Reconstruction Algorithms for Sums of Affine Powers

Ignacio Garcia-Marco, Pascal Koiran, Timothée Pecatte

Reconstruction Algorithms for Sums of Affine Powers

Ignacio García-Marco, Pascal Koiran, Timothée Pecatte
LIP*, Ecole Normale Supérieure de Lyon, Université de Lyon.

October 23, 2017

Abstract

A sum of affine powers is an expression of the form

$$f(x) = \sum_{i=1}^{s} \alpha_i (x - a_i)^{e_i}.$$

Although quite simple, this model is a generalization of two well-studied models: Waring decomposition and Sparsest Shift. For these three models there are natural extensions to several variables, but this paper is mostly focused on univariate polynomials. We present structural results which compare the expressive power of the three models; and we propose algorithms that find the smallest decomposition of f in the first model (sums of affine powers) for an input polynomial f given in dense representation. We also begin a study of the multivariate case.

This work could be extended in several directions. In particular, just as for Sparsest Shift and Waring decomposition, one could consider extensions to "supersparse" polynomials and attempt a fuller study of the multivariate case. We also point out that the basic univariate problem studied in the present paper is far from completely solved: our algorithms all rely on some assumptions for the exponents e_i in a decomposition of f, and some algorithms also rely on a distinctness assumption for the shifts a_i. It would be very interesting to weaken these assumptions, or even to remove them entirely. Another related and poorly understood issue is that of the bit size of the constants a_i, α_i in an optimal decomposition: is it always polynomially related to the bit size of the input polynomial f given in dense representation?

*UMR 5668 Ecole Normale Supérieure de Lyon, CNRS, UCBL, INRIA. The authors are supported by ANR project CompA (code ANR–13–BS02–0001–01). Email: [Pascal.Koiran, Timothee.Pecatte]@ens-lyon.fr, iggarcia@ull.es.

1 Introduction

Let $\mathbb{F}$ be any characteristic zero field and let $f \in \mathbb{F}[x]$ be a univariate polynomial. This work concerns the study of expressions of f as a linear combination of powers of affine forms.

Model 1.1. *We consider expressions of f of the form:*

$$f = \sum_{i=1}^{s} \alpha_i (x - a_i)^{e_i}$$

with $\alpha_i, a_i \in \mathbb{F}$, $e_i \in \mathbb{N}$. We denote by AffPow$_{\mathbb{F}}(f)$ the minimum value s such that there exists a representation of the previous form with s terms.

This model was already studied in [9], where we gave explicit examples of polynomials of degree d requiring $\text{AffPow}_{\mathbb{R}}(f) = \Omega(d)$ terms for the field $\mathbb{F} = \mathbb{R}$.

The main goal of this work is to design algorithms that reconstruct the optimal representation of polynomials in this model, i.e., algorithms that receive as input $f \in \mathbb{F}[x]$ and compute the exact value $s = \text{AffPow}_{\mathbb{F}}(f)$ and a set of triplets of coefficients, nodes and exponents $\{(\alpha_i, a_i, e_i) \mid 1 \leq i \leq s\} \subseteq \mathbb{F} \times \mathbb{F} \times \mathbb{N}$ such that $f = \sum_{i=1}^{s} \alpha_i (x - a_i)^{e_i}$. We assume that f is given in dense representation, i.e., as a tuple of $\deg(f) + 1$ elements of $\mathbb{F}$.

Model 1.1 extends two already well-studied models. The first one is the Waring model, where all the exponents are equal to the degree of the polynomial, i.e., $e_i = \deg(f)$ for all i.

Model 1.2. *For a polynomial f of degree d, we consider expressions of f of the form:*

$$f = \sum_{i=1}^{s} \alpha_i (x - a_i)^{d}$$

with $\alpha_i, a_i \in \mathbb{F}$. We denote by Waring$_{\mathbb{F}}(f)$ the Waring rank *of f, which is the minimum value s such that there exists a representation of the previous form with s terms.*

Waring rank has been studied by algebraists and geometers since the 19th century. The algorithmic study of Model 1.2 is usually attributed to Sylvester. We refer to [15] for the historical background and to section 1.3 of that book for a description of the algorithm (see also Kleppe [17] and Proposition 46 of Kayal [18]). Most of the subsequent work was devoted to the multivariate generalization[1] of

[1] In the literature, Waring rank is usually defined for homogeneous polynomials. After homogenization, the univariate model 1.2 becomes bivariate and the "multivariate generalization" therefore deals with homogeneous polynomials in 3 variables or more.

Model 1.2, with much of the 20th century work focused on the determination of the Waring rank of generic polynomials [1, 7, 15]. A few recent papers [21, 5] have begun to investigate the Waring rank of specific polynomials such as monomials, sums of coprime monomials, the permanent and the determinant.

The second model that we generalize is the Sparsest Shift model, where all the shifts a_i are required to be equal.

Model 1.3. *For a polynomial f, we consider expressions of f of the form:*

$$f = \sum_{i=1}^{s} \alpha_i (x - a)^{e_i}$$

with $\alpha_i, a \in \mathbb{F}, e_i \in \mathbb{N}$. We denote by Sparsest$_{\mathbb{F}}(f)$ the minimum value s such that there exists a representation of the previous form with s terms.

This model and its variations have been studied in the computer science literature at least since Borodin and Tiwari [4]. Some of these papers deal with multivariate generalizations [13, 10], with "supersparse" polynomials[2] [12] or establish condition for the uniqueness of the sparsest shift [20]. It is suggested at the end of [10] to allow "multiple shifts" instead of a single shift, and this is just what we do in this paper. More precisely, as is apparent from Model 1.1, we do not place any constraint on the number of distinct shifts: it can be as high as the number s of affine powers. It would also make sense to place an upper bound k on the number of distinct shifts. This would provide a smooth interpolation between the sparsest shift model (where $k = 1$) and Model 1.1, where $k = s$.

1.1 Our results

We provide both structural and algorithmic results. Our structural results are presented in Section 3. We compare the expressive power of our 3 models: sums of affine powers, sparsest shift and the Waring decomposition. Namely, we show that some polynomials have a much smaller expression as a sum of affine powers than in the sparsest shift or Waring models. Moreover, we show that the Waring and sparsest shift models are "orthogonal" in the sense that (except in one trivial case) no polynomial can have a small representation in both models at the same time. We also show that some real polynomials have a short expression as a sum of affine powers over the field of complex numbers, but not over the field of real numbers. Finally, we study the uniqueness of the optimal representation as a sum of affine

[2] In that model, the size of the monomial x^d is defined to be $\log d$ instead of d as in the usual dense encoding.

powers. It turns out that our reconstruction algorithms only work in a regime where the uniqueness of optimal representations is guaranteed.

As already explained, we present algorithms that find the optimal representation of an input polynomial f. We achieve this goal in several cases, but we do not solve the problem in its full generality. One typical result is as follows (see Theorem 4.5 in Section 4 for a more detailed statement which includes a description of the algorithm).

Theorem 1.4. *Let $f \in \mathbb{F}[x]$ be a polynomial that can be written as*

$$f = \sum_{i=1}^{s} \alpha_i (x - a_i)^{e_i},$$

where the constants $a_i \in \mathbb{F}$ are all distinct, $\alpha_i \in \mathbb{F} \setminus \{0\}$, and $e_i \in \mathbb{N}$. Assume moreover that $n_i \leq (3i/4)^{1/3} - 1$ for all $i \geq 2$, where n_i denotes the number of indices j such that $e_j \leq i$.

Then, $\mathrm{AffPow}_{\mathbb{F}}(f) = s$. Moreover, there is a polynomial time algorithm that receives $f = \sum_{i=0}^{d} f_i x^i \in \mathbb{F}[x]$ as input and computes the s-tuples of coefficients $C(f) = (\alpha_1, \ldots, \alpha_s)$, of nodes $N(f) = (a_1, \ldots, a_s)$ and exponents $E(f) = (e_1, \ldots, e_s)$.

From the point of view of the optimality of representations, it is quite natural to assume an upper bound on the numbers n_i. Indeed, if there is an index j such that $n_j > j + 1$ then the powers $(x - a_i)^{e_i}$ are linearly dependent, and there would be a smaller expression of f as a linear combination of these polynomials.[3] We would therefore have $\mathrm{AffPow}_{\mathbb{F}}(f) < s$ instead of $\mathrm{AffPow}_{\mathbb{F}}(f) = s$. It would nonetheless be interesting to relax the assumption $n_i \leq (3i/4)^{1/3} - 1$ in this theorem. Another restriction is the assumption that the shifts a_i are all distinct. We relax that assumption in Section 5 but we still need to assume that all the exponents e_i corresponding to a given shift $a_i = a$ belong to a "small" interval (see Theorem 5.3 for a precise statement). Alternatively, we can assume instead that there is a large gap between the exponents in two consecutive occurences of the same shift as in Theorem 5.8.

In Section 6 we extend the sum of affine powers model to several variables. We consider expressions of the form

$$f(x_1, \ldots, x_n) = \sum_{i=1}^{s} \alpha_i \ell_i(x_1, \ldots, x_n)^{e_i}, \tag{1}$$

[3] It is hardly more difficult to show that one must have $n_j \leq \lceil \frac{j+1}{2} \rceil$ for any optimal expression, see [9, Proposition 18].

where $e_i \in \mathbb{N}$, $\alpha_i \in \mathbb{F}$ and ℓ_i is a (non constant) linear form for all i. This is clearly a generalization of the univariate model 1.1 and of multivariate Waring decomposition. Work on multivariate sparsest shift has developed in a different direction: one idea [10] has been to transform the input polynomial into a sparse polynomial by applying a (possibly) different shift to each variable. The model from [13] is more general than [10], and we do *not* generalize any of these two models. Our algorithmic strategy for reconstructing expressions of the form (1) is to transform the multivariate problem into univariate problems by projection, and to "lift" the solution of n different projections to the solution of the multivariate problem. This can be viewed as an analogue of "case 1" of Kayal's algorithm for Waring decomposition [18, Theorem 5].

1.2 Main tools

Most of our results[4] hinge on the study of certain differential equations satisfied by the input polynomial f. We consider differential equations of the form

$$\sum_{i=0}^{k} P_i(x) f^{(i)} = 0 \tag{2}$$

where the P_i's are polynomials. If the degree of P_i is bounded by $i + l$ for every i, we say that (2) is a *Shifted Differential Equation (SDE)* of order k and shift l. Section 2 recalls some (mostly standard) background on differential equations and the Wronskian determinant.

When f is a polynomial with an expression of size s in Model 1.1 we prove in Proposition 2.6 that f satisfies a "small" SDE, of order $2s - 1$ and shift zero. The basic idea behind our algorithms is to look for one of these "small" SDEs satisfied by f, and hope that the powers $(x - a_i)^{e_i}$ in an optimal decomposition of f satisfy the same SDE. This isn't just wishful thinking because the SDE from Proposition 2.6 is satisfied not only by f but also by the powers $(x - a_i)^{e_i}$.

Unfortunately, this basic idea by itself does not yield efficient algorithms. The main difficulty is that f could satisfy several SDE of order $2s - 1$ and shift 0. By Remark 2.7 we can efficiently find such a SDE, but what if we don't find the "right" SDE, i.e., the SDE which (by Proposition 2.6) is guaranteed to be satisfied by f *and* by the powers $(x - a_i)^{e_i}$? One way around this difficulty is to assume that the exponents e_i are all sufficiently large compared to s. In this case we can show that every SDE of order $2s - 1$ and shift 0 which is satisfied by f is also satisfied by $(x - a_i)^{e_i}$. This fact is established in Corollary 4.2, and yields the following

[4] The structural results about real polynomials from Section 3.1 rely instead on Birkhoff interpolation [9].

result (see Theorem 4.3 in Section 4 for a more detailed statement which includes a description of the algorithm).

Theorem 1.5 (Big exponents)**.** *Let $f \in \mathbb{F}[x]$ be a polynomial that can be written as*

$$f = \sum_{i=1}^{s} \alpha_i (x - a_i)^{e_i},$$

where the constants $a_i \in \mathbb{F}$ are all distinct, $\alpha_i \in \mathbb{F} \setminus \{0\}$ and $e_i > 5s^2/2$. Then, $\mathit{AffPow}_{\mathbb{F}}(f) = s$. Moreover, there is a polynomial time algorithm that receives $f = \sum_{i=0}^{d} f_i x^i \in \mathbb{F}[x]$ as input and computes the s-tuples of coefficients $C(f) = (\alpha_1, \ldots, \alpha_s)$, of nodes $N(f) = (a_1, \ldots, a_s)$ and exponents $E(f) = (e_1, \ldots, e_s)$.

The algorithm of Theorem 1.4 is more involved: contrary to Theorem 1.5, we cannot determine all the terms $(x - a_i)^{e_i}$ in a single pass. Solving the SDE only allows the determination of some (high degree) terms. We must then subtract these terms from f, and iterate.

In the first version of this paper,[5] instead of a SDE$(2s - 1, 0)$ we used a SDE of order s and shift $\binom{s}{2}$ originating from the Wronskian determinant (compare the two versions of Proposition 2.6). Switching to the new SDE led to significant improvements in most of our algorithmic results. For instance, in the first version of Theorem 1.5 the exponents e_i had to satisfy the condition $e_i > s^2(s+1)/2$ instead of the current (less stringent) condition $e_i > 5s^2/2$.

