, we can generate straight line programs for some low order partial derivatives of f efficiently (see Baur and Strassen, 1983). Instead of a straight-line program for f, it is enough if we have black boxes for f and certain low order partial derivatives of f. It is indeed possible to use this approach even if we have just a black box for computing f. We make some remarks later on as to how to modify our approach to work in this situation.

The main contributions of this paper are:

- sufficient conditions for uniqueness of sparse shifts for multivariate polynomials;

- tight bounds on the degree of the polynomial being interpolated in terms of the sparsity bound t and a bound M on the size of the coefficients of the polynomial in the standard representation.
- two new efficient algorithms for computing sparse shifts for a multivariate polynomial.

In section 1 we discuss conditions under which a polynomial can have a unique sparse shift. In section 2, we describe our first algorithm for computing sparse shifts. In section 3, we describe our degree bounds and the second algorithm for computing sparse shifts. The first algorithm computes t-sparse shifts for a multivariate polynomial f with finitely many t-sparse shifts in all cases except one – it can fail when $\deg_{x_i}(f) < t$ for two or more x_i and f still has finitely many sparse shifts. It performs $(dt)^{O(n)}$ $\mathcal{Q}$-operations if randomization is not allowed and $t^{O(n)}$ $\mathcal{Q}$-operations if randomization is allowed. When there is a unique shift, the algorithm performs $(td^n)^{O(1)}$ $\mathcal{Q}$-operations if randomization is not allowed and $(nt)^{O(1)}$ $\mathcal{Q}$-operations if randomization is allowed. The second algorithm computes t-sparse shifts for a multivariate polynomial f without any finiteness restrictions on the number of t-sparse shifts. It has running time bounded by $(nt)^{O(n^2)}$. We conclude with a discussion of open problems and possible applications in section 4.

Actually, the algorithms could run when $\mathcal{F}$-operations are admitted, but for the complexity analysis to make the algorithms more realistic we allow just $\mathcal{Q}$-operations.

1. Observations on the Uniqueness of Sparse Shifts

In (Lakshman-Saunders 1994), the following were shown to be true.

Theorem 1 *Let $f(x) \in \mathcal{F}[x]$ be of degree d and let $t \leq (d+1)/2$. If there exists an α in some algebraic extension $\mathcal{K}$ of $\mathcal{F}$ such that $f(x)$ is t-sparse in the shifted power basis $1, (x-\alpha), (x-\alpha)^2, \ldots,$ then the shift α is unique.* □

Corollary 1 *Let $f(x) \in \mathcal{F}[x]$ be of degree d and let $t \leq (d+1)/2$. If α (in any extension $\mathcal{K}$ of $\mathcal{F}$) is a t-sparse shift (hence the unique t-sparse shift) for $f(x)$, then $\alpha \in \mathcal{F}$.* □

The situation is more complicated than this for the multivariate case. The following example illustrates one difficulty. Consider the polynomial $h(x, y) = x^d y + x^d$ with $d > 4$. h is 2-sparse with respect to the shift $(0, c)$ for any $c \in \mathcal{F}$ $(h(x, y) = x^d(y - c) + (c + 1)x^d)$. Obvious generalizations of this example to polynomials in $\mathcal{F}[x_1, x_2, \ldots, x_n]$ lead us to the following conclusion. Let $h \in \mathcal{F}[x_1, x_2, \ldots, x_n]$ and

$$B = \{\vec{b} = (b_1, b_2, \ldots, b_n) \in \bar{\mathcal{F}}^n \text{ such that } \vec{b} \text{ is a } t\text{-sparse shift for } h\}.$$

Note that if $\vec{b}$ is a t-sparse shift for h, then $\vec{b}$ is a common zero of at least some $\binom{d+n}{n} - t$ partial derivatives of h. It follows that B is an algebraic set in $\bar{\mathcal{F}}^n$ and can be of any dimension from -1 to $n - 1$ (as usually we agree that the empty set has the dimension -1).

The next two weaker uniqueness results follow from theorem 1. Let $h \in \mathcal{F}[x_1, x_2, \ldots, x_n]$ with $\deg_{x_i}(h) = d_i$ and let $t < (d_i + 1)/2$.

Lemma 1 *If $\vec{b} = (b_1, b_2, \ldots, b_n)$ and $\vec{c} = (c_1, c_2, \ldots, c_n)$ are two t-sparse shifts for h, then $b_i = c_i$ for any $1 \leq i \leq n$.*

Proof: For a fixed i consider a mapping $\Phi : \mathcal{F}[x_1, x_2, \ldots, x_n] \mapsto \mathcal{F}[x_i]$ with $\Phi(x_j) = a_j$ for $i \neq j$ where $a_j \in \mathcal{F}$ are chosen to preserve the degree in x_i, i.e., $\deg_{x_i}(h) = d_i = \deg(\Phi(h))$. Both b_i and c_i are t-sparse shifts for $\Phi(h)$ and using theorem 1, we conclude that $b_i = c_i$. $\square$

Lemma 2 *Let $d = \min_{1 \leq i \leq n}\{\deg_{x_i}(h)\}$. If $t < (d+1)/2$, then h has at most one t-sparse shift.*

Proof: Apply lemma 1 to each x_i. $\square$

Stronger uniqueness results hold for the more general case of sparsifying linear transformations under side conditions.

1.1 Sufficient conditions for Uniqueness

In this subsection, we prove two different sufficient conditions for the uniqueness of sparsifying transformations. Let

$$f = \sum_{i=1}^{t} \mathrm{F}_i x^{\alpha_i} = \sum_{i=1}^{T} \phi_i u^{\beta_i}, \quad \vec{u} = A\vec{x} + \vec{d}$$

where

$$\vec{u} = \begin{pmatrix} u_1 \\ u_2 \\ \cdots \\ u_n \end{pmatrix}, \qquad A = (a_{i,j})_{1 \leq i,j \leq n}, \qquad \vec{x} = \begin{pmatrix} x_1 \\ x_2 \\ \cdots \\ x_n \end{pmatrix}, \qquad \vec{d} = \begin{pmatrix} d_1 \\ d_2 \\ \cdots \\ d_n \end{pmatrix},$$

$a_{i,j}, d_i \in \mathcal{K}$ and A non-singular. Let $\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n$ denote the columns of A with $\mathcal{A}$ denoting the set $\{\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n\}$. Let d be the degree of f.

Theorem 2 *If $\vec{d}$ is not in the span of B, for every proper subset B of $\mathcal{A}$, then, $t + T > d$.*

Proof: Let us substitute linear forms in a new variable ζ for the x_i as

$$x_i \longrightarrow k_i(\zeta + N)$$

with $k_i, N \in \mathcal{K}$ to be chosen in a certain way. Clearly, the linear forms u_i become linear forms in ζ under this substitution. Let us try to choose k_i, N is such a way that each of the u_i-s becomes a scalar multiple of one and the same linear form in ζ, different from $\zeta + N$. The u_i are tranformed as follows:

$$\begin{pmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n,1} & a_{n,2} & \cdots & a_{n,n} \end{pmatrix} \begin{pmatrix} k_1(\zeta + N) \\ k_2(\zeta + N) \\ \vdots \\ k_n(\zeta + N) \end{pmatrix} + \begin{pmatrix} d_1 \\ d_2 \\ \vdots \\ d_n \end{pmatrix} = \begin{pmatrix} \mathbf{r}_1 \\ \mathbf{r}_2 \\ \vdots \\ \mathbf{r}_n \end{pmatrix} \zeta + \begin{pmatrix} N\mathbf{r}_1 + d_1 \\ N\mathbf{r}_2 + d_2 \\ \vdots \\ N\mathbf{r}_n + d_n \end{pmatrix}$$

where $\mathbf{r}_i = \sum_{j=1}^{n} a_{i,j} k_j$. Let

$$\vec{r} = \begin{pmatrix} \mathbf{r}_1 \\ \mathbf{r}_2 \\ \dots \\ \mathbf{r}_n \end{pmatrix}, \quad \vec{k} = \begin{pmatrix} k_1 \\ k_2 \\ \dots \\ k_n \end{pmatrix}.$$

We want each $\mathbf{r}_i\zeta + (\mathbf{r}_i N + d_i)$ to be a scalar multiple of one and the same linear form in ζ which means we want $\lambda\vec{r} = N\vec{r} + \vec{d}$ for some non-zero $\lambda \in \mathcal{K}$. So, we have $(\lambda - N)\vec{r} = \vec{d}$. Recalling that $\vec{r} = A\vec{k}$, we have $\vec{k} = 1/(\lambda - N)A^{-1}\vec{d}$. Let us choose λ, N such that $\lambda, N \neq 0$ and $\lambda \neq N$ and find the corresponding k_i by solving the above system of equations. It follows from the hypothesis (that $\vec{d}$ is not in the span of any proper subset of $\mathcal{A}$) that no k_i is zero. This implies that under the substitution considered, the degree does not decrease, i.e., $\deg(f) = \deg(\Phi(\zeta))$ where $\Phi(\zeta) = f(x_i = k_i(\zeta + N))$. Now, $\Phi(\zeta)$ has degree d, is t-sparse in the powers of $\zeta + N$ and T-sparse in the powers of $\zeta + \lambda$. Invoking the univariate theorem, we have $t + T > d$. $\square$

For the special case of sparse shifts, we get the following:

Corollary 2 *Let $f = \sum_{j=1}^{t} \mathrm{F}_j x^{\alpha_j} = \sum_{j=1}^{T} \phi_j u^{\beta_j}$ where $u_i = x_i + d_i$. If no $d_i = 0$, then $t + T > d$.*

Proof: Let A be the identity matrix in the above theorem. $\square$

The second theorem on the number of sparse shifts imposes a different criterion. Let $B = A^{-1} = (b_{i,j})_{1 \leq i,j \leq n}$. We have $\vec{x} = B\vec{u} + \vec{d'}$ where $\vec{d'} = -B\vec{d}$. Note that

$$x_i \partial/\partial x_i (u_1^{s_1} u_2^{s_2} \dots u_n^{s_n}) = u_1^{s_1} u_2^{s_2} \dots u_n^{s_n} (a_{1,i}s_1/u_1 + a_{2,i}s_2/u_2 + \dots + a_{n,i}s_n/u_n)(\sum_{j=1}^{n} b_{i,j}u_j) + \phi$$

where $\deg_u(\phi) < s_1 + s_2 + \dots + s_n$. Let d be the degree of f.