1.3 Models of computation

Our algorithms take as inputs polynomials with coefficients in an arbitrary field $\mathbb{K}$ of characteristic 0. At this level of generality, we need to be able to perform arithmetic operations (additions, multiplications) and equality tests between elements of $\mathbb{K}$. When we write that an algorithm runs in polynomial time, we mean that the number of such steps is polynomial in the input size. This is a fairly standard setup for algebraic algorithms (it is also interesting to analyze the bit complexity of our algorithms for some specific fields such as the field of rational numbers; more on this at the end of this subsection and in Section 1.4). An input polynomial of degree d is represented simply by the list of coefficients of its $d + 1$ monomials, and its size thus equals $d + 1$. In addtion to arithmetic operations and equality tests, we need to to be able to compute roots of polynomials with coefficients in $\mathbb{K}$. This is in general unavoidable: for an an optimal decomposition of $f \in \mathbb{K}[X]$ in Model 1.1, the coefficients α_i, a_i may lie in an extension field $\mathbb{F}$ of $\mathbb{K}$ (see Section 3 and more precisely Example 3.3 in Section 3.1 for the case $\mathbb{K} = \mathbb{R}, \mathbb{F} = \mathbb{C}$). If the optimal

[5] arxiv.org/abs/1607.05420v1

decomposition has size s, we need to compute roots of polynomials of degree at most $2s-1$.[6] As a rule, root finding is used only to output the nodes a_i of the optimal decomposition,[7] but the "internal working" of our algorithms remains purely rational (i.e., requires only arithmetic operations and comparisons). This is similar to the symbolic algorithm for univariate sparsest shifts of Giesbrecht, Kaltofen and Lee ([10], p. 408 of the journal paper), which also needs access to a polynomial root finder.

The one exception to this rule is the algorithm of Theorem 1.4. As mentioned at the end of Section 1.2, this is an iterative algorithm. At each step of the iteration we have to compute roots of polynomials (which may lie outside $\mathbb{K}$), and we keep computing with these roots in the subsequent iterations. For more details see Theorem 4.5 and the discussion after that theorem. We make a first step toward removing root finding from the internal working of this algorithm in Proposition 4.6.

We also take some steps toward the analysis of our algorithms in the bit model of computation. We focus on the algorithm of Theorem 1.4 since it is the most difficult to analyze due to its iterative nature. We show in Proposition 4.7 that for polynomials with integer coefficients, this algorithm can be implemented in the bit model to run in time polynomial in the bit size of the *output*. We do not have a polynomial running time bound as a function of the input size (more on this in Section 1.4). We also compute explicitly a polynomial bound on the running time of the simpler algorithm of Theorem 4.3, which deals with the case of "big exponents". Our bound is of fairly large degree and is probably not optimal.

1.4 Future work

One could try to extend the results of this paper in several directions. For instance, one could try to handle "supersparse" polynomials like in the Sparsest Shift algorithm of [12]. The multivariate case would also deserve further study. As explained above we proceed by reduction to the univariate case, but one could try to design more "genuinely multivariate" algorithms. For Waring decomposition, such an algorithm is proposed in "case 2" of [18, Theorem 5]. Its analysis relies on a randomness assumption for the input f (our multivariate algorithm is randomized, but in this paper we never assume that the input polynomial is randomized).

One should also keep in mind, however, that the basic univariate problem studied in the present paper is far from completely solved: our algorithms all rely on some assumptions for the exponents e_i in a decomposition of f, and some

[6] Except in the algorithm of Theorem 5.3, where we need to compute roots of polynomials at most $2s-1+\delta$. Here δ is a parameter of the algorithm, see Theorem 5.3 for details.

[7] Once the a_i's have been determined, we also need to do some linear algebra computations with these nodes to determine the coefficients α_i.

algorithms also rely on a distinctness assumption for the shifts a_i. It would be very interesting to weaken these assumptions, or even to remove them entirely. With a view toward this question, one could first try to improve the lower bounds from [19]. Indeed, the same tools (Wronskians, shifted differential equations) turn out to be useful for the two problems (lower bounds and reconstruction algorithms) but the lower bound problem appears to be easier. For real polynomials we have already obtained optimal $\Omega(d)$ lower bounds in [9] using Birkhoff interpolation, but it remains to give an algorithmic application of this lower bound method.

Another issue that we have only begun to address is the analysis of the bit complexity of our algorithms. We give an explicit polynomial bound on the bit complexity of the algorithm of Theorem 4.3, but this issue seems to be more subtle for Theorem 1.4 due to the iterative nature of our algorithm. It is in fact not clear that there exists a solution of size polynomially bounded in the input size (i.e., in the bit size of f given as a sum of monomials). More precisely, we ask the following question.

Question 1.6. *We define the dense size of a polynomial $f = \sum_{i=0}^{d} f_i x^i \in \mathbb{Z}[X]$ as $\sum_{i=0}^{d}[1 + \log_2(1 + |f_i|)]$. Assume that f can be written as*

$$f = \sum_{i=1}^{s} \alpha_i (x - a_i)^{e_i}$$

with $a_i \in \mathbb{Z}$, $\alpha_i \in \mathbb{Z} \setminus \{0\}$, and that this decomposition satisfies the conditions of Theorem 1.4: the constants a_i are all distinct, and $n_i \leq (3i/4)^{1/3} - 1$ for all $i \geq 2$, where n_i denotes the number of indices j such that $e_j \leq i$.

Is it possible to bound the bit size of the constants α_i, a_i by a polynomial function of the dense size of f ?

As explained at the end of Section 1.3, under the same conditions we have a decomposition algorithm that runs in time polynomial in the bit size of the *output*. It follows that the above question has a positive answer if and only if there is a decomposition algorithm that runs in time polynomial in the bit size of the input (i.e., in time polynomial in the dense size of f).

One could also ask similar questions in the case where the conditions of Theorem 1.4 do not hold. For instance, assuming that f has an optimal decomposition with integer coefficients, is there such a decomposition where the coefficients α_i, a_i are of size polynomial in the size of f ?

2 Preliminaries

In this section we present some tools that are useful for their algorithmic applications in Sections 4 and 5. Section 3 can be read independently, except for the proof of Proposition 3.11 and Theorem 5.3 which use the Wronskian.

2.1 The Wronskian

In mathematics the *Wronskian* is a tool mainly used in the study of differential equations, where it can be used to show that a set of solutions is linearly independent.

Definition 2.1 (Wronskian)**.** *For n univariate functions $f_1, \ldots, f_n$, which are $n-1$ times differentiable, the Wronskian $Wr(f_1, \ldots, f_n)$ is defined as*

$$Wr(f_1, \ldots, f_n)(x) = \begin{vmatrix} f_1(x) & f_2(x) & \ldots & f_n(x) \\ f_1'(x) & f_2'(x) & \ldots & f_n'(x) \\ \vdots & \vdots & \ddots & \vdots \\ f_1^{(n-1)} & f_2^{(n-1)} & \ldots & f_n^{(n-1)} \end{vmatrix}$$

It is a classical result, going back at least to [3], that the Wronskian captures the linear dependence of polynomials in $\mathbb{F}[x]$.

Proposition 2.2. *For $f_1, \ldots, f_n \in \mathbb{F}[X]$, the polynomials are linearly dependent if and only if the Wronskian $Wr(f_1, \ldots, f_n)$ vanishes everywhere.*

For every $f \in \mathbb{F}[x]$ and every $a \in \mathbb{F}$ we denote by $\mathrm{M}_a(f)$ the multiplicity of a as a root of f, i.e., $\mathrm{M}_a(f)$ is the maximum $t \in \mathbb{N}$ such that $(x-a)^t$ divides f. The following result from [25] gives a Wronskian-based bound on the multiplicity of a root in a sum of polynomials.

Lemma 2.3. *Let $f_1, \ldots, f_n$ be some linearly independent polynomials and $a \in \mathbb{F}$, and let $f(x) = \sum_{j=1}^{n} f_j(x)$. Then:*

$$\mathrm{M}_a(f) \leq n - 1 + \mathrm{M}_a\left(Wr(f_1, \ldots, f_n)\right),$$

where $\mathrm{M}_a(f)$ is finite since $Wr(f_1, \ldots, f_n) \not\equiv 0$.

In [23] one can find several properties concerning the Wronskian (and which have been known since the 19^{th} century). In this work we will use the following properties, which can be easily derived from those of [23]. For the sake of completeness we include a short proof.

Proposition 2.4. *Let $f_1, \ldots, f_n \in \mathbb{F}[x]$ be linearly independent polynomials and let $a \in \mathbb{F}$. If $f_j = Q_j^{d_j} g_j$ with $Q_j \in \mathbb{F}[x]$ and $d_j \geq n$ for all j, then $Q := \prod_{j=1}^{n} Q_j^{d_j - n+1}$ divides Wr$(f_1, \ldots, f_n)$. Moreover, if $Q(a) \neq 0$, then*

$$\mathrm{M}_a\left(Wr(f_1, \ldots, f_n)\right) \leq \sum_{j=1}^{n} \left[\deg(g_j) + (n-1)\deg(Q_j)\right] - \binom{n}{2}.$$

Hence, if we set $f := \sum_{j=1}^{n} f_j$, then

$$\mathrm{M}_a\left(f\right) \leq n - 1 + \sum_{j=1}^{n} \left[\deg(g_j) + (n-1)\deg(Q_j)\right] - \binom{n}{2}.$$

Proof. Consider the $n \times n$ Wronskian matrix W whose $(i+1, j)$-th entry is $f_j^{(i)}(x)$ with $0 \leq i \leq n-1$, $1 \leq j \leq n$. Since $Q_j^{d_j}$ divides f_j, then $f_j^{(i)} = Q_j^{d_j - i} g_{i,j} = Q_j^{d_j - n+1} Q_j^{n-1-i} g_{i,j}$, for some $g_{i,j} \in \mathbb{F}[x]$ of degree $\deg(g_j) + i\deg(Q_j) - i$. Since $Q_j^{d_j - n+1}$ divides every element in the j-th column of W, we can factor it out from the Wronskian. This proves that Q divides $\mathrm{Wr}(f_1, \ldots, f_n)$. Once we have factored out $Q_j^{d_j - n+1}$ for all j, we observe that $\mathrm{Wr}(f_1, \ldots, f_n) = Q(x)\, h(x)$, where $h(x)$ is the determinant of a matrix whose $(i+1, j)$-th entry has degree $\deg(g_j) + (n-1)\deg(Q_j) - i$ for all $0 \leq i \leq n-1$ and $1 \leq j \leq n$. Hence, $\deg(h) \leq \sum_{j=1}^{n} \left[\deg(g_j) + (n-1)\deg(Q_j)\right] - \binom{n}{2}$. Finally, we observe that if $Q(a) \neq 0$:

$$\mathrm{M}_a\left(\mathrm{Wr}(f_1, \ldots, f_n)\right) = \mathrm{M}_a\left(Q\right) + \mathrm{M}_a\left(h\right) = \mathrm{M}_a\left(h\right) \leq \deg(h).$$

For $f = \sum_{j=1}^{n} f_j$, the upper bound for $\mathrm{M}_a\left(f\right)$ follows directly from Lemma 2.3. □

We observe that the result holds when some of the $Q_i(x) = 1$.

2.2 Shifted Differential Equations

Definition 2.5. *A* Shifted Differential Equation *(SDE) is a differential equation of the form*

$$\sum_{i=0}^{k} P_i(x) f^{(i)}(x) = 0$$

where f is the unknown function and the P_i are polynomials in $\mathbb{F}[x]$ with $\deg(P_i) \leq i + l$.
The quantity k is called the order *of the equation, and the quantity l is called the* shift. *We will usually denote such a differential equation by SDE(k, l).*

One of the key ingredients for our results is that if $\mathrm{AffPow}(f)$ is small, then f satisfies a "small" SDE. More precisely:

Proposition 2.6. *Let $\delta \in \mathbb{Z}^+$ and let $f \in \mathbb{F}[x]$ be written as*

$$f = \sum_{i=1}^{t} Q_i(x)(x - a_i)^{e_i}.$$

where $a_i \in \mathbb{F}$, $e_i \in \mathbb{N}$ and $\deg(Q_i(x)) \leq \delta$ for all i.

Then, f satisfies a SDE$(2t-1, \delta)$ which is also satisfied by the t terms $f_i(x) = Q_i(x)(x-a_i)^{e_i}$. In particular, if AffPow$_{\mathbb{F}}(f) = s$, then f satisfies a SDE$(2s-1, 0)$.

Proof. If we can find a SDE$(2t-1, \delta)$ which is satisfied by all the f_i, by linearity the same SDE will be satisfied by f and the theorem will be proved. The existence of this common SDE is equivalent to the existence of a nonzero solution for the following linear system in the unknowns $a_{j,k}$:

$$\sum_{j,k} a_{j,k} x^j f_i^{(k)}(x) = 0,$$

where $1 \leq i \leq t$, $0 \leq k \leq 2t-1$ and $0 \leq j \leq k + \delta$. There are $(\delta+1) + (\delta+2) + \cdots + (\delta+2t) = (2\delta + 2t + 1)t$ unknowns, so we need to show that the matrix of this linear system has rank smaller than $(2\delta + 2t + 1)t$. It suffices to show that for each fixed value of $i \in \{1, \ldots, t\}$, the subsystem:

$$\sum_{j,k} a_{j,k} x^j f_i^{(k)}(x) = 0 \ (0 \leq k \leq 2t-1, 0 \leq j \leq k + \delta)$$

has a matrix of rank $< 2\delta + 2t + 1$. In other words, we have to show that the subspace V_i spanned by the polynomials $x^j f_i^{(k)}(x)$ has dimension less than $2\delta + 2t + 1$. But V_i is included in the subspace spanned by the polynomials

$$\{(x - a_i)^{e_i + j};\ -(2t-1) \leq j \leq \delta, e_i + j \geq 0\}.$$

This is due to the fact that the polynomials x^j and $Q_i(x)$ belong respectively to the spans of the polynomials $\{(x-a_i)^\ell \,|\, 0 \leq \ell \leq j\}$, and $\{(x-a_i)^\ell \,|\, 0 \leq \ell \leq \delta\}$. We conclude that $\dim V_i \leq 2t + \delta < 2\delta + 2t + 1$. □

Remark 2.7. *A polynomial f satisfies a SDE(k, l) if and only if the polynomials $(x^j f^{(i)}(x))_{0 \leq i \leq k, 0 \leq j \leq i+l}$ are linearly dependent over $\mathbb{F}$. The existence of such a SDE can therefore be decided efficiently by linear algebra, and when a SDE(k, l) exists it can be found explicitly by solving the corresponding linear system (see, e.g., [24, Corollary 3.3a] for an analysis of linear system solving in the bit model of computation). We use this fact repeatedly in the algorithms of Sections 4 and 5.*

In this paper we will use some results concerning the set of solutions of a SDE. They are particular cases of properties that apply to linear homogeneous differential equations.

Lemma 2.8. *The set of polynomial solutions of a SDE of order k is a vector space of dimension at most k.*

Given two SDE of order k:

$$\sum_{i=0}^{k} p_i(x)g^{(i)}(x) = 0 \quad \text{and} \quad \sum_{i=0}^{k} q_i(x)g^{(i)}(x) = 0,$$

we say that they are equivalent if $p_k q_i = q_k p_i$ for all $i \in \{0, \ldots, k-1\}$. The following result can be found in [23, Property 61] and will only be used in the appendix. We include a short proof.

Lemma 2.9. *For any set of $\mathbb{F}$-linearly independent polynomials $f_1, \ldots, f_k \in \mathbb{F}[x]$, there exists a unique SDE (up to equivalence) of order k satisfied simultaneously by all the f_i's.*

Proof. Suppose there exist two different SDE satisfied by $f_1, \ldots, f_k$, namely:

$$\sum_{i=0}^{k} p_i(x)g^{(i)}(x) = 0 \quad \text{and} \quad \sum_{i=0}^{k} q_i(x)g^{(i)}(x) = 0.$$

Then, we set $r_i := p_k q_i - q_k p_i$ for all $i \in \{0, \ldots, k\}$. By definition we have that $r_k = 0$ and we aim at proving that $r_i = 0$ for all i. Assume that there exists $j \in \{0, \ldots, k-1\}$ such that $r_j \neq 0$. Then, the following SDE

$$\sum_{i=0}^{k-1} r_i(x)\, g^{(i)}(x) = 0$$

has order $\leq k-1$ and is satisfied by $f_1, \ldots, f_k$, a contradiction to Lemma 2.8. □

3 Structural results

In this section we compare the expressive power of our 3 models: sums of affine powers, sparsest shift and the Waring decomposition. We will see in Section 3.2 that some polynomials have a much smaller expression as a sum of affine powers than in the sparsest shift or Waring models. Moreover, we show that the Waring

and sparsest shift models are "orthogonal" in the sense that (except in one trivial case) no polynomial can have a small representation in both models at the same time.

We begin this investigation of structural properties with the field of real numbers, where an especially strong version of orthogonality holds true. We also show that some real polynomials have a short expression as a sum of affine powers over the field of complex numbers, but not over the field of real numbers. This observation has algorithmic implications: given a polynomial $f \in \mathbb{F}[X]$, we may have to work in a field extension of $\mathbb{F}$ to find the optimal representation for f. These "real" results can be derived fairly quickly from results in our previous paper [9]. We then move to arbitrary fields of characteristic zero in Section 3.2. Finally, we study the uniqueness of optimal representations in Section 3.3. It turns out that the algorithms of Sections 4 and 5 only work in a regime where the uniqueness of optimal representations is guaranteed.

3.1 The real case

In [9] the authors considered polynomials with real coefficients and proved the following result.

Theorem 3.1. *[9, Theorem 13] Consider a polynomial identity of the form:*

$$\sum_{i=1}^{k} \alpha_i (x - a_i)^d = \sum_{i=1}^{l} \beta_i (x - b_i)^{e_i}$$

where the $a_i \in \mathbb{R}$ are distinct constants, the constants $\alpha_i \in \mathbb{R}$ are not all zero, the $\beta_i \in \mathbb{R}$ and $b_i \in \mathbb{R}$ are arbitrary constants, and $e_i < d$ for every i. Then, we must have $k + l \geq \lceil (d+3)/2 \rceil$.

Theorem 3.1 will be our main tool in Section 3.1. As a consequence of this result, we first give a sufficient condition for a polynomial to have a unique optimal expression in the model $\text{AffPow}_{\mathbb{R}}$.

Corollary 3.2. *Let $f \in \mathbb{R}[x]$ be a polynomial of the form:*

$$f = \sum_{i=1}^{s} \alpha_i (x - a_i)^{e_i} \tag{3}$$

with $\alpha_i \neq 0$. For every $e \in \mathbb{N}$ we denote by n_e the number of exponents smaller than e, i.e., $n_e = \#\{i : e_i \leq e\}$.
If $2n_e \leq \lceil (e+3)/2 \rceil$ for all $e \in \mathbb{N}$, then $\mathit{AffPow}_{\mathbb{R}}(f) = s$. Moreover, if $2n_e < \lceil (e+3)/2 \rceil$ for all e then (3) is the unique optimal expression for f.

Proof. Suppose that f can be written in another way

$$f = \sum_{j=1}^{p} \beta_j (x - b_j)^{f_j} \tag{4}$$

with $p \leq s$. Set $d = \max\left((e_i)_{1\leq i\leq s} \cup (f_j)_{1\leq j\leq p}\right)$ and denote by s' (respectively, p') the index such that $d = e_1 = \cdots = e_{s'} > e_{s'+1} \geq \cdots \geq e_s$ (respectively, $d = f_1 = \cdots = f_{p'} > f_{p'+1} \geq \cdots \geq f_p$). Note that one of the two indices s', p' will be equal to 0 if the exponent d appears only in one of the two expressions (3) and (4).

Combining equations (3) and (4), we obtain the following equality:

$$\sum_{i=1}^{s'} \alpha_i (x - a_i)^d - \sum_{j=1}^{p'} \beta_j (x - b_j)^d = -\sum_{i=s'+1}^{s} \alpha_i (x - a_i)^{e_i} + \sum_{j=p'+1}^{p} \beta_j (x - b_j)^{f_j}$$

We can rewrite this as

$$\sum_{i=1}^{k} \alpha'_i (x - a'_i)^d = \sum_{i=1}^{l} \beta'_i (x - b'_i)^{e'_i}$$

with $\alpha'_i \neq 0$, $k \leq s' + p'$ and $l \leq (s - s') + (p - p')$.
To prove the first assertion, let us assume that $2n_e \leq \lceil (e+3)/2 \rceil$ for all e. Assume also for contradiction that $p < s$ and $k > 0$. By Theorem 3.1, we must have $k + l \geq \lceil (d + 3)/2 \rceil$. The upper bounds on k and l imply $2s > s + p \geq k + l \geq \lceil (d+3)/2 \rceil$. However we have from our assumption that $2s = 2n_d \leq 2\lceil (d+3)/2 \rceil$, which contradicts the previous inequality. This shows that $p < s \Rightarrow k = 0$, i.e., if $p < s$ then the highest degree terms are the same. Continuing by induction, we find that all the terms in the two expressions are equal. In particular we would have $p = s$, a contradiction. This shows that $p \geq s$, i.e., that $\text{AffPow}_{\mathbb{R}}(f) = s$.

To prove the second assertion, let us now assume further that $2n_e < \lceil (e + 3)/2 \rceil$ for all e. Assume also that $p = s$. By Theorem 3.1, either $k = 0$ or $k + l \geq \lceil (d + 3)/2 \rceil$. In the second case, the upper bounds on k and l imply that $2s = s+p \geq k+l \geq \lceil (d+3)/2 \rceil$. This is in contradiction with the assumption that $2n_d < \lceil (d + 3)/2 \rceil$. We conclude that that k must be equal to 0, i.e., the highest degree terms are the same. Continuing by induction, we obtain that all the terms of the two decompositions are equal, thus showing that (3) is the unique optimal expression for f in this model. □

Let $\mathbb{K}$ be a field extension of $\mathbb{F}$. Theorem 1 in [20] shows that whenever the value $\text{Sparsest}_{\mathbb{K}}(f)$ is "small", then it is equal to $\text{Sparsest}_{\mathbb{F}}(f)$; more precisely, if $\text{Sparsest}_{\mathbb{K}}(f) \leq (d + 1)/2$ then $\text{Sparsest}_{\mathbb{K}}(f) = \text{Sparsest}_{\mathbb{F}}(f)$. This is no longer the case for the Affine Power model as the following example shows.

Example 3.3. *For every $d \in \mathbb{N}$, we consider the polynomial*

$$f_d := \sum_{\substack{j \equiv 3 \pmod 4 \\ 0 \leq j \leq d}} 4\binom{d}{j} x^{d-j} \in \mathbb{R}[x]. \tag{5}$$

We can express f_d as $f_d = (x+1)^d - (x-1)^d + i(x+i)^d - i(x-i)^d$, which proves that $\mathit{AffPow}_{\mathbb{C}}(f_d) \leq 4$. Moreover, in expression (5) we have $n_e \leq \lceil (e+1)/4 \rceil$ for all $e \in \mathbb{N}$. Since $2\lceil (e+1)/4 \rceil \leq \lceil (e+3)/2 \rceil$, it follows from Corollary 3.2 that this expression for f_d is optimal over the reals, i.e., $\mathit{AffPow}_{\mathbb{R}}(f_d) = \lfloor (d+1)/4 \rfloor$.

As a consequence of Theorem 3.1 we can easily derive the following result.

Corollary 3.4. *Let $f \in \mathbb{R}[x]$ be a polynomial of degree d. Either $f = \alpha(x-a)^d$ for some $\alpha, a \in \mathbb{R}$ (and $\mathit{Waring}_{\mathbb{R}}(f) = \mathit{Sparsest}_{\mathbb{R}}(f) = 1$), or the following holds:*

$$\mathit{Waring}_{\mathbb{R}}(f) + \mathit{Sparsest}_{\mathbb{R}}(f) \geq \frac{d+3}{2}$$

Proof. We set $k = \mathrm{Waring}_{\mathbb{R}}(f)$ and $l = \mathrm{Sparsest}_{\mathbb{R}}(f)$ and assume that $l \geq 2$. We write f in two different ways:

$$f = \sum_{i=1}^{k} \alpha_i (x - a_i)^d = \sum_{j=1}^{l} \beta_i (x-a)^{e_i},$$

where the $a_j \in \mathbb{R}$ are all distinct, and $e_1 < \cdots < e_l = d$. Let us move the term $\beta_l(x-a)^d$ to the left hand side of the equation. We then have two cases to consider:

- if $a \neq a_i$ for all i, we have $k+1$ terms on the left hand side of the equation and $l-1$ terms on the right hand side. Theorem 3.1 shows that $(k+1) + (l-1) \geq (d+3)/2$.
- If $a = a_i$ for some i, we have k or $k-1$ terms on the left hand side of the equation and $l-1$ terms on the right hand side. By Theorem 3.1, $k + (l-1) \geq (d+3)/2$.

□

Remark 3.5. *Consider the degree $d \geq 2$ polynomial*

$$f := (x+1)^d + (x-1)^d = \sum_{\substack{i \text{ even} \\ 0 \leq i \leq d}} 2\binom{d}{i} x^{d-i}.$$

We observe that $\mathit{Waring}_{\mathbb{R}}(f) = 2$ and $\mathit{Sparsest}_{\mathbb{R}}(f) \leq \lceil (d+1)/2 \rceil$. Hence, the inequality in Corollary 3.4 is optimal up to one unit.

A similar proofs to that of Corollary 3.4 yield the following result:

Corollary 3.6. *Let $f \in \mathbb{R}[x]$ be a polynomial of degree d. Either $\mathit{AffPow}_{\mathbb{R}}(f) = \mathit{Waring}_{\mathbb{R}}(f)$ or the following inequality holds:*

$$\mathit{Waring}_{\mathbb{R}}(f) + \mathit{AffPow}_{\mathbb{R}}(f) \geq \frac{d+3}{2}$$

3.2 Fields of characteristic zero

We now switch from the real field to an arbitrary field $\mathbb{F}$ of characteristic zero. By definition we have $\text{AffPow}_{\mathbb{F}}(f) \leq \text{Waring}_{\mathbb{F}}(f)$ and $\text{AffPow}_{\mathbb{F}}(f) \leq \text{Sparsest}_{\mathbb{F}}(f)$ for any polynomial $f \in \mathbb{F}[X]$. We show in Example 3.7 that there are polynomials f such that $\text{AffPow}_{\mathbb{F}}(f)$ is much smaller than both $\text{Waring}_{\mathbb{F}}(f)$ and $\text{Sparsest}_{\mathbb{F}}(f)$.