Theorem 3 *If for some i, $\prod_{j=1}^{n} a_{j,i} \prod_{j=1}^{n} b_{i,j} \neq 0$, then $tT > d/n$ for $n \geq 0$.*

Assume the contrary. We set up some notation and prove supporting claims and then present the proof of the theorem. Let $\mathsf{D} = x_i\partial/\partial x_i$. There exist $\epsilon_0, \epsilon_1, \dots, \epsilon_t \in \mathcal{F}$ such that $\sum_{0 \leq j \leq t} \epsilon_j \mathsf{D}^j f = 0$ (see lemma 4). Let $t_0 = \max\{j : \ \epsilon_j \neq 0\} \leq t$.

Let $S_n(d) \subset \mathcal{Z}^n$ denote the set of integer points of the $(n-1)$-dimensional simplex $\{(z_1, z_2, \dots, z_n) : z_i \geq 0, \ 1 \leq i \leq n, \sum_{1 \leq i \leq n} z_i = d\}$. Call the set $\{\beta_1, \dots, \beta_T\}$ the *support* of f. For any two vectors $s, w \in S_n(d)$, let $var(s,w) = \sum_{1 \leq i \leq n} \max(s_i - w_i, 0) = -\sum_{1 \leq i \leq n} \min(s_i - w_i, 0)$. For any vector $v = (v_1, v_2, \dots, v_n)$ such that $\sum_{1 \leq i \leq n} v_i = 0$, represent v uniquely as $v = v^{(+)} + v^{(-)}$ where $v_i^{(+)} = \max(v_i, 0)$ and $v_i^{(-)} = \min(v_i, 0)$. We have $var(v) = \sum_{1 \leq i \leq n} v_i^{(+)} = -\sum_{1 \leq i \leq n} v_i^{(-)}$.

Let T_0 be the number of points in the intersection of the support of f with $S_n(d)$ and denote the points in this intersection by $\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_{T_0}$. Observe that for any $b \in \mathcal{Z}^n$ satisfying the conditions $var(b) = t_0, \ \mathbf{a}_j + b \in S_n(d)$ (see above) the point $\mathbf{a}_j + b$ belongs to the support of $\mathsf{D}^{t_0}(u^{\mathbf{a}_j})$ (due to the assumption in the statement of the theorem) and does not belong to the support of $\mathsf{D}^l(u^{\mathbf{a}_j})$ for any $l < t_0$.

For each m, $1 \le m \le n$, re-order the $\mathbf{a}_i$ in non-decreasing order on their m-th coordinates as $0 \le \mathbf{a}_{1,m} \le \mathbf{a}_{2,m} \le \ldots \le \mathbf{a}_{T_0,m} \le d$.

Lemma 3 *There is an m, $1 \le m \le n$, for which either $\mathbf{a}_{1,m} \ge t_0$, or, for a certain l, $1 \le l \le T_0-1$, $\mathbf{a}_{l+1,m} - \mathbf{a}_{l,m} \ge 2t_0 + 1$.*

Proof: Suppose not; then for every m we have $\mathbf{a}_{l,m} \le t_0 - 1 + (l-1)2t_0$ for each l, $1 \le l \le T_0$. Hence,

$$\sum_{1 \le l \le T_0} \mathbf{a}_{l,m} \le (t_0 - 1)T_0 + 2t_0T_0(T_0-1)/2 = T_0(t_0T_0 - 1) \le T_0(d/n - 1),$$

the latter inequality follows from the assumption that the theorem is wrong. On the other hand, $\sum_{1 \le m \le n} \sum_{1 \le l \le T_0} \mathbf{a}_{l,m} = dT_0$ since $\mathbf{a}_l \in S_n(d)$ for $1 \le l \le T_0$, we get a contradiction. Hence, the lemma is true. $\square$

Proof: **(of theorem 3):** Assume that $tT \le d/n$. Fix an m, $1 \le m \le n$ satisfying lemma 3.

First, consider the case $\mathbf{a}_{1,m} \ge t_0$ (see the beginning of the proof of the theorem). Consider those points among $\mathbf{a}_1, \ldots, \mathbf{a}_{T_0}$ that belong to the $(n-2)$-dimensional simplex $S_n(d) \cap \{\mathbf{a}_{j,m} = \mathbf{a}_{1,m}\}$. Without loss of generality, assume that they are $\mathbf{a}_1, \ldots, \mathbf{a}_{T_1}$, $T_1 \le T_0$. Among these points, choose the one that is largest in the lexicographic order, without loss of generality, let it be $\mathbf{a}_1$. Since $\mathbf{a}_{1,m} \ge t_0$, we have $\sum_{k \ne m} \mathbf{a}_{1,k} \le d - t_0$. Consider the point $\bar{\mathbf{a}}_1 = (\mathbf{a}_{1,1} + t_0, \mathbf{a}_{1,2}, \ldots, \mathbf{a}_{1,m-1}, \mathbf{a}_{1,m} - t_0, \ldots, \mathbf{a}_{1,n}) \in S_n(d)$. Clearly, $var(\mathbf{a}_1, \bar{\mathbf{a}}_1) = t_0$. We will show that for any point $\mathbf{a}_j$, $2 \le j \le T_0$, $var(\mathbf{a}_j, \bar{\mathbf{a}}_1) > t_0$, thereby showing that $\bar{\mathbf{a}}_1$ does not belong to the support of $\mathsf{D}^l(u^{\mathbf{a}_j})$ for all $0 \le l \le t_0$ and $2 \le j \le T_0$ and proving the theorem for the case $\mathbf{a}_{1,m} \ge t_0$. Suppose that $var(\mathbf{a}_j, \bar{\mathbf{a}}_1) \le t_0$; this means that $v = \mathbf{a}_j - \bar{\mathbf{a}}_1 = v^{(+)} + v^{(-)}$ with $v^{(+)} = (0, \ldots, 0, t_0, 0, \ldots, 0)$ (the t_0 appearing in the m-th position) and $\mathbf{a}_j \in \{\mathbf{a}_1, \ldots, \mathbf{a}_{T_1}\}$. Let $\mathbf{a}_j = (\mathbf{a}_{j,1}, \ldots, \mathbf{a}_{j,n})$. Since $\mathbf{a}_{j,1} \le \mathbf{a}_{1,1}$, (as $\mathbf{a}_1$ is lexicographically the largest among the $\mathbf{a}_j$), $v^{(-)} = (-t_0, 0, \ldots, 0)$, but then $\mathbf{a}_j = \mathbf{a}_1$, a contradiction! This proves that $var(\mathbf{a}_j, \bar{\mathbf{a}}_1) > t_0$.

Now, consider the case $\mathbf{a}_{l+1,m} - \mathbf{a}_{l,m} \ge 2t_0 + 1$ for a certain $l, 1 \le l \; T_0 - 1$ (according to lemma 3 we need just a trial of this and of the previous case). Among the points in $\{\mathbf{a}_1, \ldots, \mathbf{a}_{T_0}\} \cap \{\mathbf{a}_{j,m} = \mathbf{a}_{l+1,m}\}$, choose the largest in lexicographic order, call it $\mathbf{a}_{l+1}$. As before, consider the point $\bar{\mathbf{a}}_{l+1} = \mathbf{a}_{l+1} + (t_0, 0, \ldots, 0, -t_0, 0, \ldots, 0) \in S_n(d)$ (the $-t_0$ is in position m) and prove that $var(\mathbf{a}_j, \bar{\mathbf{a}}_{l+1}) > t_0$ for every j, $1 \le j \le T_0$ and $j \ne l+1$ and complete the proof as in the previous case. $\square$

2. Computing Sparse Multivariate Shifts

We assume in our discussion that f has a finite (possibly zero) number of sparse shifts. If there are infinitely many t-sparse shifts with respect to one or more x_i (when f is seen as an element of $\mathcal{F}[x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n][x_i]$ with $t > \deg_{x_i}(f)$) but only finitely many of them can be combined to t-sparse shifts for the polynomial f, then the algorithm of this section can fail. This situation is taken care of by the algorithm of next section as the "low degree case". In this section, we assume that $t \leq \deg_{x_i}(f)$ for $1 \leq i \leq n$. As before, let $f = \sum_{j=1}^{t} \mathrm{F}_j u^{\alpha_j}$. Recall that α_j denotes the multi-index $(\alpha_{j1}, \alpha_{j2}, \ldots, \alpha_{jn})$, and u^{α_j} indicates the power product $u_1^{\alpha_{j1}} u_2^{\alpha_{j2}} \ldots u_n^{\alpha_{jn}}$. Consider the ideal $I \subset \mathcal{Q}[y_1, y_2, \ldots, y_n]$ which is the ideal of the points $\{\alpha_1, \alpha_2, \ldots, \alpha_t\}$. We shall construct a reduced Gröbner basis for I under any admissible term ordering. For convenience, we choose the lexicographic term ordering with $y_1 \prec y_2 \prec \ldots \prec y_n$. In fact, we can construct a triangular set decomposition of the ideal I instead of a Gröbner basis for I. For a good description of zero-dimensional Gröbner bases and triangular sets, we refer the reader to (Lazard 1992, Kapur–Lakshman 1992).