We first make some basic observations about Sparsest Shift. For any $a \in \mathbb{F}$, the polynomials $\{(x-a)^i \mid i \in \mathbb{N}\}$ are linearly independent, hence f can be uniquely expressed as $f = \sum_{i=0}^{d} \alpha_i (x-a)^i$ where $\alpha_i = f^{(i)}(a)/i!$. Consider such a decomposition for f, and let s be the number of nonzero terms. It follows that the $d+1-s$ derivatives $f^{(i)}$ with $\alpha_i = 0$ admit a as a common root.

Example 3.7. *For every $d \in \mathbb{N}$, we consider the polynomial $f_d := (x+1)^d - dx^{d-1} \in \mathbb{C}[x]$. It is easy to check that $\mathit{AffPow}(f_d) = 2$ for all $d \geq 2$. By [5, Proposition 3.1] we have that if $x^{d-1} = \sum_{i=1}^{s} \alpha_i (x-a_i)^d$ with $\alpha_i, a_i \in \mathbb{C}$, then $s \geq d$; and thus we get that $\mathit{Waring}_{\mathbb{C}}(f_d) \geq d-1$.*

One can easily check that for every $i \in \{0, \ldots, d-1\}$, the polynomials $f_d^{(i)} = \frac{d!}{(d-i)!} f_{d-i}$ and $f_d^{(i+1)} = \frac{d!}{(d-i-1)!} f_{d-i-1}$ do not share a common root. Consider a decomposition of f in the sparsest shift model. By the above observations, for any pair of consecutive coefficients in this decomposition at least one of the 2 coefficients is nonzero. This implies that $\mathit{Sparsest}_{\mathbb{C}}(f) \geq \lceil (d+1)/2 \rceil$.

In the remainder of Section 3.2 we give (in Proposition 3.9) a weaker version of Corollary 3.4 that works for any field of characteristic zero. Moreover, for $\mathbb{F} = \mathbb{C}$ we provide a family of polynomials showing that the bound from Proposition 3.9 is sharp.

We will use Jordan's lemma [14] (see [15, Lemma 1.35] for a recent reference), which can be restated as follows.

Lemma 3.8. *Let $d \in \mathbb{Z}^+$, $e_1, \ldots, e_t \in \{1, \ldots, d\}$, and let $a_1, \ldots, a_t \in \mathbb{F}$ be distinct constants. If $\sum_{i=1}^{t} (d+1-e_i) \leq d+1$, then the set of polynomials*

$$\{(x-a_i)^e \mid 1 \leq i \leq t,\, e_i \leq e \leq d\}$$

is linearly independent.

Proposition 3.9. *Let $f \in \mathbb{F}[x]$ be a polynomial of degree d. Either $f = \alpha(x-a)^d$ for some $\alpha, a \in \mathbb{F}$ (and $\mathit{Waring}_{\mathbb{F}}(f) = \mathit{Sparsest}_{\mathbb{F}}(f) = 1$), or the following holds:*

$$\mathit{Waring}_{\mathbb{F}}(f) \cdot \mathit{Sparsest}_{\mathbb{F}}(f) \geq d+1$$

Proof. We set $k = \text{Waring}_{\mathbb{F}}(f)$ and $l = \text{Sparsest}_{\mathbb{F}}(f)$ and assume that $k, l \geq 2$. We express f in two different ways:

$$f = \sum_{i=1}^{k} \alpha_i (x-a_i)^d = \sum_{j=1}^{l} \beta_j (x-a)^{e_j},$$

with $a_j \in \mathbb{F}$ all distinct and $e_0 := -1 < e_1 < \cdots < e_l = d$. First, we are going to prove that $e_{i+1} - e_i \leq k$ for all $i \in \{0, \ldots, l-1\}$. Indeed, if there exists $t \in \{0, \ldots, l-1\}$ such that $e_{t+1} - e_t \geq k+1$, then we set $r := e_t + 1$ and differentiate the previous equality r times to obtain

$$f^{(r)} = \sum_{i=1}^{k} \alpha_i \frac{d!}{(d-r)!}(x-a_i)^{d-r} = \sum_{j=t+1}^{l} \beta_j \frac{e_j!}{(e_j-r)!}(x-a)^{e_j-r},$$

where $e_j - r = e_j - e_t - 1 \geq e_{t+1} - e_t - 1 \geq k$ for all $j \in \{t+1, \ldots, l\}$. From this equality, we deduce that the set

$$\mathcal{B} := \{(x-a_i)^{d-r} \,|\, 1 \leq i \leq k\} \cup \{(x-a)^{e_i-r} \,|\, t+1 \leq i \leq l\}$$

is linearly dependent. However,

$$\mathcal{B} \subseteq \{(x-a_i)^{d-r} \,|\, 1 \leq i \leq k\} \cup \{(x-a)^{i} \,|\, k \leq i \leq d-r\}.$$

The $d-r+1$ polynomials on the right-hand side are of degree at most $d-r$, and they are linearly independent by Jordan's lemma. This is a contradiction since $\mathcal{B}$ is linearly dependent. We have proved that $e_{i+1} - e_i \leq k$ for all $i \in \{0, \ldots, l-1\}$, and we conclude that

$$d+1 = e_l - e_0 = \sum_{i=1}^{l} (e_i - e_{i-1}) \leq kl.$$

□

Remark 3.10. *One can slightly modify [9, Proposition 19] to obtain the following equality of complex polynomials of degree d:*

$$\sum_{j=1}^{k} (x + \xi^j)^d = \sum_{\substack{0 \leq i \leq d \\ i \equiv 0 \pmod{k}}} k \binom{d}{i} x^{d-i}$$

where $k \in \mathbb{N}$ and $\xi \in \mathbb{C}$ is a k-th primitive root of unity. This equality shows that there are polynomials of degree d such that Waring$_{\mathbb{C}}(g) \leq k$ and Sparsest$_{\mathbb{C}}(g) \leq \lceil (d+1)/k \rceil$ and, thus the bound from Proposition 3.9 is tight.

3.3 Uniqueness results for sums of affine powers

The following result is an analogue of Theorem 3.1 for polynomials with coefficients over $\mathbb{F}$, where $\mathbb{F}$ is any field of characteristic zero.

Proposition 3.11. *Consider a polynomial identity of the form:*

$$\sum_{i=1}^{k} \alpha_i (x - a_i)^d = \sum_{i=1}^{l} \beta_i (x - b_i)^{e_i}$$

where the $a_i \in \mathbb{F}$ are distinct, the $\alpha_i \in \mathbb{F}$ are not all zero, $\beta_i, b_i \in \mathbb{F}$ are arbitrary, and $e_i < d$ for every i. Then we must have $k + l > \sqrt{2(d+1)}$.

Proof. We assume $\alpha_1 \neq 0$ and we have the following equality:

$$\alpha_1 (x - a_1)^d = -\sum_{i=2}^{k} \alpha_i (x - a_i)^d + \sum_{i=1}^{l} \beta_i (x - b_i)^{e_i}$$

Consider an independent subfamily on the right hand side of this equality. We obtain a new identity of the form:

$$g = \sum_{i=1}^{p} \lambda_i \ell_i^{r_i}$$

with $g(x) = \alpha_1 (x - a_1)^d$, and $p \leq k + l - 1$. Since $\deg(g) = d$ and $e_i < d$ for all i; then there exists i such that $r_i = d$. We assume without loss of generality that $\ell_1 = x - a_2$ and $r_1 = d$.

We take the derivatives of this equality to obtain the following system:

$$\begin{aligned} g &= \sum_{i=1}^{p} \lambda_i \ell_i^{r_i} \\ g' &= \sum_{i=1}^{p} \lambda_i \left[\ell_i^{r_i}\right]' \\ &\vdots \\ g^{(p-1)} &= \sum_{i=1}^{p} \lambda_i \left[\ell_i^{r_i}\right]^{(p-1)} \end{aligned}$$

Using Cramer's rule, we obtain:

$$\lambda_1 = \frac{\mathrm{Wr}(g, \ell_2^{r_2}, \ldots, \ell_p^{r_p})}{\mathrm{Wr}(\ell_1^{r_1}, \ell_2^{r_2}, \ldots, \ell_p^{r_p})}$$

We define $\Delta = \{i : 2 \leq i \leq p, r_i \geq p\}$ and, following Proposition 2.4, we factorise the Wronskians:

$$\lambda_1 = \frac{(x-a_1)^{d-(p-1)} \prod_{i\in\Delta} \ell_i^{r_i-(p-1)} \cdot W_1}{(x-a_2)^{d-(p-1)} \prod_{i\in\Delta} \ell_i^{r_i-(p-1)} \cdot W_2}$$

where W_1, W_2 are the remaining determinants.
After some simplifications, we obtain the following identity:

$$\lambda_1 (x-a_2)^{d-(p-1)} W_2 = (x-a_1)^{d-(p-1)} W_1$$

Notice now that since we have factorised the large r_i's, the i^{th} row of W_1 and W_2 contains polynomials with degree bounded by $p-i$, thus $\deg W_1, \deg W_2 \leq p(p-1)/2$.

Moreover, since $a_1 \neq a_2$, we compute the multiplicity of a_1 on both sides of the identity and obtain that

$$\mathrm{M}_{a_1}\left((x-a_1)^{d-(p-1)} W_1\right) = \mathrm{M}_{a_1}\left(\lambda_1 (x-a_2)^{d-(p-1)} W_2\right) = \mathrm{M}_{a_1}\left(W_2\right).$$

The previous remark on the degree of W_2 therefore implies that

$$d - (p-1) \leq \frac{p(p-1)}{2}$$

Finally, we set $s = l + k$ and we use the fact that $p \leq s - 1$ to obtain the desired lower bound:

$$\begin{aligned} d &\leq \frac{(p+2)(p-1)}{2} \\ d &\leq \frac{(s+1)(s-2)}{2} \\ 2d &\leq s^2 - s - 2 \end{aligned}$$

and finally, $2(d+1) < s^2$. □

Remark 3.12. *The same equality as in Remark 3.10 shows that the order of this bound is tight when $\mathbb{F} = \mathbb{C}$, the field of complex numbers. Indeed, choosing $k = \sqrt{d+1}$ leads to the equality*

$$\sum_{i=1}^{k} (x + \xi^i)^d = \sum_{j=0}^{k-1} k \binom{d}{jk} x^{d-jk}$$

which has $2k = 2\sqrt{d+1}$ terms.

As a consequence of Proposition 3.11 we obtain that whenever $\mathrm{AffPow}_{\mathbb{F}}(f)$ is sufficiently small, the terms of highest degree in an optimal expression of f as $f = \sum_{i=1}^{s} \alpha_i (x - a_i)^{e_i}$ are uniquely determined.

Corollary 3.13. *Let $f \in \mathbb{F}[x]$ be a polynomial of the form :*

$$f = \sum_{i=1}^{k} \alpha_i (x - a_i)^d + \sum_{j=1}^{l} \beta_j (x - b_j)^{e_j}$$

with $e_j < d$. If $k + l \leq \sqrt{\frac{d+1}{2}}$, then the highest degree terms are unique. In other words, for every expression of f as

$$f = \sum_{i=1}^{k'} \alpha'_i (x - a'_i)^d + \sum_{j=1}^{l'} \beta'_j (x - b'_j)^{e'_j}$$

with $e'_j < d$ and $k' + l' \leq \sqrt{\frac{d+1}{2}}$, then $k = k'$ and there exists a permutation $\pi : \{1, \ldots, k\} \to \{1, \ldots, k\}$ such that $\alpha_i = \alpha'_{\pi(i)}$ and $a_i = a'_{\pi(i)}$ for all $i \in \{1, \ldots, k\}$.

Proof. Let us assume that we have another different decomposition for f:

$$f = \sum_{i=1}^{k'} \alpha_i'(x - a_i')^d + \sum_{j=1}^{l'} \beta_j'(x - b_j')^{e_j'}$$

with $k' + l' \leq \sqrt{(d+1)/2}$. Hence, we have the following equality:

$$\sum_{i=1}^{k} \alpha_i(x - a_i)^d - \sum_{i=1}^{k'} \alpha_i'(x - a_i')^d = \sum_{j=1}^{l} \beta_j(x - b_j)^{e_j} - \sum_{j=1}^{l'} \beta_j'(x - b_j')^{e_j'}$$

Since $k + k' + l + l' \leq \sqrt{2(d+1)}$, the result follows from Proposition 3.11. □

Finally, as a direct consequence of Corollary 3.13, we obtain a a sufficient condition for a polynomial to have a unique optimal expression in the AffPow model:

Corollary 3.14. *Let $f \in \mathbb{F}[x]$ be a polynomial of the form:*

$$f = \sum_{i=1}^{s} \alpha_i(x - a_i)^{e_i}$$

For every $e \in \mathbb{N}$ we denote by n_e the number of exponents smaller than e, i.e., $n_e = \#\{i : e_i \leq e\}$. If $n_e \leq \sqrt{\frac{e+1}{2}}$ for all $e \in \mathbb{N}$, then AffPow$_{\mathbb{F}}(f) = s$ and the optimal representation of f is unique.

Remark 3.15. *Whenever $f \in \mathbb{R}[x]$ satisfies the hypotheses of Corollary 3.14 and one term in the expression of f is of the form $\alpha_i(x - a_i)^{e_i}$ with $a_i \in \mathbb{C} - \mathbb{R}$, then there exists $j \neq i$ such that $\alpha_j = \overline{\alpha_i}, a_j = \overline{a_i}$ and $e_j = e_i$. Indeed, if we have a decomposition for f, taking the conjugate of α_i and a_i for all i gives another decomposition of f, but by Corollary 3.14 these two decompositions must be identical. In Proposition 4.6 we will prove a more general version of this fact.*

Another consequence of Proposition 3.11 is the following upper bound on the degree of the terms involved in an optimal expression of f in the model AffPow$_{\mathbb{F}}$.

Corollary 3.16. *Let $f \in \mathbb{F}[x]$ be a polynomial of degree d written as*

$$f = \sum_{i=1}^{s} \alpha_i(x - a_i)^{e_i}$$

with $\alpha_i, a_i \in \mathbb{F}$, $e_i \in \mathbb{N}$ and $s =$ AffPow$_{\mathbb{F}}(f)$. We set $e := \max\{e_i : 1 \leq i \leq s\}$, then $e < d + \frac{s^2}{2}$ and, if $\mathbb{F} = \mathbb{R}$, then $e \leq d + 2s - 2$. In particular, we have that $e < d + \frac{(d+2)^2}{8}$ and, if $\mathbb{F} = \mathbb{R}$, then $e \leq 2d$.

Proof. If $e = d$, then the result is trivial. Assume therefore that $e > d$. Now, we differentiate $d+1$ times the expression for f to obtain the identity:

$$0 = f^{(d+1)} = \sum_{e_i > d} \alpha_i \frac{e_i!}{(e_i - d - 1)!}(x - a_i)^{e_i - d - 1}.$$

By Proposition 3.11 we have $s > \sqrt{2(e-d)}$ and we conclude that $e < d + \frac{s^2}{2}$. When $\mathbb{F} = \mathbb{R}$, by Theorem 3.1 we have $s \geq (e - d + 2)/2$ and we conclude that $e \leq d + 2s - 2$. To finish the proof it suffices to recall that $s = \text{AffPow}_{\mathbb{F}}(f) \leq \lceil (d+1)/2 \rceil \leq (d+2)/2$; see [9, Proposition 18]. □

Remark 3.17. *On can find examples that are close to the bound of Corollary 3.16. Indeed, if we take $k = \sqrt{d+1}$ in Remark 3.10, we get an expression of the 0 polynomial with $2k$ terms, namely:*

$$\sum_{j=1}^{k} (x + \xi^j)^d - \sum_{\substack{0 \leq i \leq d \\ i \equiv 0 \pmod{k}}} k \binom{d}{i} x^{d-i} = 0$$

If we integrate this expression $7(d+1)$ times we get a polynomial

$$f := \sum_{j=1}^{k} (x + \xi^j)^{8d+7} - \sum_{\substack{0 \leq i \leq d \\ i \equiv 0 \pmod{k}}} k \binom{d}{i} x^{8d+7-i},$$

of degree $< 7(d+1)$ with $s := \mathit{AffPow}_{\mathbb{F}}(f) = 2k$ (by Corollary 3.14) and whose maximum exponent in the optimal expression is $8d+7 = 7(d+1)+d < \deg(f) + (s^2-4)/4$.

Remark 3.18. *As a consequence of Corollary 3.16, we obtain a naive brute force algorithm to find one optimal expression for any polynomial f. Indeed, for a fixed integer s, there are only a finite number of sequences of exponents $(e_1, \ldots, e_s)$ with $e_i \leq d + s^2/2$. For one sequence, one can try to find an expression with these exponents by solving a system of polynomial equations in $2s$ variables. The smallest s with a solution gives the value of $\mathit{AffPow}_{\mathbb{F}}(f)$.*

Also, as a byproduct of Corollary 3.16, we obtain the exact value of $\text{AffPow}_{\mathbb{F}}(f)$ for a generic polynomial f of degree d. It turns out to be equal to the worst case value of $\text{AffPow}_{\mathbb{F}}(f)$, obtained in [9, Proposition 18].

Corollary 3.19. *For a generic polynomial $f \in \mathbb{F}[x]$ of degree d, $\mathit{AffPow}_{\mathbb{F}}(f) = \lceil \frac{d+1}{2} \rceil$.*

Proof. The set of polynomials of degree $\leq d$ can be seen as a variety W of dimension $d+1$. Given $f \in \mathbb{F}[x]$ a polynomial of degree d, by [9, Proposition 18] we have $\text{AffPow}_{\mathbb{F}}(f) \leq \lceil \frac{d+1}{2} \rceil$. For $k < \lceil \frac{d+1}{2} \rceil$, let us show that the set of polynomials g of degree d such that $\text{AffPow}_{\mathbb{F}}(g) \leq k$ is contained in a variety of dimension $2k < d+1$. For every $e_1, \ldots, e_k \in \mathbb{N}$ the set of polynomials that can be written as $\sum_{i=1}^{k} \alpha_i (x - a_i)^{e_i}$ with $a_i, \alpha_i \in \mathbb{F}$ is contained in a variety $V_{e_1,\ldots,e_k}$ of dimension $2k$. If we set $M := d + \frac{(d+2)^2}{8}$, Corollary 3.16 proves that in every optimal expression of a polynomial of degree d, the exponents e_i are $\leq M$; thus the set of polynomials with $\text{AffPow}_{\mathbb{F}}(f) \leq k$ and degree d is contained in $\bigcup_{e_i \leq M} V_{e_1,\ldots,e_k}$, which is a variety of dimesion $\leq 2k$ (it is a finite union of varieties of dimension $\leq 2k$). □

4 Algorithms for distinct nodes

The goal of this and the following section is to provide algorithms that receive as input a polynomial f and computes $s = \text{AffPow}_{\mathbb{F}}(f)$ and the triplets (α_i, a_i, e_i) for $i \in \{1, \ldots, s\}$ such that $f = \sum_{i=1}^{s} \alpha_i (x - a_i)^{e_i}$. We will not able to solve the problem in all its generality but under certain hypotheses. This section concerns the case where the a_i in the optimal expression of f are all distinct. In this setting, our main result is Theorem 4.5 where we solve the problem when the number n_e of exponents in the optimal expression that are $\leq e$ is 'small'. A key point to obtain the algorithms is given by the following Proposition. Roughly speaking, this result says that if f satisfies a SDE, then every term in the optimal expression of f with exponent e_i big enough also satisfies the same SDE.

Proposition 4.1. *Let $f \in \mathbb{F}[x]$ be written as*

$$f = \sum_{i=1}^{s} \alpha_i (x - a_i)^{e_i},$$

with $\alpha_i \in \mathbb{F}$ nonzero, the $a_i \in \mathbb{F}$ are all distinct, and $e_i \in \mathbb{N}$. Whenever f satisfies a SDE(k, l), then for all $e_i \geq k + (k+l)(s-1) + \binom{s}{2}$ we have that $(x - a_i)^{e_i}$ satisfies the same SDE.

Proof. We assume that $e_1 \geq k + (k+l)(s-1) + \binom{s}{2}$ and that f satisfies the following SDE(k, l):

$$\sum_{i=0}^{k} P_i(x)\, g^{(i)}(x) = 0,$$

with $\deg(P_i) \leq i + l$. By contradiction, we assume that $(x - a_1)^{e_1}$ does not satisfy this equation. For every $j \in \{1, \ldots, s\}$, we denote by f_j and R_j the polynomials

such that

$$f_j = \sum_{i=0}^{k} P_i(x)\,((x-a_j)^{e_j})^{(i)} = R_j(x)\,(x-a_j)^{d_j},$$

where $d_j := \max\{e_j - k, 0\}$. We observe that $\deg(f_j) \leq e_j + l$, so $\deg(R_j) \leq k + l$, and that $-f_1 = \sum_{j=2}^{s} f_j \neq 0$. We consider a linearly independent subfamily of $f_2, \ldots, f_s$, namely $\{f_j \,|\, j \in J\}$ with $J = \{j_1, \ldots, j_p\} \subseteq \{2, \ldots, s\}$. Then by Proposition 2.4 we have that

$$\begin{aligned} e_1 - k = d_1 \leq \mathrm{M}_{a_1}(f_1) &\leq p - 1 + \textstyle\sum_{j\in J} \deg(R_j) + (p-1)p - \binom{p}{2} \\ &\leq p - 1 + (k+l)p + \binom{p}{2}. \end{aligned}$$

Since $p \leq s-1$, we get that $e_1 \leq k + s - 2 + (k+l)(s-1) + \binom{s-1}{2} < k + (k+l)(s-1) + \binom{s}{2}$, a contradiction. □

As a consequence of Proposition 4.1, we get Corollary 4.2 and Theorem 4.3. They provide an effective method to obtain the optimal expression of a polynomial f in the Affine Power model whenever all the terms involved have big exponents and all the nodes are different.

Corollary 4.2. *Let $f \in \mathbb{F}[x]$ be written as $f = \sum_{i=1}^{s} \alpha_i (x - a_i)^{e_i}$, with $\alpha_i \in \mathbb{F} \setminus \{0\}$, $a_i \in \mathbb{F}$ all distinct, and $e_i \geq 5s^2/2$ for all i. Then,*

a) $\{(x-a_i)^{e_i} \,|\, 1 \leq i \leq s\}$ are linearly independent,

b) If $f = \sum_{i=1}^{t} \beta_i (x - b_i)^{d_i}$ with $t \leq s$, then $t = s$ and we have the equality $\{(\alpha_i, a_i, e_i) \,|\, 1 \leq i \leq s\} = \{(\beta_i, b_i, d_i) \,|\, 1 \leq i \leq s\}$; in particular, $\mathit{AffPow}_{\mathbb{F}}(f) = s$,

c) f satisfies a SDE$(2s-1, 0)$,

d) if f satisfies a SDE$(k, 0)$ with $k \leq 2s-1$ then $(x-a_i)^{e_i}$ also satisfies it for all $i \in \{1, \ldots, s\}$, and

e) f does not satisfy any SDE$(k, 0)$ with $k < s$.

Proof. Notice first that (b) implies (a). Assume now that (b) does not hold, then there is another expression of f as $f = \sum_{i=1}^{t} \beta_i (x - b_i)^{d_i}$ with $t \leq s$. Hence, by Proposition 3.11, we get that

$$2s \geq t + s > \sqrt{2(\min(\{e_1, \ldots, e_s\}) + 1)} \geq \sqrt{5s^2},$$

a contradiction. From Proposition 2.6 we get (c). If f satisfies a SDE$(k,0)$ with $k \leq 2s-1$, then for all $i \in \{1,\ldots,s\}$ we have that

$$e_i \geq 5s^2/2 \geq (2s-1)s + \binom{s}{2} \geq ks + \binom{s}{2}.$$

Hence, Proposition 4.1 yields that $(x-a_i)^{e_i}$ is also a solution of this equation for all i, proving (d). Finally, f cannot satisfy a SDE$(k,0)$ with $k < s$; otherwise by (a) and (d), the vector space of solutions to this equation has dimension $\geq s$, which contradicts Lemma 2.8. □

Theorem 4.3 (Big exponents)**.** *Let $f \in \mathbb{F}[x]$ be a polynomial that can be written as*

$$f = \sum_{i=1}^{s} \alpha_i (x-a_i)^{e_i},$$

where the constants $a_i \in \mathbb{F}$ are all distinct, $\alpha_i \in \mathbb{F} \setminus \{0\}$ and $e_i > 5s^2/2$. Then, $\mathit{AffPow}_{\mathbb{F}}(f) = s$. Moreover, there is a polynomial time algorithm `Build`(f) *that receives $f = \sum_{i=0}^{d} f_i x^i \in \mathbb{F}[x]$ as input and computes the s-tuples of coefficients $C(f) = (\alpha_1,\ldots,\alpha_s)$, of nodes $N(f) = (a_1,\ldots,a_s)$ and exponents $E(f) = (e_1,\ldots,e_s)$. The algorithm* `Build`(f) *works as follows:*

- **Step 1.** *Take r the minimum value such that f satisfies a SDE$(r,0)$ and compute explicitly one of these SDE.*
- **Step 2.** *Compute $B = \{(x-b_i)^{d_i} \mid 1 \leq i \leq l\}$, the set of all the solutions of the SDE of the form $(x-b)^e$ with $(r+1)^2/2 \leq e \leq \deg(f) + (r^2/2)$.*
- **Step 3.** *Determine $\beta_1,\ldots,\beta_l$ such that $f = \sum_{i=1}^{l} \beta_i (x-b_i)^{d_i}$*
- **Step 4.** *Set $I := \{i \mid \beta_i \neq 0\}$ and output the sets $C(f) = (\beta_i \mid i \in I)$, $N(f) = (b_i \mid i \in I)$ and $E(f) = (d_i \mid i \in I)$.*

Proof. Corollary 4.2 proves the correctness of this algorithm. Indeed, by Corollary 4.2.(c) and (e), the value r computed in **Step 1** satisfies that $s \leq r \leq 2s-1$. We claim that the set B computed in **Step 2** satisfies that:

(1) it contains the set $\{(x-a_i)^{e_i} \mid 1 \leq i \leq s)\}$,

(2) it has at most r elements, and

(3) all its elements are $\mathbb{F}$-linearly independent.