The Gröbner basis G for I under the chosen term ordering looks as follows:

$$
\begin{array}{l}
G_1^{(1)}(y_1) = y_1^{\delta_1} + g_{1,\delta_1-1}^{(1)} y_1^{\delta_1-1} + \ldots + g_{1,1}^{(1)} y_1 + g_{1,0}^{(1)}, \\
G_1^{(2)}(y_1, y_2), G_2^{(2)}(y_1, y_2), \ldots, G_{k_2}^{(2)}(y_1, y_2), \\
G_1^{(3)}(y_1, y_2, y_3), G_2^{(3)}(y_1, y_2, y_3), \ldots, G_{k_3}^{(3)}(y_1, y_2, y_3), \\
\quad \vdots \\
G_1^{(n)}(y_1, y_2, \ldots, y_n), G_2^{(n)}(y_1, y_2, \ldots, y_n), \ldots, G_{k_n}^{(n)}(y_1, y_2, \ldots, y_n).
\end{array}
$$

We recall a few standard terms and facts from the theory of Gröbner bases. For further details, see (Becker et al 1993).

- the *head term* of a polynomial h, $\textsc{HeadTerm}(h)$ is the largest term (under the term ordering $\prec$) appearing in h with a non-zero coefficient.
- a term s is *reduced* with respect to the Gröbner basis G if it is not divisible by $\textsc{HeadTerm}(g)$ for any $g \in G$.
- The number of *reduced terms* with respect to the Gröbner basis G is equal to the dimension of the residue class ring $\mathcal{Q}[y_1, \ldots, y_n]/I$ as a $\mathcal{Q}$-vector space and, in our case, is $\leq t$.

Suppose $h \in I$ and

$$h = y^{\epsilon_1} + h_2 y^{\epsilon_2} + h_3 y^{\epsilon_3} + \ldots + h_L y^{\epsilon_L}$$

where $y^{\epsilon_i} = y_1^{\epsilon_{i,1}} y_2^{\epsilon_{i,2}} \ldots y_n^{\epsilon_{i,n}}$ and $y^{\epsilon_{i+1}} \prec y^{\epsilon_i}$ for $0 \leq i < L$. Let us define f_{ϵ_i} for a multi-index $\epsilon_i = (\epsilon_{i,1}, \epsilon_{i,2}, \ldots, \epsilon_{i,n})$ with $\epsilon_{i,1}, \epsilon_{i,2}, \ldots, \epsilon_{i,n} \geq 0$ as follows:

$$
\begin{array}{rcll}
f_{(0,0,\ldots,0)} & = & f, & \\
f_{(\epsilon_{i,1},\epsilon_{i,2},\ldots,\epsilon_{i,n})} & = & (x_1 - z_1)\partial f_{(\epsilon_{i,1}-1,\epsilon_{i,2},\ldots,\epsilon_{i,n})}/\partial x_1 & \text{if } \epsilon_{i,1} > 0, \text{ else} \\
 & = & (x_2 - z_2)\partial f_{(\epsilon_{i,1},\epsilon_{i,2}-1,\ldots,\epsilon_{i,n})}/\partial x_2 & \text{if } \epsilon_{i,2} > 0, \text{ else} \\
 & \vdots & & \\
 & = & (x_n - z_n)\partial f_{(\epsilon_{i,1},\epsilon_{i,2},\ldots,\epsilon_{i,n}-1)}/\partial x_n & \text{otherwise.}
\end{array}
$$

Furthermore, we extend the ordering $\prec$ to the set of polynomials f_{ϵ_i} as

$$f_{\epsilon_i} \prec f_{\epsilon_j} \text{ iff } y^{\epsilon_i} \prec y^{\epsilon_j}.$$

Let

$$\widehat{f_{\epsilon_i}} = f_{\epsilon_i}(z_1 = b_1, \ldots, z_n = b_n, x_1, x_2, \ldots, x_n) \in \mathcal{K}[x_1, \ldots, x_n]$$

where $(b_1, b_2, \ldots, b_n)$ is a t-sparse shift for f.

Lemma 4 *Let $h \in I$ and $h = y^{\epsilon_1} + h_2 y^{\epsilon_2} + h_3 y^{\epsilon_3} + \ldots + h_L y^{\epsilon_L}$. Then the polynomials $\widehat{f_{\epsilon_1}}, \widehat{f_{\epsilon_2}}, \ldots, \widehat{f_{\epsilon_L}}$ satisfy the $\mathcal{Q}$-linear relation*

$$h_1 \widehat{f_{\epsilon_1}} + h_2 \widehat{f_{\epsilon_2}} + h_3 \widehat{f_{\epsilon_3}} + \ldots + h_L \widehat{f_{\epsilon_L}} = 0, \ \textit{with } h_1 = 1.$$

Proof: We have

$$\widehat{f_{\epsilon_i}} = \sum_{j=1}^{t} \alpha_j^{\epsilon_i} \mathrm{F}_j (x - b)^{\alpha_j}$$

where $\alpha_j^{\epsilon_i} = \alpha_{j,1}^{\epsilon_{i,1}} \alpha_{j,2}^{\epsilon_{i,2}} \ldots \alpha_{j,n}^{\epsilon_{i,n}}$,and, $(x-b)^{\alpha_j} = (x_1 - b_1)^{\alpha_{j,1}} (x_2 - b_2)^{\alpha_{j,2}} \ldots (x_n - b_n)^{\alpha_{j,n}}$. Therefore,

$$\begin{aligned}
\sum_{i=1}^{L} h_i \widehat{f_{\epsilon_i}} &= \sum_{i=1}^{L} h_i \Big(\sum_{j=1}^{t} \alpha_j^{\epsilon_i} \mathrm{F}_j (x-b)^{\alpha_j}\Big) \\
&= \sum_{j=1}^{t} \sum_{i=1}^{L} h_i \alpha_j^{\epsilon_i} \mathrm{F}_j (x-b)^{\alpha_j} \\
&= \sum_{j=1}^{t} h(\alpha_j) \mathrm{F}_j (x-b)^{\alpha_j} \\
&= 0, \qquad \text{since } h \in I.
\end{aligned}$$

□

For polynomials $f_1, f_2, \ldots, f_m, f$ let us define a generalized Wronskian matrix and the ω-vector as

$$W_m(f_1, f_2, \ldots, f_m) = \begin{pmatrix} f_1 & f_2 & \ldots & f_m \\ \mathcal{D}(f_1) & \mathcal{D}(f_2) & \ldots & \mathcal{D}(f_m) \\ \vdots & \vdots & \ddots & \vdots \\ \mathcal{D}^{m-1}(f_1) & \mathcal{D}^{m-1}(f_2) & \ldots & \mathcal{D}^{m-1}(f_m) \end{pmatrix}, \ \omega_m(f) = \begin{pmatrix} f \\ \mathcal{D}(f) \\ \vdots \\ \mathcal{D}^{m-1}(f) \end{pmatrix},$$

where $\mathcal{D}$ is a generic linear combination of $\partial/\partial x_i$, i.e., $\mathcal{D} = \sum_{i=1}^{n} l_i \partial/\partial x_i, \quad l_i \in \mathcal{F}$. As usual, $\mathcal{D}^i$ denotes the operator $\mathcal{D}$ applied i times. Let

$$\mathcal{W}_m(f_1, f_2, \ldots, f_m) = \det(W_m(f_1, f_2, \ldots, f_m)).$$

Clearly,

$$\mathcal{W}_L(f_{\epsilon_1}, \ldots, f_{\epsilon_L}) \in \mathcal{F}[z_1, \ldots, z_n, x_1, \ldots, x_n]$$

and

$$\mathcal{W}_L(\widehat{f_{\epsilon_1}}, \ldots, \widehat{f_{\epsilon_L}}) \in \mathcal{K}[x_1, \ldots, x_n].$$

Lemma 5 *If a set of polynomials $F = \{\widehat{f_{\epsilon_1}}, \widehat{f_{\epsilon_2}}, \ldots, \widehat{f_{\epsilon_L}}\}$ satisfies the $\mathcal{K}$-linear relation*

$$\widehat{f_{\epsilon_1}} + h_2\widehat{f_{\epsilon_2}} + h_3\widehat{f_{\epsilon_3}} + \ldots + h_L\widehat{f_{\epsilon_L}} = 0,$$

and no proper subset of F satifies a $\mathcal{K}$-linear relation, then $\vec{h} = (h_2\ h_3\ \ldots\ h_L)^{Tr}$ is the unique solution to the system of equations

$$W_{L-1}(\widehat{f_{\epsilon_2}}, \ldots, \widehat{f_{\epsilon_L}})\vec{\phi} = -\omega_{L-1}(\widehat{f_{\epsilon_1}}).$$

Proof: The proof is classical and we give only a brief sketch. By rewriting the linear dependency as

$$-\widehat{f_{\epsilon_1}} = h_2\widehat{f_{\epsilon_2}} + h_3\widehat{f_{\epsilon_3}} + \ldots + h_L\widehat{f_{\epsilon_L}}$$

and applying the operator $\mathcal{D}$ successively $L-1$ times, it follows that

$$W_{L-1}(\widehat{f_{\epsilon_2}}, \ldots, \widehat{f_{\epsilon_L}})\vec{h} = -\omega_{L-1}(\widehat{f_{\epsilon_1}}).$$

If h is not the only solution, then $W_{L-1}(\widehat{f_{\epsilon_2}}, \ldots, \widehat{f_{\epsilon_L}})$ is singular, i.e., the Wronskian $\mathcal{W}_{L-1}(\widehat{f_{\epsilon_2}}, \ldots, \widehat{f_{\epsilon_L}})$ vanishes identically. This implies that the polynomials $\widehat{f_{\epsilon_2}}, \ldots, \widehat{f_{\epsilon_L}}$ are $\mathcal{K}$-linearly dependent (Kaplanski 1957). But no proper subset of F is supposed to satify a $\mathcal{K}$-linear relation, hence, h is the only solution. □

From lemmas 4,5, it follows that if $\vec{b} = (b_1, b_2, \ldots, b_n)$ is a t-sparse shift for f, and if $h = y^{\epsilon_1} + h_2y^{\epsilon_2} + h_3y^{\epsilon_3} + \ldots + h_Ly^{\epsilon_L} \in I$, then $\mathcal{W}_L(f_{\epsilon_1}, \ldots, f_{\epsilon_L})$ vanishes identically under the substitution $z_1 = b_1, z_2 = b_2, \ldots, z_n = b_n$. A partial converse of lemma 4 is also true and it gives us a way to determine sparse shifts of f.

For a polynomial $h = y^{\epsilon_1} + h_2y^{\epsilon_2} + h_3y^{\epsilon_3} + \ldots + h_Ly^{\epsilon_L}$, let *support*$(h)$ denote the list of polynomials $f_{\epsilon_1}, f_{\epsilon_2}, \ldots, f_{\epsilon_L}$ and $\mathcal{W}(h)$ denote $\mathcal{W}_L(\mathit{support}(h))$ where L is the cardinality of *support*(h).

Lemma 6 *Let $G = \{g_1, g_2, \ldots, g_r\} \subset \mathcal{Q}[y_1, y_2, \ldots, y_i]$ be a reduced Gröbner basis for a zero-dimensional ideal J with $\dim(\mathcal{Q}[y_1, y_2, \ldots, y_i]/J) \le t$, and $\vec{b_i} = (b_1, b_2, \ldots, b_i) \in \mathcal{K}^i$ such that*

$$\mathcal{W}(g_1)_{z_j=b_j,\ j=1,\ldots,i} = \mathcal{W}(g_2)_{z_j=b_j,\ j=1,\ldots,i} = \ldots = \mathcal{W}(g_r)_{z_j=b_j,\ j=1,\ldots,i} = 0.$$

Then $\vec{b_i}$ is a partial t-sparse shift for f, i.e., $f = \sum_{j=1}^{t} \phi_j(x-b)^{\beta_j})$ where $(x-b)^{\beta_j})$ denotes $(x_1-b_1)^{\beta_{j,1}}(x_2-b_2)^{\beta_{j,2}}\ldots(x_i-b_i)^{\beta_{j,i}}$ and J is the ideal of the points $\{\beta_1, \beta_2, \ldots, \beta_t\}$.

Proof: Let $f = \sum_{j=1}^{k} \phi_j(x-b)^{\beta_j}$ with $\phi_j \neq 0$. We will show that $k \le t$. Suppose $g_1 = g_{1,1}y^{\epsilon_1} + g_{1,2}y^{\epsilon_2} + g_{1,3}y^{\epsilon_3} + \ldots + g_{1,L}y^{\epsilon_L}$. Since $\mathcal{W}(g_1)_{z_j=b_j,\ j=1,\ldots,i} = 0$, by lemma 5, $g_{1,1}\widehat{f_{\epsilon_1}} + g_{1,2}\widehat{f_{\epsilon_2}} + g_{1,3}\widehat{f_{\epsilon_3}} + \ldots + g_{1,L}\widehat{f_{\epsilon_L}} = 0$, i.e.,

$$\begin{aligned}
&= \sum_{p=1}^{L} g_{1,p}(\sum_{j=1}^{k} \beta_j^{\epsilon_p}\phi_j(x-b)^{\beta_j}) \\
&= \sum_{j=1}^{k}\sum_{p=1}^{L} g_{1,p}\beta_j^{\epsilon_p}\phi_j(x-b)^{\beta_j}
\end{aligned}$$

$$
\begin{aligned}
&= \sum_{j=1}^{k} g_1(\beta_j)\phi_j(x-b)^{\beta_j} \\
&= 0.
\end{aligned}
$$

Therefore, $g_1(\beta_j) = 0$ for $j = 1, \ldots, k$. In a similar fashion, we can show that $g_p(\beta_j) = 0$ for each $g_p \in G$. In other words, each β_j is a zero of the ideal J. Since $\dim(\mathcal{Q}[y_1, y_2, \ldots, y_i]/J) \le t$, J can have at most t distinct zeros. Therefore, $k \le t$. $\square$

2.1 The Sparse Shift Algorithm – Case of Finitely Many Shifts

Our goal is to construct the Gröbner basis G. Given any term s (in $y_1, \ldots, y_n$), it is either

- *reduced* with respect to G, or
- the *head term* of some polynomial in G, or
- a *multiple* of some head term in G.

A term s is called a *simple* multiple of the term s' if $s = y_j s'$ for some y_j. Every head term in G is a *simple* multiple of some reduced term. The idea is to enumerate f_{ϵ_i} in the increasing order according to $\prec$ and decide, with the help of a Wronskian test, which of the above properties the term y^{ϵ_i} satisfies. The idea of systematic enumeration is borrowed from the well-known FGLM basis conversion algorithm (Faugère et al 1993).

• The function *Complete_basis* constructs a reduced Gröbner basis for the ideal I under the pure lexicographic term order with $y_1 \prec y_2 \prec \ldots \prec y_n$. In the process, it also computes the corresponding t-sparse shifts for f. If f has several t-sparse shifts satisfying the assumptions stated earlier, *Complete_basis* constructs reduced Gröbner bases for the ideal I corresponding to each of the sparse shifts. It returns a set of ordered pairs $(\vec{b}, G)$ where $\vec{b}$ is a t-sparse shift for f and G is a reduced Gröbner basis for the corresponding ideal I. It uses the functions *Next_term* and *Wronskian_test*.

• The function *Next_term* takes four parameters:

- *current_basis*, *reduced_terms* and *last_shift_index* are passed in unmodified by *Complete_basis*, and,
- *new_var* is a flag that is set by *Next_term*.

Next_term returns the smallest term $y^{\epsilon}(=: s)$ (according to the ordering $\prec$) that is neither in *reduced_terms* nor is a multiple of some known head term in *current_basis*; it returns **null** if no such y^{ϵ} exists. When such a term exists, if it has a new variable, i.e., the number of variables in the term is $>$ *last_shift_index*, the flag *new_var* is set to **true**, else *new_var* is set to **false**.

• The function *Wronskian_test* takes five parameters: s,*reduced_terms*, *list_of_bis*, *last_shift_index* and *new_var* which are all passed in unmodified by *Complete_basis* and it operates in two distinct modes.

- If $new\_var =$ **true**, then *Wronskian_test* tries to extend the partial t-sparse shift $list\_of\_bis = (b_1, \ldots, b_k)$ to the next variable, i.e., variable whose index is $last\_shift\_index + 1 = k + 1$. If it finds a possible shift, it returns the pair of values [**true**, *bis_or_poly*] where *bis_or_poly* contains all the zeros of the *content* of $\mathcal{W}_{L+1}(f_\epsilon, f_{\epsilon_1}, \ldots, f_{\epsilon_L})_{z_i=b_i,\ i=1,\ldots,k}$ as an element of $\mathcal{K}[z_{k+1}][x_1, x_2, \ldots, x_n]$ where $y^{\epsilon_1}, \ldots, y^{\epsilon_L} \in reduced\_terms$, and $L = Cardinality(reduced\_terms)$. If the content is 1, then *Wronskian_test* returns the pair of values [**false**, [ ]] .
- If $new\_var =$ **false**, then *Wronskian_test* attempts to solve the system of linear equations

$$W_L(\widehat{f_{\epsilon_1}}, \ldots, \widehat{f_{\epsilon_L}})\vec{\phi} = -\omega_L(\widehat{f_\epsilon})$$

  where $y^{\epsilon_1}, \ldots, y^{\epsilon_L} \in reduced\_terms$, and $L = Cardinality(reduced\_terms)$. If there is a solution $\phi = (g_1\ g_2\ \ldots\ g_L)^{Tr}$, then *Wronskian_test* returns the pair of values [**true**, $y^\epsilon + g_1 y^{\epsilon_1} + \ldots + g_L y^{\epsilon_L}$] where $y^\epsilon = s$. If there is no solution, then *Wronskian_test* returns the pair of values [**false**, [ ]] .

Complete_basis(f,

current_basis,	/* set containing a partially constructed Gröbner basis; */
reduced_terms,	/* set containing terms known to be reduced with respect to *current_basis*; */
list_of_bis,	/* list containing partial shifts; if $list\_of\_bis = (b_1, \ldots, b_k)$, then the b_i are possible t-sparse shifts for x_i, $i = 1, \ldots, k$; */
last_shift_index	/* index of the last variable for which a shift has been computed; $last\_shift\_index =\ Cardinality(list\_of\_bis)$ always; */
term_limit	/* bound on the number of terms (t) in the shifted sparse representation of f; */