The first claim follows from Corollary 4.2.(d), the fact that $(r+1)^2/2 \leq (2s)^2/2 < 5s^2/2$, and from Corollary 3.16, since $e_i \leq \deg(f)+(s^2/2) \leq \deg(f)+(r^2/2)$ for all i. To prove the second claim assume that B has more than r elements, then we take $t_1, \ldots, t_{r+1} \in B$. To reach a contradiction, by Lemma 2.8 it suffices to prove that $t_1, \ldots, t_{r+1}$ are linearly independent. If this were not the case, by Proposition 3.11, we would have that $r+1 > \sqrt{(r+1)^2+2}$, which is not possible. A similar argument and the fact that B has at most r elements proves the third claim. By (1) and (3), the expression of f as a combination of the elements of B is unique and is the desired one.

Finally, the four steps can be perfomed in polynomial time. Only the first two steps require a justification. See Remark 2.7 in Section 2 regarding Step 1. In Step 2 we substitute for each value of e the polynomial $(x-b)^e$ in the SDE. This yields a polynomial $g(x)$ whose coefficients are polynomials in b of degree at most $r \leq 2s-1$. We are looking for the values of b which make g identically 0, so we find b as a root of the gcd of the coefficients of g. □

In the following result we are going to analyze the bitsize complexity of the algorithm proposed; for this purpose we assume that the output (and, hence, the input) have integer coefficients. With this analysis we intend to show a rough overestimate on the number of bitsize operations showing the polynomial time nature of the algorithm.

We recall that by the dense size of a polynomial $f = \sum_{i=0}^{d} f_i x^i \in \mathbb{Z}[X]$ we mean $size(f) := \sum_{i=0}^{d}[1+\log_2(1+|f_i|)]$. Also for an $n \times m$ matrix M with rational entries p_{ij}/q_{ij} where $p_{ij} \in \mathbb{Z}$, $q_{ij} \in \mathbb{Z}^+$, the bit size of M is $size(M) := \sum_{i=1}^{n}\sum_{j=1}^{m}[1+\log_2(1+|p_{ij}|)+\log_2(1+q_{ij})]$. The notation $f(n) = \overline{\mathcal{O}}(g(n))$ means that there exists a $k \in \mathbb{N}$ such that $f(n) = \mathcal{O}(g(n)\log^k(\max(|g(n)|,2)))$.

Proposition 4.4. *Let f be a polynomial of degree d that can be written as*

$$f = \sum_{i=1}^{s} \alpha_i (x-a_i)^{e_i},$$

where the constants $a_i \in \mathbb{Z}$ are all distinct, $\alpha_i \in \mathbb{Z} \setminus \{0\}$ and $e_i > 5s^2/2$. The algorithm `Build`(f) *in Theorem 4.3 outputs the optimal expression of f in the AffPow model in $\mathcal{O}(d^{6.5}size(f)+d^8)$ bitsize operations.*

Proof. A first observation is that the value r computed in **Step 1** of the algorithm is upper bounded in terms of d. Indeed, by hypothesis $5s^2/2 \leq \max(e_i)$ and, by Corollary 3.16, $\max(e_i) \leq d+(s^2/2)$, which implies that $d \geq 2s^2$. Moreover, in Corollary 4.2 we show that $r \leq 2s-1$; this gives $r = \mathcal{O}(\sqrt{d})$. Also by Corollary 4.2, we have that $s \leq r$ and then $\max(e_i) = O(d)$.

Let us study now the number of bitsize operations needed to obtain a SDE$(r,0)$ satisfied by f assuming that we know in advance the value of r in **Step 1** of the algorithm. We propose to follow the idea of Remark 2.7 and find the SDE by computing a vector in the kernel of the matrix M whose entries are the coefficients of the polynomials $x^j f^{(i)}$ with $0 \leq j \leq i \leq r$. We have that M has $1 + \cdots + r = (r+1)r/2$ rows and $d+1$ columns. Since $size(x^j f^{(i)}) = \mathcal{O}(size(f) + id\log(d))$, we have that $size(M) = \mathcal{O}(\sum_{i=0}^{r}(i+1)(size(f) + id\log(d))) = \mathcal{O}(r^2(size(f) + rd\log(d)))$, which is $\overline{\mathcal{O}}(d\, size(f) + d^{2.5})$. Now, we can obtain the required SDE by means of the Gauss pivoting method on M. Let E be the matrix in echelon form obtained by the Gauss method. By [24, Theorem 3.3], to compute E one needs $\mathcal{O}(r^4 d)$ arithmetic operations, which is $O(d^3)$, and the biggest size of a coefficient appearing during the process of elimination by pivoting is $\mathcal{O}(size(M))$. Thus, the number of bitsize operations needed to obtain the SDE$(r,0)$ is $\overline{\mathcal{O}}(d^3\, size(M))$. Also, the biggest size of a coefficient appearing in the SDE$(r,0)$ found is $\mathcal{O}(size(M))$. After multiplying by an appropriate integer, we can assume that each of these coefficients are integers of size $\mathcal{O}(size(M))$.

We now lift the assumption that r is known in advance. To perform **Step 1** we follow Remark 2.7 and we check whether f satisfies a SDE$(\ell,0)$ starting from $\ell = 0$ and increasing ℓ. We observe that at each step, we can check if f satisfies a SDE$(\ell,0)$ by checking if the matrix M_ℓ whose rows are the coefficients of the polynomials $x^i f^{(j)}$ with $0 \leq i \leq j \leq \ell$ has full row rank. This can be easily checked from the matrix E_ℓ in echelon form obtained by applying the Gauss method to M_ℓ. Since M_ℓ and E_ℓ are respectively submatrices of $M_{\ell+1}$ and $E_{\ell+1}$, the procedure of computing the SDE of smallest order satisfied by f can be done incrementally. Moreover, all the matrices M_ℓ and E_ℓ are submatrices of the matrices M and E described above. So, it is interesting to notice that knowing the exact value of r in advance does not give any advantage and **Step 1** can be performed in $\overline{\mathcal{O}}(d^3 size(M))$ bitsize operations.

To perform **Step 2** we propose the following strategy. Assume that the SDE obtained in **Step 1** is $\sum_{i=0}^{r} P_i(x) f^{(i)}(x) = 0$. For each value $e : (r+1)^2/2 \leq e \leq d + (r^2/2)$, we input in the SDE the polynomial $(x-Y)^e$, where Y is a new variable; we obtain an equation of the form $g(x,Y) = 0$. We first observe that $g(x,Y) = (x-Y)^{e-r} h(x,Y)$, where $h(x,Y) \in \mathbb{Z}[x,Y]$ has degree $\leq r$. We write $h = \sum_{i=0}^{r} h_i x^i$, where $h_i \in \mathbb{Z}[Y]$ is of degree $\leq r-i$.

The bit size of any coefficient of h_i is $\overline{\mathcal{O}}(r^2\, size(M))$, which is $\overline{\mathcal{O}}(d\, size(M))$. Moreover, since every $h_i \in \mathbb{Z}[Y]$ has degree $\leq r$, by [6, Proposition 21.22], the cost of computing the integer roots of each h_i is $\overline{\mathcal{O}}(d^2\, size(M))$. Since we have to solve $r+1$ equations and take the common roots, this makes $\overline{\mathcal{O}}(d^{2.5} size(M))$ bitsize operations and since we have to do it for at most d values of e, this gives $\overline{\mathcal{O}}(d^{3.5} size(M))$ bit operations overall. Moreover, the b_j's computed divide the

independent term of all the h_i and, hence, the bit size of each b_i is $\overline{\mathcal{O}}(d\,size(M))$.

In **Step 3**, the corresponding matrix has at most $d+1+(r^2/2)$ rows, at most r columns (see the proof of Theorem 4.3), and its size is $\overline{\mathcal{O}}(d^{3.5}size(M))$. Since the rank of this matrix is $\leq r$ (indeed, as we proved in Theorem 4.3, this matrix has full column rank), when we are performing Gaussian elimination and treating a new row we have at most r already treated nonzero rows. As a consequence of this, we have to perform $\mathcal{O}(r^2d)$ arithmetic operations to solve the system of equations by Gaussian elimination through pivoting. Hence the cost of this step is $\overline{\mathcal{O}}(d^{5.5}size(M))$, giving a total cost of $\mathcal{O}(d^{6.5}size(f)+d^8)$ bitsize operations.

Now, we can proceed with the main result of this section:

Theorem 4.5 (Different nodes)**.** *Let $f\in\mathbb{F}[x]$ be a polynomial that can be written as*

$$f=\sum_{i=1}^{s}\alpha_i(x-a_i)^{e_i},$$

where the constants $a_i\in\mathbb{F}$ are all distinct, $\alpha_i\in\mathbb{F}\setminus\{0\}$, and $e_i\in\mathbb{N}$. Assume moreover that $n_i\leq(3i/4)^{1/3}-1$ for all $i\geq 2$, where n_i denotes the number of indices j such that $e_j\leq i$.

Then, AffPow$_{\mathbb{F}}(f)=s$. Moreover, there is a polynomial time algorithm $\texttt{Build}(f)$ *that receives $f=\sum_{i=0}^{d}f_ix^i\in\mathbb{F}[x]$ as input and computes the s-tuples of coefficients $C(f)=(\alpha_1,\ldots,\alpha_s)$, of nodes $N(f)=(a_1,\ldots,a_s)$ and exponents $E(f)=(e_1,\ldots,e_s)$. The algorithm* $\texttt{Build}(f)$ *works as follows:*

- **Step 1.** *We take t the minimum value such that f satisfies a SDE$(t,0)$ and compute explicitly one of these SDE.*
- **Step 2.** *Consider $B:=\{(x-b_i)^{d_i}\mid 1\leq i\leq l\}$, the set of all the solutions of the SDE of the form $(x-b)^e$ with $(t+1)^2/2\leq e\leq\deg(f)+\frac{(\deg(f)+2)^2}{8}$ and assume that $d_1\geq d_2\geq\cdots\geq d_l\geq d_{l+1}:=(t+1)^2/2$.*
- **Step 3.** *We take $r\in\{1,\ldots,l\}$ such that $d_r-d_{r+1}>r^2/2$ and $d_{r+1}<\deg(f)$.*
- **Step 4.** *We set $j:=d_r-(r^2/2)$ and express $f^{(j)}$ as $f^{(j)}=\sum_{i=1}^{r}\beta_i\frac{d_i!}{(d_i-j)!}(x-b_i)^{d_i-j}$ with $\beta_1,\ldots,\beta_r\in\mathbb{F}$. We set $I:=\{i\mid\beta_i\neq 0\}$.*
- **Step 5.** *We set $\widetilde{f}:=\sum_{i=1}^{r}\beta_i(x-b_i)^{d_i}$ and $h:=f-\widetilde{f}$.*

  If $h=0$, then $C(f)=(\beta_i\mid i\in I)$, $N(f)=(b_i\mid i\in I)$ and $E(f)=(d_i\mid i\in I)$.

  Otherwise, we set $h:=f-\widetilde{f}$ and we have that $C(f)=(\beta_i\mid i\in I)\cup C(h)$, $N(f)=(b_i\mid i\in I)\cup N(h)$ and $E(f)=(d_i\mid i\in I)\cup E(h)$, where the triplet $(C(h),N(h),E(h))$ is the output of $\texttt{Build}(h)$.

Proof. By Corollary 3.14 we have that $\mathrm{AffPow}_{\mathbb{F}}(f) = s$. Concerning the algorithm, first we observe that the value t computed in **Step 1** is $\leq 2s-1$ by Proposition 2.6. Moreover, we claim that the set B computed in **Step 2** has $l \leq t$ elements. Otherwise, by Lemma 2.8, there exists a set $I \subseteq \{1,\dots,l\}$ of size $\leq t+1$ and there exist $\{\gamma_i \,|\, i \in I\} \subseteq \mathbb{F} \setminus \{0\}$ such that $\sum_{i\in I} \gamma_i (x-b_i)^{d_i} = 0$. Setting $m := \max\{d_i \,|\, i \in I\} \geq (t+1)^2/2$, Proposition 3.11 yields that $t+1 \geq |I| > \sqrt{2(m+1)} > t+1$, a contradiction.

Now we set $L := 5s^2/2$ and consider the set $C := \{(x-a_i)^{e_i} \,|\, e_i \geq L\}$ where the a_i's are the nodes in the optimal expression of f. We have that $C \neq \emptyset$; indeed, if we set $e_{\max} := \max\{e_i \,|\, 1 \leq i \leq s\}$, then $s = n_{e_{\max}} \leq (3e_{\max}/4)^{1/3} - 1$ and $L \leq 4(s+1)^3/3 \leq e_{\max}$.

Since

$$ts + \binom{s}{2} \leq (2s-1)s + \binom{s}{2} \leq 5s^2/2,$$

Proposition 4.1 yields that all the elements of C are solution of the SDE and, by Corollary 3.16 we know that $e_i \leq \deg(f) + \frac{(\deg(f)+2)^2}{8}$ for all $i \in \{1,\dots,s\}$. Hence $C \subseteq B$. In particular, there exists a $\tau \in \{1,\dots,l\}$ such that $d_1 \geq d_\tau = e_{\max} \geq \frac{4}{3}(s+1)^3$.