)

begin
local

set_of_GBs,	/* $(\vec{b}, G)$ are accumulated in this; */
new_var, *Wr_flag*,	
bis_or_poly;	/* flags and place-holders for return values and return status of *Next_term*, *Wronskian_test*; */

```
set_of_GBs := { };
if Cardinality(reduced_terms) ≤ term_limit then
   if s := Next_term( reduced_terms, current_basis,
                      last_shift_index, new_var)  then
      [ Wr_flag, bis_or_poly ] := Wronskian_test( s, reduced_terms,
                      list_of_bis, last_shift_index, new_var);
      if new_var then
         if Wr_flag then
```

```
        for  each b ∈ bis_or_poly do
          set_of_GBs :=  set_of_GBs ∪  Complete_basis(reduced_terms,
                  current_basis, [list_of_bis, b], last_shift_index+1, term_limit);
        od ;
        set_of_GBs :=  set_of_GBs ∪  Complete_basis(reduced_terms ∪ s,
                  current_basis, list_of_bis, last_shift_index, term_limit);
        return set_of_GBs      /* new_var is true, and the flag Wr_flag is set to trueby
                                 Wronskian_test ; therefore, one or more possible shifts were
                                 found for the last_shift_index+1-th variable; branch out to
                                 complete each of the shifts; continue the original branch also
                                 in search of other shifts. */
      else
        set_of_GBs :=  Complete_basis(reduced_terms ∪ s,
                  current_basis, list_of_bis, last_shift_index, term_limit)
        return set_of_GBs;  fi  /* No possible shift was found for the last_shift_index+1-
                                 th variable; no Q-linear combination of the term s and all
                                 the lower terms known to be reduced with respect to cur-
                                 rent_basis belongs to I; classify s as reduced with respect to
                                 current_basisand continue.*/
    else
      if Wr_flag then
        set_of_GBs :=  Complete_basis(reduced_terms, current_basis ∪
                  bis_or_poly, list_of_bis, last_shift_index, term_limit)
        return set_of_GBs      /* new_var is false, and the flag Wr_flagis set to trueby
                                 Wronskian_test , therefore, a Q-linear combination of the
                                 term s and all the lower terms known to be reduced with
                                 respect to current_basis was found to belong to I; update the
                                 current_basis and continue. */
      else
        set_of_GBs :=  Complete_basis(reduced_terms ∪ s,
                  current_basis, list_of_bis, last_shift_index, term_limit)
        return set_of_GBs;  fi ; /* no Q-linear combination of the term s and all the lower
                                 terms known to be reduced with respect to current_basis
                                 belongs to I; classify s as reduced with respect to cur-
                                 rent_basisand continue.*/
     fi
  else
    return { (list_of_bis, current_basis) }
                                 /* Next_term failed to return a new term; so, every term is
                                 either known to be reduced with respect to current_basis, or
                                 is a multiple of some head term in current_basis. This means
                                 that current_basis is a zero-dimensional Gröbner basis; return
                                 the basis and the corresponding shift. */
  fi
else
```

```
    return { } ;              /* The basis being built has more than t reduced terms which
                                 means that the shift being computed can not be completed
                                 to a t-sparse shift. */
end.
```

As described earlier, the function *Wronskian_test* returns a pair of values consisting of a flag (to denote what was computed) and either a list of possible shifts or a new element of the Gröbner basis being constructed. It uses two functions, *Roots_of_Content* and *Lin_Sys_Solve*.

- The function *Roots_of_Content* takes 2 parameters, an index k, $0 \le k \le n-1$, and a polynomial $w \in \mathcal{K}[z_{k+1}][x_1, x_2, \ldots, x_n]$. *Roots_of_Content* returns a list of all the zeros of the content of the polynomial w.
- The function *Lin_Sys_Solve* takes 2 parameters, $W(\in \mathcal{K}^{m\times m}), \omega(\in \mathcal{K}^m)$, and attempts to solve the $m \times m$ system of linear equations $W\vec{\phi} = \omega$. If the system has a solution, it returns a $\vec{g} \in \mathcal{K}^m$ such that $W\vec{g} = \omega$, else it returns a null list. In our case, the system of equations either has a unique solution or no solution.

Wronskian_test(s,
reduced_terms,
list_of_bis,
last_shift_index,
new_var
)

begin
local

shift_candidates, /* used to accumulate the list of zeroes of the *content* of $\mathcal{W}_{L+1}(f_\epsilon, f_{\epsilon_1}, \ldots, f_{\epsilon_L})_{z_i=b_i,\ i=1,\ldots,k}$ as an element of $\mathcal{K}[z_{k+1}][x_1, x_2, \ldots, x_n]$ where $y^{\epsilon_1}, \ldots, y^{\epsilon_L} \in$ *reduced_terms*, and $L =$ *Cardinality*(*reduced_terms*). */
list_of_coeffs, /* used to store the list of coefficients returned by *Lin_Sys_Solve*. */
basis_element, /* used to store a polynomial $y^\epsilon + g_1 y^{\epsilon_1} + \ldots + g_L y^{\epsilon_L}$ that will become part of the Gröbner basis being constructed. */
k;

$k :=$ *last_shift_index*;
if *new_var* **then**
  shift_candidates := *Roots_of_Content*(*last_shift_index* + 1,
    $\mathcal{W}_{L+1}(f_\epsilon, f_{\epsilon_1}, \ldots, f_{\epsilon_L})_{z_i=b_i,\ i=1,\ldots,k}$);
  if *shift_candidates* $\neq$ [ ] **then**
    return [**true**, *shift_candidates*]

else
return [**false**, [ ]] **fi**
else
list_of_coeffs := $Lin\_Sys\_Solve(W_L(\widehat{f_{\epsilon_1}}, \ldots, \widehat{f_{\epsilon_L}}), -\omega_L(\widehat{f_\epsilon}))$;
if *list_of_coeffs* $\neq$ [ ] **then** /* *list_of_coeffs* is a list of coefficients $[g_1, g_2, \ldots, g_l]$ */
basis_element := $y^\epsilon + g_1 y^{\epsilon_1} + g_2 y^{\epsilon_2} + \ldots + g_L y^{\epsilon_L}$;
return [**true**, *basis_element*]
else
return [**false**, [ ]] **fi**
fi
end.

Initially, *Complete_basis* is invoked with the following parameter values: the polynomial f, *current_basis* set to { } , *reduced_terms* set to $\{1\}$, *list_of_bis* set to ( ), *last_shift_index* set to 0, and *term_limit* set to t.

Correctness of the Algorithm: The algorithm is in one of two states always, *new_var*being **true** and *new_var*being **false**.

State 1, *new_var* **true:** In this state, the algorithm is attempting to extend *current_basis* to include a polynomial in $\mathcal{F}[y_1, \ldots, y_{i+1}]$ (where i = *last_shift_index*). Since the Gröbner basis being constructed is a lexicographically ordered basis, and the partial basis *current_basis* is being built from the smallest head term up, *current_basis* is actually a reduced Gröbner basis for a zero-dimensional ideal J in $\mathcal{F}[y_1, \ldots, y_i]$ with $\dim(\mathcal{Q}[y_1, y_2, \ldots, y_i]/J) \leq t$, and the elements of *current_basis* satisfy the Wronskian tests with respect to the candidate shift *list_of_bis*. By theorem 6, *list_of_bis* is indeed a partial shift for f and a candidate for a complete shift. There are at most finitely many ways in which this partial shift can be extended to variable x_{i+1} (that is our assumption). Each possible way to extend the shift appears as a root of the content of an appropriate Wronskian, the algorithm tries to find the contents of all such Wronskians by exhaustive enumeration. If the content of a Wronskian has more than one root, each root is a possible way to continue the shift *list_of_bis* and the algorithm branches into as many branches as the roots and also continues the computations along the parent branch. When the algorithm starts a new branch, it enters state 2.

State 2, *new_var* **false:** In this state, the algorithm already has a candidate shift for $x_1, \ldots, x_i$ and is attempting to extend *current_basis* to include relevant elements of the Gröbner basis that are in $\mathcal{F}[y_1, \ldots, y_i]$ (where i = *last_shift_index*). If a polynomial $h \in \mathcal{F}[y_1, \ldots, y_i]$ belongs to the ideal under construction, then by lemmas 4 and 5, h satisfies the Wronskian test. If h is indeed an element of the target Gröbner basis, then it will be generated by the algorithm because of the particular order in which the Wronskian tests are performed (this is one of the key ideas in the FGLM algorithm; see (Faugere et al) for details). The algorithm enters state 1 when after it generates a basis element whose head term is a pure power of x_i (because *Next_term* now sets *new_var* to **true**).

In every branch, in either state, after a Wronskian test, either a new reduced term is deduced or a new basis polynomial is generated. The number of basis elements is bounded by *term_limit* and the number of reduced terms is bounded by *term_limit*. If the cardinality of *reduced_terms* exceeds *term_limit*, then the ideal J has more than t zeros, i.e., the partial shift *list_of_bis* cannot be extended to x_{i+1} and the algorithm terminates that branch. Therefore, each branch terminates after at most $(n+1)t$ Wronskian tests. When a branch terminates, if the number of reduced terms is $\leq$ *term_limit*, then the branch must have terminated because *Next_term* returned the empty list; this means that every term is known to be either reduced or a head term or a multiple of a head term with respect to *current_basis* which means that *current_basis* is a reduced Gröbner basis in $\mathcal{Q}[x_1, \ldots, x_n]$.

If there are infinitely many t-sparse shifts with respect to one or more x_i ($t > \deg_{x_i}(f)$) but only finitely many of them can be combined to t-sparse shifts for the polynomial f, then the algorithm can encounter an identically vanishing Wronskian in state 1 and will fail. An example of a polynomial for which such a phenomenon happens is $(x-1)^2(y-1)^2(z-1)^2 + (x-1)(y-1)(z-1)+2$. $(1,1,1)$ is a 3-sparse shift for this polynomial and it is the only 3-sparse shift for the polynomial. However, with respect to any single variable, there are infinitely many 3-sparse shifts. This situation is taken care of by the algorithm of the next section as the "low degree case".

The function *Complete_basis* returns a set of ordered pairs $(\vec{b}, G)$ where $\vec{b}$ is a t-sparse shift for f and G is a reduced Gröbner basis for the corresponding ideal I. From these, it is quite straightforward to compute the shifted sparse representation $f = \sum_{i=1}^{t} \mathrm{F}_i u^{\alpha_i}$ corresponding to the pair $(\vec{b}, G)$.

- Find the zeros of the ideal (G). We know that the zeros are the multi-indices α_i in the above representation. Since we already have a reduced, lexicographically ordered Gröbner basis and know that all the roots are n-tuples of integers, this can be done fast. See (Lazard 1992).
- Once we know the shift $\vec{b}$ and the multi-indices α_i, we can find the coefficients of f from its values at t selected points by solving a $t \times t$ system of linear equations.

Complexity Analysis: In the following analysis, our main goal is to get an upper bound on the number of $\mathcal{Q}$-operations performed by the sparse shift algorithm. Our emphasis is not so much on getting the sharpest possible bounds (as that would depend on the intricate details of how each step is implemented) as on finding the coarse dependence (polynomial or exponential) of the running time (number of $\mathcal{Q}$-operations) on n, t, d. We choose the primitive element methos for computing with algebraic numbers for convenience and the other models that one finds in the literature are polynomially related to this (polynomial in the degree of the extension under consideration).

The polynomial f is assumed to have rational number coefficients. In fact, the algorithm could run over any field of characteristic zero, but in the complexity analysis we assume that the coefficients of f are rationals. The t-sparse shifts may be algebraic over $\mathcal{Q}$ as we have seen. We assume that the algebraic numbers that arise in a particular branch of the algorithm are expressed as $\mathcal{Q}$-linear combinations of $1, \zeta, \zeta^2, \ldots$ where ζ is a primitive element of the smallest degree algebraic extension over $\mathcal{Q}$ that contains all the algebraic numbers that arise in that branch.

The main operation in the algorithm is the Wronskian test in *Complete_basis*. We know that in each branch generated by *Complete_basis*, there are at most $n(t+1)$ Wronskian tests. How many distinct branches can there be? Notice that branching can take place only when *Complete_basis* is in state 1. Branching corresponds to finding more than one root for the content of a Wronskian. Consider a Wronskian test in state 1 with *last_shift_index* $= i$. The Wronskians are determinants of matrices of size no more than $(t+1) \times (t+1)$ and the degree of each entry in z_{i+1} is no more than t, therefore, the content of the Wronskian, which is a polynomial in z_{i+1}, has degree $O(t^2)$ in z_{i+1}. Hence a worst case bound on the number of branches at a time is $O(t^2)$. The main branch in state 1 can branch at most t times and the main branches are the only ones that can branch. The branching stops when *last_shift_index* becomes equal to n. Therefore, the number of branches is bounded by $O(t^{3n})$. Conclude that there are $O(nt^{3n+1})$ Wronskian tests in all.

Consider any branch generated by *Complete_basis* at a time when *last_shift_index* $= i$. The entries of the Wronskian are polynomials in z_{i+1} and $x_1, \ldots, x_n$ with coefficients from an algebraic extension of the rationals of degree $O(t^{2i})$ (the contents of the previous Wronskians whose roots form the partial shift along the chosen branch are of degree $O(t^2)$ in z_j, $j < i+1$, and if each of the contents is irreducible over the earlier extensions, the current extension will have degree $O(t^{2i})$ over $\mathcal{Q}$). Each arithmetic operation in such an extension costs $O(t^{4i})$ $\mathcal{Q}$-operations.

To compute the content of a Wronskian, we have to compute the Wronskian (a $(t+1) \times (t+1)$ determinant at most). The Wronskian is a polynomial of degree $O(dt)$ in the x_i and $O(t^2)$ in z_{i+1}. Computing the Wronskian and then its content costs $O(t^3 (dt)^{2n} t^{4n}) = O(d^{2n} t^{6n+3})$ $\mathcal{Q}$-operations if we are to do it deterministically. If we are allowed to use randomization, we can substitute two different sets of random rational numbers for the x_i in the matrix corresponding to the Wronskian and compute the univariate gcd of the determinants of the matrix under the two specializations. With high probability, the gcd will be the content of the Wronskian. The cost of doing this is $O(t^3.t^4.t^{4n}) = O(t^{4n+7})$ $\mathcal{Q}$-operations.

The last two steps in the sparse shift algorithm are much simpler. Finding the integer roots of a zero-dimensional lexicographic Gröbner basis with at most t zeros can be done by finding all the integer roots of a univariate polynomial of degree at most t and evaluating $n-1$ other univariate polynomials of degree $< t$ at the roots this polynomial and solving an $n \times n$ linear system (see (Lakshman 1990)). The total cost is bounded by $O(t^3 + nt^2 + n^3)$ $\mathcal{Q}$-operations. This has to be done at most $O(nt^{3n+1})$ times (once for each branch of *Complete_basis*).

Setting up and solving a $t \times t$ linear system to compute the coefficients in the shifted sparse representation costs $O(t^3)$ $\mathcal{Q}$-operations and this too has to be done at most $O(nt^{3n+1})$ times, once for each branch of *Complete_basis*. Adding up the costs of all the steps, we have:

Theorem 4 *The algorithm of this section computes all shifted t-sparse representations for f provided $\deg_{x_i}(f) \geq t$ for each x_i. If randomization is not allowed, the algorithm performs $(n(dt)^n)^{O(1)}$ $\mathcal{Q}$-operations. If randomization is allowed, the algorithm performs $(nt^n)^{O(1)}$ $\mathcal{Q}$-operations.* □

For the special case of $t \leq (\deg_{x_i}(f)+1)/2$ for each x_i, the polynomial f has at most one t-sparse shift by lemma 2, the algorithm runs much faster. In this case, there is essentially no branching and all individual shifts are rational. For this case, we have:

Theorem 5 *If $t \leq (\deg_{x_i}(f)+1)/2$ for each x_i, the algorithm of this section computes a shifted t-sparse representations for f (if it has one) in time polynomial in t. More specifically, if random-*

ization is not allowed, the algorithm performs $(ntd^n)^{O(1)}$ $\mathcal{Q}$*-operations. If randomization is allowed, the algorithm performs* $(nt)^{O(1)}$ $\mathcal{Q}$*-operations.* $\square$

Remark: If we do not have the derivarives of f available, but only a black box that evaluates f, we can still use ideas close to the ones describes in this section to construct a Gröbner basis for the ideal of points $p^{\alpha_1}, p^{\alpha_2}, \ldots, p^{\alpha_t}$ with $p_j^{\alpha_j}$ denoting the power product $p_1^{\alpha_{j,1}} p_2^{\alpha_{j,2}} \ldots p_t^{\alpha_{j,t}}$ where the p_i are distinct prime integers. In place of f_ϵ, one can use $f(p_1^{\epsilon_1} + z_1, p_2^{\epsilon_2} + z_2, \ldots, p_n^{\epsilon_n} + z_n)$.

3. Sparse Shift Algorithm – The General Case

In this section, we deal with the case of finding t-sparse shifts for a polynomial $f \in \mathcal{F}[x_1, \ldots, x_n]$ (we suppose that $\mathcal{F} \subset \mathcal{R}$) for which $\deg_{x_i}(f) < 2t$ for one or more x_i and consequently, there might be several (possibly infinitely many) t-sparse shifts with respect to x_i alone. This covers the case that causes the algorithm in the previous section to fail. We assume that the polynomial f is given to us as a black box or a straight line program. Assuming that we are given a bound M on the sizes of the coefficients of f in the standard representation (and without knowing anything about the degree of f), we identify three kinds of variables in f:

- variables that have very high degree ($\Omega((M + t^2)^{1+\epsilon})$), these appear with a unique t-sparse shift that is either 0, or 1 or -1. We make repeated use of the Ben-Or and Tiwari algorithm (Ben-Or, Tiwari 1988) in conjunction with some bounds proved in this section to identify the high degree variables and to find the corresponding shifts.
- variables that have moderate degree ($2t <$ degree $< c(M+t^2)^{1+\epsilon}$ for some constant c), these appear with a unique, rational shift in any t-sparse representation of f. We make repeated use of the sparse-shift algorithm in (Lakshman-Saunders 1994) to identify the moderate degree variables.
- variables that have low degree (degree $< 2t$), these may appear with different, algebraic shifts in different t-sparse representation of f. We make use of the algorithm in (Grigoriev-Karpinski 1993) to identify the low degree variables and find their sparse shifts.

The algorithm used to find sparse shifts for the low degree variables (Grigoriev-Karpinski 1993) is very general and in fact solves the problem of computing sparse shifts completely without making such distinctions as listed here. However, we apply it selectively, to polynomials of low degree ($d \leq 2tn$), and as a result, the complexity of the algorithm comes down from $O(d^{O(n^4)})$ to $O((nt)^{O(n^2)})$ operations.

3.1 Degree Bounds on Shifted Sparse Polynomials

We establish some bounds on the degrees of f in each x_i in terms of t, n, M. These bounds are used in the main algorithm which unfolds in the rest of the section. The main steps of the algorithm are collected together at the end of the section.

Let $f = \sum_{i=0}^{d} f_i x^i = \sum_{j=1}^{t} \mathrm{F}_j (x-b)^{\delta_j}$. For a rational number $p/q \in \mathcal{F}$, define $\mathrm{SIZE}(p/q) =$

$\lceil \log_2(p+1)\rceil + \lceil \log_2(q+1)\rceil$ and for the polynomial $f \in \mathcal{F}[x]$ as above, define $\textsc{size}(f) = \max\{\textsc{size}(f_i)\}$. Let $\textsc{size}(f) \leq M$ for an integer M.

Lemma 7 *For any $\epsilon > 0$ there exists c such that if f has two sparse representations as above where $b \notin \{0, 1, -1\}$, then $d \leq c(M + t^2)^{1+\epsilon}$.*

Proof: If $d \leq 3t$, the lemma is obvious. If $d > 3t$, then b is unique and rational (Lakshman-Saunders 1994) and we have $f = \sum_{i=0}^{d} f_i x^i = \sum_{j=1}^{t} \mathrm{F}_j (x-b)^{\delta_j}$ with $d = \delta_t > \delta_{t-1} > \ldots > \delta_1$. Rewrite this as

$$\vec{\mathrm{F}} \mathcal{B}_t \mathcal{C} \mathcal{D}_{d+1} = \vec{f}$$

where $\vec{\mathrm{F}} = (\mathrm{F}_{\delta_1}\ \mathrm{F}_{\delta_2}\ \ldots \mathrm{F}_{\delta_t})$, $\mathcal{B}_t$ is the $t \times t$ diagonal matrix with $\mathcal{B}_t(j,j) = b^{\delta_j}$, $\mathcal{C}$ is a $t \times (d+1)$ matrix with $\mathcal{C}(i,j) = \binom{\delta_i}{d+1-j}$, $\mathcal{D}_{d+1}$ the $(d+1) \times (d+1)$ diagonal matrix with $\mathcal{B}_t(j,j) = b^{-d+j-1}$, and, $\vec{f} = (f_d\ f_{d-1}\ \ldots\ f_0)$. Let $\vec{u} = (u_1\ u_2\ \ldots\ u_t) = \vec{\mathrm{F}}\mathcal{B}_t$. Let $\mathcal{C}_t$ be the $t \times t$ submatrix of $\mathcal{C}$ consisting of the last t columns of $\mathcal{C}$.