Now we take $k := \max\{i \,|\, d_i > L\}$ (we have that $1 \leq k \leq l \leq t \leq 2s-1$) and we are going to prove that

- there exists $r \in \{\tau,\dots,k-1\}$ such that $d_r - d_{r+1} > r^2/2$, or
- $d_k - L > k^2/2$.

Indeed, if this is not the case, then we get the following contradiction:

$$\begin{aligned} \tfrac{4s^3}{3} \leq\ & \tfrac{4(s+1)^3}{3} - L \leq e_{\max} - L = d_\tau - L = \textstyle\sum_{i=\tau}^{k-1}(d_i - d_{i+1}) + d_k - L \leq \\ \leq\ & \tfrac{1}{2}\textstyle\sum_{i=\tau}^{k} i^2 \leq \tfrac{1}{2}\sum_{i=1}^{k} i^2 = \tfrac{k(k+1)(2k+1)}{12} \leq \tfrac{(2s-1)2s(4s-1)}{12} < \tfrac{4s^3}{3}. \end{aligned}$$

We take $r \in \{1,\dots,k-1\}$ such that $d_r - d_{r+1} > r^2/2$, or $r = k$ if such a value does not exist (and $d_k - L > k^2/2$). We claim that $e_{\max} \geq d_r$ if and only if $d_{r+1} < \deg(f)$ and, thus, the r described in **Step 3** always exists. If $d_{r+1} < \deg(f)$, since $\deg(f) \leq e$ and $C \subseteq B$, then $d_r \leq e_{\max}$ (since $e_{\max} = d_\tau$, it cannot be sandwiched between two consecutive elements d_r, d_{r+1} of this sequence).

Conversely, assume now that $e_{\max} \geq d_r$ and let us prove that $d_{r+1} < \deg(f)$. To prove this we first observe that setting $j := d_r - (r^2/2)$, then $f^{(j)}$ can be uniquely expressed as a linear combination of $B' := \{(x-b_i)^{d_i - j} \mid 1 \leq j \leq r\}$. Indeed, $f^{(j)} = \sum_{e_i \geq j} \alpha_i \frac{e_i!}{(e_i-j)!}(x-a_i)^{e_i-j}$ with $\alpha_i \neq 0$ and $(x-a_i)^{e_i-j} \in B'$ for all $e_i \geq j$, and if there is another way of expressing $f^{(j)}$ as a linear combination of B', then by Proposition 3.11 we get that $r > \sqrt{2(\min\{d_i \,|\, 1 \leq i \leq r\} - j + 1)} \geq$

$\sqrt{r^2+2} > r$, a contradiction. So, if $d_{r+1} \geq \deg(f)$, then $f^{(j)} = 0$ and the only expression of $f^{(j)}$ as a linear combination of B' would be the one in which every coefficient is 0, a contradiction. Hence, the value r computed in **Step 3** exists.

We have seen that $f^{(j)}$ can be uniquely expressed as a linear combination of B' as $f^{(j)} = \sum_{e_i \geq j} \alpha_i \frac{e_i!}{(e_i-j)!}(x-a_i)^{e_i-j}$. Hence, in **Step 4**, one finds all the (α_i, a_i, e_i) such that $e_i \geq j$. In **Step 5**, either the polynomial h is 0 and we have finished or $h = \sum_{e_i<j} \alpha_i (x-a_i)^{e_i}$ is written as a linear combination of strictly less than s terms and satisfies the hypotheses of the Theorem, so by induction we are done. □

Note that in Step 2 of this algorithm we need to compute polynomial roots, just as in the corresponding step of Theorem 4.3 (see the proof of Theorem 4.3 for details). One difference, however, is that we do not use the roots b_i only to output the coefficients of the optimal decomposition: we also use the b_i in the subsequent iterations of the algorithm since the polynomials $\tilde{f}$ and h of Step 5 are defined in terms of the b_i, and we call the algorithm recursively on input h. From this discussion one might be lead to think that if f has its coefficients in a subfield $\mathbb{K}$ of $\mathbb{F}$, the coefficients of $\tilde{f}$ and h may lie outside $\mathbb{K}$. We show in Proposition 4.6 that this is not the case: $\tilde{f}$ and $h = f - \tilde{f}$ always lie in $\mathbb{K}[X]$. We do not know if $\tilde{f}$ can be computed from f with a polynomial number of arithmetic operations and comparisons (in the words of Section 1.3, this would be a way to eliminate root finding from the "internal working" of the algorithm).

Proposition 4.6. *Let $\mathbb{K}$ be a subfield of $\mathbb{F}$. Let $f \in \mathbb{K}[x]$ be a polynomial that can be expressed in the AffPow$_{\mathbb{F}}$ model as*

$$f = \sum_{i=1}^{s} \alpha_i (x-a_i)^{e_i} \text{ with } \alpha_i, a_i \in \mathbb{F},$$

and $n_e = \#\{i : e_i \leq e\} \leq \sqrt{\frac{e+1}{2}}$ for all $e \in \mathbb{N}$. Then, for all $m, M \in \mathbb{N}$, the truncated expression

$$\widetilde{f} = \sum_{m \leq e_i \leq M} \alpha_i (x-a_i)^{e_i}$$

belongs to $\mathbb{K}[x]$. In particular, whenever $f \in \mathbb{F}[x]$ satisfies the hypotheses of Theorem 4.5 and $f \in \mathbb{K}[x]$, then the polynomial $\tilde{f}$ computed in **Step 5** *of the algorithm also belongs to $\mathbb{K}[x]$.*

Proof. By Corollary 3.14, we know that $\text{AffPow}_{\mathbb{F}}(f) = s$ and, hence, α_i, a_i are algebraic over $\mathbb{K}$. We denote by $\mathbb{T}$ the splitting field of the minimal polynomials of all the α_i, a_i over $\mathbb{K}$ (i.e., the smallest field $\mathbb{T}$ such that $\mathbb{K}(\alpha_i, a_i) \subset \mathbb{T}$ and

$\mathbb{K} \subset \mathbb{T}$ is normal). Since $\mathbb{K}$ is of characteristic 0 (and, thus, the extension $\mathbb{K} \subset \mathbb{T}$ is separable), then $\mathbb{K} \subset \mathbb{T}$ is a Galois extension.

Take now σ any element of the Galois group of the extension $\mathbb{K} \subset \mathbb{T}$. Since $f \in \mathbb{K}[x]$, if we apply σ to f we obtain that $f = \sigma(f) = \sum_{i=1}^{s} \sigma(\alpha_i)(x - \sigma(a_i))^{e_i}$. Moreover, by Corollary 3.14, we know that $\mathrm{AffPow}_{\mathbb{T}}(f) = s$ and f has a unique optimal expression in the $\mathrm{AffPow}_{\mathbb{T}}$ model, then $\{(\alpha_i, a_i, e_i) \,|\, 1 \leq i \leq s\} = \{(\sigma(\alpha_i), \sigma(a_i), e_i) \,|\, 1 \leq i \leq s\}$. In particular, for every $e \in \mathbb{N}$, we have that

$$\{(\alpha_i, a_i, e_i) \,|\, e_i = e\} = \{(\sigma(\alpha_i), \sigma(a_i), e_i) \,|\, e_i = e\}. \tag{6}$$

Now, we consider $\widetilde{f} = \sum_{m \leq e_i \leq M} \alpha_i (x - a_i)^{e_i}$, by (6) we get that

$$\sigma(\widetilde{f}) = \sum_{m \leq e_i \leq M} \sigma(\alpha_i)(x - \sigma(a_i))^{e_i} = \sum_{m \leq e_i \leq M} \alpha_i (x - a_i)^{e_i} = \widetilde{f}.$$

Summarizing, if we denote $\widetilde{f} = \sum_{i=m}^{M} f_i x^i \in \mathbb{T}[x]$, we have proved that $\sigma(f_i) = f_i$ for every $i \in \{0, \ldots, M\}$ and every σ in the Galois group of the extension $\mathbb{K} \subset \mathbb{T}$. This proves (see, e.g., [8, Theorem 7.1.1]) that $f_i \in \mathbb{K}$ for all $i \in \{0, \ldots, M\}$ and $\widetilde{f} \in \mathbb{K}[x]$.

□

We define the size of the set of triplets $\{(\alpha_i, a_i, e_i) \,|\, 1 \leq i \leq s\} \subset \mathbb{Z} \times \mathbb{Z} \times \mathbb{N}$ as $\sum_{i=1}^{s} [1 + \log_2(1 + |a_i|) + \log_2(1 + |\alpha_i|) + e_i]$. As mentioned in the introduction, it is not clear that the size of the output of the algorithm proposed in Theorem 4.5 is polynomially bounded in the input size (i.e., in the bit size of f given as a sum of monomials). However, it is straightforward to check that the input size is polynomially bounded by the output size. Indeed, the degree of f is upper bounded by the maximum value of the e_i and every coefficient of f can be seen as the evaluation of a small polynomial in the α_i, a_i's. In the following result we prove that the algorithm works in polynomial time in the size of the output. Hence, a positive answer to Question 1.6 together with Corollary 3.16 would directly yield that the algorithm works in polynomial time (in the size of the input).

Proposition 4.7. *Let $f \in \mathbb{Z}[x]$ be written as*

$$f = \sum_{i=1}^{s} \alpha_i (x - a_i)^{e_i}$$

with $a_i \in \mathbb{Z}$, $\alpha_i \in \mathbb{Z} \setminus \{0\}$, $e_i \in \mathbb{N}$ and assume that this decomposition satisfies the conditions of Theorem 4.5: the constants a_i are all distinct, and $n_i \leq (3i/4)^{1/3} - 1$ for all $i \geq 2$, where n_i denotes the number of indices j such that $e_j \leq i$.

Then, the algorithm in Theorem 4.5 works in polynomial time in the size of the output.

Proof. We write $f = \sum_{j=0}^{d} f_j x^j$ with $f_j \in \mathbb{Z}$ and $d = \deg(f) \leq \max\{e_1, \ldots, e_s\}$. We have that $f_j = \sum_{e_i \geq j} \alpha_i \binom{e_i}{j} a_i^{e_i - j}$ for all $j \in \{0, \ldots, d\}$. Thus, the size of f is polynomially bounded by the size of the output. To perform **Step 1** we follow Remark 2.7. We note that the coefficients of the polynomials appearing in the SDE are polynomially bounded by the size of f. In **Step 2** we have to compute the integral roots of polynomials of degree $t \leq s$ with integral coefficients, which can also be done in polynomial time (see, e.g., [22]). **Step 4** can also be performed in polynomial time by solving a linear system of equations (see, e.g., [24, Corollary 3.3a]) . The result follows from the fact that the polynomial h defined in **Step 5** can be written as $h = \sum_{j \in J} \alpha_j (x - a_j)^{e_j}$ for some set $J \subset \{1, \ldots, s\}$ of at most $s - 1$ elements. After the first iteration, the algorithm is therefore called recursively on polynomials h with an output size bounded by the output size of the original f. □

5 Algorithms for repeated nodes

This section is a continuation of the previous one and concerns the case where the nodes a_i in the optimal expression of f in the Affine Power model are not necessarily different. The section is divided in two. In the first subsection we provide algorithms when all the exponents corresponding to a repeated node appear in a small interval. The second one handles the case where the difference between two consecutive exponents corresponding to the same node is always large.

5.1 Small intervals

We begin with the following result generalizing Proposition 4.1, which corresponds to $\delta = 0$.

Proposition 5.1. *Let $\delta \in \mathbb{N}^+$ and let $f \in \mathbb{F}[x]$ be written as*

$$f = \sum_{i=1}^{t} Q_i(x) \, (x - a_i)^{e_i},$$

with distinct $a_i \in \mathbb{F}$, $Q_i \in \mathbb{F}[X]$ with $\deg(Q_i) \leq \delta$ and $e_i \in \mathbb{N}$ for all i. Assume that f satisfies the following SDE of parameters k, l:

$$\sum_{i=0}^{k} P_i(x) f^{(i)}(x) = 0.$$

If $e_i \geq k + (t-1)(k + l + \delta) + \binom{t}{2}$, then $Q_i(x) \, (x - a_i)^{e_i}$ satisfies the same SDE; as a consequence $P_k(a_i) = 0$.

Proof. We take $i = 1$. We assume that $e_1 \geq k + (t-1)(k+l+\delta) + \binom{t}{2}$ and that f satisfies a SDE(k, l)

$$\sum_{i=0}^{k} P_i(x) f^{(i)}(x) = 0$$

By contradiction, suppose that $Q_1(x)(x-a_1)^{e_1}$ does not satisfy this equation. For every $j \in \{1, \ldots, t\}$, we denote by g_j and R_j the polynomials such that

$$g_j = \sum_{i=0}^{k} P_i(x) \left(Q_j(x)(x-a_j)^{e_j}\right)^{(i)} = R_j(x)(x-a_j)^{d_j}$$

where $d_j := \max\{0, e_j - k\}$ for all j, and with $\deg R_j \leq \delta + e_j + l - d_j \leq k + l + \delta$. We have the equality

$$-g_1 = \sum_{i=2}^{t} g_i \neq 0. \tag{7}$$

We consider a linearly independent subfamily of $g_2, \ldots, g_t$, namely $\{g_j \mid j \in J\}$ with $J = \{j_1, \ldots, j_p\} \subseteq \{2, \ldots, t\}$. Then by Proposition 2.4 we have that

$$\begin{array}{rl} e_1 - k = d_1 \leq \mathrm{M}_{a_1}(g_1) & \leq p - 1 + \sum_{j \in J} \deg(R_j) + p(p-1) - \binom{p}{2} \\ & \leq p - 1 + (k+l+\delta)p + \binom{p}{2}. \end{array}$$

Taking into account that $p \leq t - 1$, we finally obtain the inequality

$$e_1 - k \leq t - 2 + (t-1)(k+l+\delta) + \binom{t-1}{2},$$

which yields a contradiction.