We know that $\mathcal{C}_t$ is non-singular (see (Lakshman-Saunders 1994)) and hence,

$$\vec{u} = (f_{t-1}\ f_{t-2}\ \ldots\ f_0) \begin{pmatrix} b^{t-1} & 0 & \ldots & 0 \\ 0 & b^{t-2} & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & b^0 \end{pmatrix} \mathcal{C}_t^{-1}.$$

Therefore, $\textsc{size}(u_1) < M + t\textsc{size}(b) + ct^2 \log d$ for some constant c, the $ct^2 \log d$ term coming from $\mathcal{C}_t^{-1}$. Since $f_d = u_1 b^{-d}$, we get

$$\textsc{size}(u_1) > \textsc{size}(b^d) - M > d(\textsc{size}(b) - 2) - M$$

Since $b \notin \{0, 1, -1\}$, we have $\textsc{size}(b) \geq 3$ and therefore, $d - 3t \leq (d-3t)(\textsc{size}(b)-2) \leq c'(M + t^2 \log d)$ for some constant c'. It follows that $d/logd \leq c'(M + t^2)$ and $d = O((M+t^2)^{1+\epsilon})$ for any $\epsilon > 0$. $\Box$

Lemma 8 *Let $g \in \mathcal{F}[x_1, x_2, \ldots, x_n]$ be a non-zero polynomial that is shifted t-sparse and $S = \{a_1 < a_2 < \ldots < a_{2t}\} \subset \mathcal{R}$. Then g cannot vanish everywhere on S^n.*

Proof: The lemma follows from successive applications of Descartes' rule to each variable $x_1, x_2, \ldots, x_n$. $\Box$

Choose a constant $\hat{c}$ and an arbitrarily small $\epsilon > 0$ such that $deg_{x_i}(f) < \hat{c}(M + t^2)^{1+\epsilon}$ for $1 \leq i \leq n$ in lemma 7. Denote this degree bound $\hat{c}(M + t^2)^{1+\epsilon}$ by D.

Let $1 \leq i \leq n$. Fix b to be one of $0, -1, -1$ and let $X_i = x_i - b$. For each $W = (w_1, w_2, \ldots, w_n) \in S^{n-1}$, consider the univariate polynomial $f_W(X_i) = f(w_1, \ldots, w_{i-1}, x_i, w_{i+1}, \ldots, w_n)$. By lemma 8, it follows that there is a W such that $\deg(f_W(X_i)) = \deg_{x_i}(f)$. Fix such a W.

Let $A_1 = \{-2, (-2)^2, (-2)^3, \ldots, (-2)^{2t}\}$, $A_2 = \{-3, (-3)^2, (-3)^3, \ldots, (-3)^{2t}\}$, $A_3 = \{-5, (-5)^2, (-5)^3, \ldots, (-5)^{2t}\}$, and, $A_4 = \{-7, (-7)^2, (-7)^3, \ldots, (-7)^{2t}\}$. Apply the Ben-Or and Tiwari interpolation algorithm (Ben-Or, Tiwari 1988) to $f_W(X_i)$ at the four sets of points A_1, A_2, A_3, A_4 respectively. If the algorithm succeeds, it returns a t-sparse polynomial in $\mathcal{F}[X_i]$ for each set of evaluation points A_1, A_2, A_3, A_4. Let us denote the polynomials returned by the Ben-Or and Tiwari algorithm by $g_1, g_2, g_3, g_4 \in \mathcal{F}[X_i]$ respectively.

Lemma 9 $\deg_{x_i}(f) > D$ *if and only if* $g_1 = g_2 = g_3 = g_4$ *and* $\deg(g_1) > D$.

Proof: $(\Longrightarrow)$: Since $\deg_{x_i}(f) > D$, then by lemma 7, we have $b_i \in \{-1, 0, 1\}$ in any t-sparse shift $(b_1, b_2, \ldots, b_n)$ for f. Therefore, $f_W(X_i)$ is t-sparse for any $W \in S^{n-1}$ and since the interpolating polynomial produced by the Ben-Or and Tiwari algorithm (Ben-Or, Tiwari 1988) is unique, we have $g_1 = g_2 = g_3 = g_4 = f_W(X_i)$. Since we choose a W such that $\deg(f_W(X_i)) = \deg_{x_i}(f)$, it follows that $\deg(g_1) > D$.

$(\Longleftarrow)$: Assume that for a certain W_0, the Ben-Or and Tiwari algorithm returns four t-sparse polynomials in X_i such that $g_1 = g_2 = g_3 = g_4$ and $\deg(g_1) > D$. The polynomial $f_{W_0}(X_i)$ is shifted t-sparse and coincides with the t-sparse polynomial g_1 at $4t$ positive points and $4t$ negative points. Therefore, by theorem 5 in (Lakshman-Saunders 1994), we have $f_{W_0}(X_i) = g_1(X_i)$. □

If the Ben-Or, Tiwari algorithm fails to return, for each $b = 0, 1, -1$, and for each W four t-sparse polynomials in X_i $g_1 = g_2 = g_3 = g_4$ with $\deg_{X_i}(g_1) > D$, then $f_W(x_i)$ has a non-zero t-sparse shift and $\deg_{x_i}(f) < D$. Denote the set of all indices i such that $\deg_{x_i}(f) > D$ by I_B. The set I_B can be determined by performing $((\log d)t^n)^{O(1)}$ arithmetic operations (for each i, for each b, and for each $W \in S^{n-1}$, we have to perform the Ben-Or and Tiwari algorithm 4 times).

Let $j \in \{1, 2, \ldots, n\} \setminus I_B$. For each $W \in S^{n-1}$, interpolate $f_W(x_j)$ as a dense univariate polynomial. If for some W_0, $\deg(f_{W_0}(x_j) > 2t$, then we can use the sparse-shift algorithm from (Lakshman-Saunders 1994) to find the unique $b_j \in \mathcal{F}$ such that $f_{W_0}(X_j)$ is t-sparse relative to $x_j - b_j$.

Lemma 10 *Let* $j \in \{1, 2, \ldots, n\} \backslash I_B$. *Then* $\deg_{x_j}(f) > 2t$ *iff for a certain* $W_0 \in S^{n-1}$, $\deg(f_{W_0}(x_j)) > 2t$. *If the latter is true, then there is unique* $b_j \in \mathcal{F}$ *such that* $f_{W_0}(x_j)$ *is* t*-sparse relative to* $x_j - b_j$. *Moreover,in any* t*-sparse shift* $(b'_1, b'_2, \ldots, b'_n)$ *of* f, *we have* $b'_j = b_j$.

Proof: The existence of W_0 follows from lemma 8. Since $f_{W_0}(x_j)$ is shifted t-sparse, and its degree $> 2t$, the shift is unique (theorem 1, Lakshman-Saunders, 1994). If $(b'_1, b'_2, \ldots, b'_n)$ is a t-sparse shift for f, then $f_{W_0}(x_j)$ is t-sparse with respect to $x_j - b'_j$. From the uniqueness of the shift, it follows that $b'_j = b_j$. □

Denote the set of all indices $j \in \{1, 2, \ldots, n\} \setminus I_B$ such that $\deg_{x_j}(f) > 2t$ by I_C. The set I_C can be determined by performing $(Mt^n)^{O(1)}$ arithmetic operations (for each j and for each $W \in S^{n-1}$, we have to perform a dense univariate interpolation and the sparse-shift algorithm of Lakshman-Saunders once; since the degree with respect to x_j is bounded by D, we have the above bound.)

From lemmas 9, 10, it follows that for every $i \in I_B$ only the powers of x_i and for every $j \in I_C$, only the powers of $x_j - b_j$ appear in any t-sparse representation of f. Moreover, any two shifted t-sparse representations of f contain the same power products of x_i and $x_j - b_j$. The last statement follows from lemma 8 by considering suitable f_W's. Consider the representation of f as

$$f = \sum_{1 \le i \le t} f_i X_{I_B}^{\alpha_i} (x-b)_{I_C}^{\beta_i}$$

where $f_i \in \mathcal{F}[\{x_k\}]$, $k \in \{1, 2, \ldots, n\} \setminus (I_B \cup I_C)$, $X_{I_B}^{\alpha_i}$ denotes a power product of X_i, $i \in I_B$, and $(x-b)_{I_C}^{\beta_i}$. Note that the degrees of the f_i in any x_k are less than $2t$. Therefore, the dense representations of f_i as $\mathcal{F}$-linear combinations of power products of x_k for $k \in \{1, 2, \ldots, n\} \setminus (I_B \cup I_C)$ have $O(t^n)$ terms. We explicitly compute the f_i by dense interpolation from their values at $O(t^n)$ points. The values of the f_i for a particular specialization $\mathcal{S}$ taking x_k to $v_k \in \mathcal{F}$ for $k \in \{1, 2, \ldots, n\} \setminus (I_B \cup I_C)$ are obtained by constructing $\mathcal{S}(f) \in \mathcal{F}[I_B \cup I_C]$ ($\mathcal{S}(f)$ denotes the image of f under the substitution $\mathcal{S}$). For any $\mathcal{S}$, $\mathcal{S}(f)$ is obtained by sparse interpolation using the Ben-Or and Tiwari algorithm (since we know the non-zero shifts b_i, this can be achieved by direct application of the Ben-Or and Tiwari algorithm). The cost of constructing the f_i this way is $((\log d) t^n)^{O(1)}$ $\mathcal{F}$-operations.

Once we have the f_i, the problem is to find shifts b_k for the x_k for $k \in \{1, 2, \ldots, n\} \setminus (I_B \cup I_C)$ such that the *total number of terms* in all the f_i represented in the power products of $x_k - b_k$ is at most t. This is done by finding a shifted t-sparse representation for the polynomial $\Psi(x_k, z) = f_1 z^{2t} + f_2 z^{2t-1} + \ldots + f_t z^{t+1}$ where z is a new unknown. In any t-sparse shift of the above polynomial, the shift with respect to z has to be 0 since its degree in z is greater than $2t-1$. To find a t-sparse shift for Ψ, we apply the algorithm from (Grigoriev-Karpinski 1993) which finds the variety $\mathcal{V}$ of all t-sparse shifts of f, i.e., set of r-tuples $(b_{k_1}, b_{k_2}, \ldots, b_{k_r})$ ($r = n - \text{cardinality}(I_B \cup I_C)$ and $\{k_1, \ldots, k_r\} = \{1, \ldots, n\} \setminus (I_B \cup I_C)$) such that the *total number of terms* in all the f_i represented in the power products of $x_k - b_k$ is at most t. The algorithm from (Grigoriev-Karpinski 1993) returns the variety $\mathcal{V}$ as a union of its irreducible components $\cup_l \mathcal{V}_l$ and for each $\mathcal{V}_l$ and i, the algorithm returns a set of exponent vectors $e_1, \ldots, e_s$ and a set of rational functions $g_1(b_k | k \in \{1, \ldots, n\} \setminus (I_B \cup I_C)), \ldots, g_s(b_k | k \in \{1, \ldots, n\} \setminus (I_B \cup I_C))$ such that $f_i = \sum_{l=1}^{s} g_l x^{e_l}$ for any $(b_{k_1}, b_{k_2}, \ldots, b_{k_r}) \in \mathcal{V}_l$. For this input, the number of operations performed by the algorithm of Grigoriev and Karpinski is bounded by $O((nt)^{O(n^2)})$ since $\deg(f_i) \le 2tn$. We now collect the main steps of the algorithm together and summarize its asymptotic time complexity in theorem 6:

Algorithm to find all multivariate sparse shifts

- *Compute the index set I_B of variables of high degree in f. If the Ben-Or and Tiwari algorithm fails to produce t-sparse g_1, g_2, g_3, g_4 satisfying lemma 9 for a variable $X_i = x_i - b$, for every $b \in \{0, 1, -1\}$, then that variable has degree $\le D$.*
- *For each variable x_j whose index j is not in I_B, construct $f_W(x_j)$ for all possible W by dense interpolation assuming that $\deg(f_W(x_j)) \le D$. Denote the interpolant by $g_W(x_j)$ and perform the sparse shift algorithm of Lakshman-Saunders on $g_W(x_j)$ whenever $\deg(g_W(x_j)) > 2t$. If each time, we discover the same shift, note that $j \in I_C$.*
- *Consider the representation of f as*

$$f = \sum_{1 \le i \le t} f_i X_{I_B}^{\alpha_i} (x-b)_{I_C}^{\beta_i}. \tag{1}$$

Determine the f_i by dense interpolation as outlined in the text and compute sparse shifts for $\Psi(x_k, z) = f_1 z^{2t} + f_2 z^{2t-1} + \ldots + f_t z^{t+1}$ by the algorithm of Grigoriev and Karpinski. For each such shift of the f_i, return the corresponding t-sparse representation f obtained by substituting the shifted sparse representations of the f_i into the representation (1) above.

Theorem 6 *The algorithm of this section computes all shifted t-sparse representations for f. The algorithm performs $O(M^{O(1)}(nt)^{O(n^2)})$ operations.* □

4. Discussion

In this paper, we have investigated the problem of finding t-sparse shifts for multivariate polynomials. The first algorithm has the advantage that the unknown shifts are obtained as zeros of univariate polynomials over algebraic extensions of $\mathcal{Q}$ (as opposed to having to solve general systems of polynomial equations). The algorithm uses Gröbner basis techniques. However, this algorithm cannot handle all polynomials with t-sparse shifts. The second algorithm is a complete algorithm slower than the first, but improves significantly over previously known algorithms. Several interesting issues remain unresolved at this time:

- find a necessary and sufficient condition for the uniqueness of t-sparse shifts and sparsifying linear transformations for multivariate polynomials.
- find an algorithm that handles the low degree case more efficiently than our algorithm. It is intriguing to see what might happen if we try to construct Gröbner bases as in the first algorithm, but with respect to other term orderings to handle the low degree case.
- find efficient algorithms for finding sparsifying linear tranformations (see Grigoriev and Karpinski, 1993). The Gröbner basis techniques can be extended to find sparsifying linear transformations for bivariate polynomials efficiently (polynomial in t, d).

It is also interesting to consider more general transformations such as ones leading to sparse decompositions of polynomials.

References

Baur, W., and Strassen, V., (1983), "The complexity of partial derivatives," Theoretical Computer Science, Vol. 22, pp. 317–330.

Becker, T., and Weispfenning, V., (1993), "Gröbner bases – a computational approach to commutative algebra," Graduate texts in Mathematics, Vol. 141, Springer-Verlag, New York.

Ben-Or, M. and Tiwari, P. (1988), "A deterministic algorithm for sparse multivariate polynomial interpolation," Proc. 20th Symp. Theory of Computing, ACM Press, pp. 301–309.

Borodin, A. and Tiwari, P. (1990), "On the decidability of sparse univariate polynomial interpolation," Proc. 22nd Symp. Theory of Computing, ACM Press, pp. 535–545.

Clausen, M., Dress, A., Grabmeier, J., Karpinski, M. (1988), "On zero testing and interpolation of k-sparse multivariate polynomials over finite fields," TR 88.06.006, IBM Germany, Heidelberg Scientific Center, June 1988.

Faugère J.C., Gianni P., Lazard D., Mora T., (1994) " Efficient Computation of Zero-dimensional Gröbner Bases by Change of Ordering ", Jour. Symb. Computation, 1994.

Grigoriev, D. Yu. and Karpinski, M. (1987), "The matching problem for bipartite graphs with polynomially bounded permanents is in NC," Proc. 28th IEEE Symp. Foundations Comp. Sci., pp. 166–172.
Grigoriev, D., Karpinski, M., and Singer, M. (1990), "Fast parallel algorithms for sparse multivariate polynomial interpolation over finite fields," SIAM J.Comp, Vol. 19, pp. 1059–1063.
Grigoriev, D., Karpinski, M., and Singer, M. (1991), "The interpolation problem for k-sparse sums of eigenfunctions of operators," Advances in Appl Math, Vol 12, pp. 76-81.
Grigoriev, D., Karpinski, M., and Odlyzko, A.M.(1992), Existence of short proofs of non-divisibility of sparse polynomials under the extended Riemann hypothesis," Proc. ISSAC 92, ACM Press, pp. 117-122.
Grigoriev, D., Karpinski, M., and Singer, M. (1993a), "Computational complexity of sparse real algebraic function interpolation," Proc. MEGA '92, Progress in Mathematics, Birkhauser, Vol. 109, pp. 91–104.
Grigoriev, D., and Karpinski, M., (1993), "A zero-test and an interpolation algorithm for the shifted sparse polynomials," Proc. AAECC-93, Lect. Notes in Comp. Sci., Vol. 673, pp. 162–169.
Grigoriev, D., Karpinski, M., and Singer, M. (1994), "Computational complexity of sparse rational interpolation," SIAM J. Comp., Vol. 23, pp. 1-11.
Kaltofen, E. (1987), "Single-factor Hensel lifting and its application to the straight-line complexity of certain polynomials," Proc. 19th Symp. Theory of Computing, ACM Press, pp. 443–452.
Kaltofen, E. and Lakshman, Y.N. (1988), "Improved sparse multivariate polynomial interpolation algorithms," Proc. ISSAC 1988, Rome, Italy, Springer-Verlag LNCS **358**, pp. 467–474.
Kaltofen, E., and Trager, B. (1990), "Computing with polynomials given by black boxes for their evaluations: Greatest common divisors, factorization, separation of numerators and denominators," J. Symb. Comp. **9** (1990), pp. 301-320.
Kaplanski, I. (1957), "An introduction to differential algebra," Hermann, Paris.
Kapur, D., and Lakshman Y.N., (1992), " Elimination methods: An introduction," in *Symbolic and Numerical Computation for Artificial Intelligence,* (Ed. Bruce Donald, Deepak Kapur, Joe Mundy), Academic Press, 1992, pp. 45–89.
Lakshman Y.N., (1990) " On the Complexity of Computing Gröbner Bases for Zero Dimensional Polynomial Ideals ", Ph.D. Thesis, Rensselaer Polytechnic Institute, Troy, New York, December 1990.
Lakshman, Y.N., and Saunders, B.D., (1993), "Sparse polynomial interpolation in non-standard bases," to appear in SIAM J. Comp.
Lakshman, Y.N., and Saunders, B.D., (1994), "On computing sparse shifts for univariate polynomials," Proc. Int. Symp. Symb. Alg. Comp., 1994 (ISSAC 94), Oxford, UK, ACM Press.
Lazard D., (1989), " Solving Zero–Dimensional Algebraic Systems ", Tech. Report.no.89–48, LITP, Universite Paris VI, June 1989.
Mansour, Y. (1992), "Randomized interpolation and approximation of sparse polynomials ," Proc. ICALP '92.
Zippel, R. (1990), "Interpolating polynomials from their values," Jour. Symb. Comp., Vol. 9, No. 3, pp. 375–403.