Now, we take $l_1 \geq e_1$ and $R_1 \in \mathbb{F}[x]$ such that $(x-a_1)^{l_i} R_1(x) = (x-a_1)^{e_1} Q_1(x)$ and $R_1(a_1) \neq 0$. Since $(x-a_1)^{e_1} Q_1(x)$ is a solution of the SDE, we have that:

$$\sum_{i=0}^{k} P_i(x) \left((x-a_1)^{l_1} R_1(x)\right)^{(i)} = 0,$$

we deduce that there exists $q \in \mathbb{F}[x]$ such that $P_k(x)(x-a_1)^{l_1-k} h(x) = (x-a_1)^{l_1-k+1} q(x)$, from where we deduce that $P_k(a_1) = 0$. □

From Proposition 5.1 we shall now derive Corollary 5.2 and Theorem 5.3. They provide an effective method to obtain the optimal expression of a polynomial f in the Affine Power model whenever all the exponents corresponding to a repeated node are required to lie in a small interval.

Corollary 5.2. *Let $\delta \in \mathbb{Z}^+$ and let $f \in \mathbb{F}[x]$ be a polynomial written as*

$$f = \sum_{i=1}^{t} Q_i(x)\,(x-a_i)^{e_i},$$

where:

- $Q_i(x) = \sum_{j=1}^{s_i} \gamma_{i,j}(x-a_i)^{\epsilon_{i,j}} \in \mathbb{F}[x]$ *with* $\gamma_{i,j} \neq 0$ *and* $0 = \epsilon_{i,0} < \epsilon_{i,1} < \cdots < \epsilon_{i,s_i} \leq \delta$,
- *the a_i's are elements of $\mathbb{F}$ and are all distinct, and*
- $e_i \geq 5t^2(\delta+1)^2/2$ *for all i.*

Then,

a) the set of polynomials $\{Q_i(x)\,(x-a_i)^{e_i} \mid 1 \leq i \leq t\}$ is linearly independent,

b) AffPow$_{\mathbb{F}}(f) = \sum_{i=1}^{t} s_i$ and the optimal representation of f is unique,

c) f satisfies a SDE$(2t-1, \delta)$,

d) if f satisfies the SDE(k, δ)

$$\sum_{i=0}^{k} P_i(x) f^{(i)}(x) = 0$$

and $k \leq 2t-1$; then $Q_i(x)(x-a_i)^{e_i}$ also satisfies it and $P_r(a_i) = 0$ for all $i \in \{1, \ldots, s\}$, and

e) f does not satisfy any SDE(k, δ) with $k < t$.

Proof. Notice that (b) implies (a). To prove (b), we observe that f is written as

$$f = \sum_{i=1}^{t} \sum_{j=1}^{s_i} \gamma_{i,j}\,(x-a_i)^{e_i+\epsilon_{i,j}},$$

so AffPow$_{\mathbb{F}}(f) \leq \sum_{i=1}^{t} s_i$. Now assume that f can also be expressed as $f = \sum_{i=1}^{r} \beta_i (x-b_i)^{d_i}$ with $\beta_i \in \mathbb{F}$ and $r \leq \sum_{i=1}^{t} s_i \leq t(\delta+1)$. By Proposition 3.11 we get that either both expressions are the same, or

$$2t(\delta+1) \geq r + \sum_{i=1}^{t} s_i \geq \sqrt{2(\min\{e_1, \ldots, e_t\}+1)} > \sqrt{5t^2(\delta+1)^2},$$

which is not possible. Thus $\text{AffPow}_{\mathbb{F}}(f) = \sum_{i=1}^{t} s_i$ and the optimal representation of f is unique.

From Proposition 2.6 we get (c). If f satisfies a SDE(k, d) with $k \leq 2t-1$, then for all $i \in \{1, \ldots, t\}$ we have that

$$\begin{aligned} e_i &\geq \tfrac{5}{2}t^2(\delta+1)^2 > \tfrac{5}{2}t^2 - \tfrac{3}{2}t + 2t\delta - 2\delta \\ &= 2t - 1 + (t-1)(2t-1+2\delta) + \tbinom{t}{2} \\ &\geq k + (t-1)(k+2\delta) + \tbinom{t}{2}. \end{aligned} \tag{8}$$

Hence, Proposition 5.1 yields that $Q_i(x)(x-a_i)^{e_i}$ is also a solution of this equation for all i, proving (d). Finally, f cannot satisfy a SDE(k, δ) with $k < t$; otherwise by (a) and (d), the vector space of solutions to this equation has dimension $\geq t$, which contradicts Lemma 2.8.

□

Theorem 5.3 (repeated nodes in small intervals)**.** *Let $\delta \in \mathbb{Z}^+$ and let $f \in \mathbb{F}[x]$ be a polynomial of degree d that can be written as*

$$f = \sum_{i=1}^{t} Q_i(x)\,(x-a_i)^{e_i},$$

with

- $Q_i(x) = \sum_{j=1}^{s_i} \gamma_{i,j}(x-a_i)^{\epsilon_{i,j}} \in \mathbb{F}[x]$ *with* $\gamma_{i,j} \neq 0$ *and* $0 = \epsilon_{i,0} < \epsilon_{i,1} < \cdots < \epsilon_{i,s_i} \leq \delta$,
- *the a_i's are elements of $\mathbb{F}$ and are all distinct, and*
- $e_i \geq \frac{5}{2}t^2(\delta+1)^2$ *for all i.*

Then $\text{AffPow}_{\mathbb{F}}(f) = \sum_{i=1}^{t} s_i$. Moreover, there is a polynomial time algorithm `Build`(f, δ) *that receives $f = \sum_{i=0}^{d} f_i x^i \in \mathbb{F}[x]$ and δ as input and computes the t-tuples of nodes $N(f) = (a_1, \ldots, a_t)$, the values $s_1, \ldots, s_t$ and the tuple of coefficients $C(f) = (\gamma_{i,j} : 1 \leq i \leq t,\, 1 \leq j \leq s_i)$, and exponents $E(f) = (e_i + \epsilon_{i,j} : 1 \leq i \leq t,\, 1 \leq j \leq s_i)$. The algorithm* `Build`$(f, \delta)$ *works as follows:*

Step 1. *Take r the minimum value such that f satisfies a SDE(r, δ). Compute explicitly one of these SDE, i.e., compute $P_0, \ldots, P_r \in \mathbb{F}[x]$ such that $\sum_{i=0}^{r} P_i(x) f^{(i)}(x) = 0$ and $\deg(P_i) \leq i + \delta$.*

Step 2. *Compute $\mathcal{R} = \{c_1, \ldots, c_p\} \subseteq \mathbb{F}$ the set of roots of P_r.*

For each $i \in \{1, \ldots, p\}$, consider the $\mathbb{F}$-vector space V_i spanned by the solutions of the SDE of the form $R(x)(x - c_i)^e$, with $\frac{(r+1)^2(\delta+1)^2}{2} < e < d + \frac{r^2(\delta+1)^2}{2}$ and $R(x)$ a polynomial of degree $\leq \delta$.

We take $B_i = \{g_{i,1}, \ldots, g_{i,l_i}\}$ a base of V_i, where $g_{i,j} = R(x)(x - c_i)^e$ with $\frac{(r+1)^2(\delta+1)^2}{2} < e < d + \frac{r^2(\delta+1)^2}{2}$ and $\deg(R(x)) \leq \delta$. We set $B := \cup_{i=1}^{p} B_i$.

Step 3. *Express f as a linear combination of the elements of B, namely, $f = \sum_{i=1}^{p} \sum_{j=1}^{l_i} \lambda_{i,j}\, g_{i,j}$ with $\lambda_{i,j} \in \mathbb{F}$.*

Step 4. *Denote $f_i = \sum_{j=1}^{l_i} \lambda_{i,j}\, g_{i,j}$, for all $i \in \{1, \ldots, p\}$. Write f_i in the shift c_i, i.e., $f_i = \sum_{j=1}^{r_i} \beta_{i,j}(x - c_i)^{\mu_{i,j}}$ with $\beta_{i,j} \in \mathbb{F} \setminus \{0\}$.*

Step 5. *Output $N(f) = (c_1, \ldots, c_p)$, $r_1, \ldots, r_p \in \mathbb{N}$, $C(f) = (\beta_{i,j} \,|\, 1 \leq i \leq p, 1 \leq j \leq r_i)$ and $E(f) = (\mu_{i,j} \,|\, 1 \leq i \leq p, 1 \leq j \leq r_i)$.*

Proof. We observe that f satisfies the hypotheses of Corollary 5.2; then, by Corollary 5.2.(b), we have that $\text{AffPow}_{\mathbb{F}}(f) = \sum_{i=1}^{t} s_i$ and that there is a unique optimal expression of f in the AffPow model.

Let us prove the correctness of the algorithm `Build`(f, δ). By Corollary 5.2.(c), (d) and (e), the value r computed in **Step 1** satisfies that $t \leq r \leq 2t - 1$. For all $i \in \{1, \ldots, t\}$ we have that

- $a_i \in \mathcal{R}$, and
- $Q_i(x)\,(x - a_i)^{e_i}$ is a solution of the SDE computed in **Step 1**

Moreover, the input polynomial f can be expressed as a linear combination of the elements of B, because:

- f can be written as a combination of $Q_i(x)\,(x - a_i)^{e_i}$.
- Since $\text{AffPow}_{\mathbb{F}}(f) = \sum_{i=1}^{t} s_i$ and $s_i \leq \delta$, we have $\text{AffPow}_{\mathbb{F}}(f) \leq t(\delta + 1) \leq r(\delta + 1)$ and thus using Corollary 3.16 we have $\max\{e_i\} < d + \frac{r^2(\delta+1)^2}{2}$. On the other hand, we have $e_i \geq \frac{5}{2}t^2(\delta + 1)^2 > 2t^2(\delta + 1)^2 \geq \frac{(r+1)^2(\delta+1)^2}{2}$. This implies that $Q_i(x)\,(x - a_i)^{e_i}$ belongs to V_i and thus can be written as a linear combination of the elements of B_i.

So, let us assume (we will prove it later) that all the elements of B are linearly independent. Then, in **Step 3** there is a unique way of writing of f as a linear combination of the elements of B. Finally, it suffices to write $f_i = R_i(x)(x -$

$c_i)^{d_i}$ and consider the Taylor expansion of $R_i(x)$ with respect to c_i for every $i \in \{1, \ldots, p\}$ as in **Step 4** to get the desired sets of nodes, coefficients and exponents.

To prove the correctness of the algorithm, it only remains to prove that the elements of B are linearly independent. To prove this we will follow a similar argument to that of Propposition 3.11. Assume that the elements of B are not linearly independent. We take $W = \{P_i(x)(x-b_i)^{d_i} \,|\, 1 \leq i \leq w\} \subset B$ a minimal $\mathbb{F}$-linearly dependent set. By Lemma 2.8, the size of this set is $w \leq r+1 \leq 2t$. Then, there exist $\lambda_1, \ldots, \lambda_w \in \mathbb{F} \setminus \{0\}$ such that $\sum_{i=1}^{w} \lambda_i P_i(x)(x-b_i)^{d_i} = 0$. We set $Z := \{i \,|\, b_i \neq b_1\}$. We also set $\tau := \min\{d_i \,|\, b_i = b_1\}$ and take $R(x)$ such that

$$R(x)(x-b_1)^{\tau} = \sum_{i \notin Z} -\lambda_i P_i(x)(x-b_1)^{d_i} = \sum_{i \in Z} \lambda_i P_i(x)(x-b_i)^{d_i}.$$

We observe that $R(x) \neq 0$ because $\{P_i(x)(x-b_1)^{d_i} \,|\, i \notin Z\}$ is $\mathbb{F}$-linearly independent. We assume that $Z = \{b_2, \ldots, b_{z+1}\}$ with $z \leq w-1 \leq r$. For all $j \in \{0, \ldots, z-1\}$, if we differentiate the above expression j times we have that

$$R_j(x)(x-b_1)^{\tau-j} = \sum_{i=2}^{z+1} \lambda_i P_{i,j}(x)(x-b_i)^{d_i-j}, \tag{9}$$

where $P_{i,j}(x) \in \mathbb{F}[x]$ are polynomials of degree $\leq \delta$. We set $g_i := P_i(x)(x-b_i)^{d_i}$ for all $i \in \{2, \ldots, z+1\}$ and apply Cramer's rule to the system of equations (9) to get that

$$\lambda_1 = \frac{\mathrm{Wr}(R(x)(x-b_1)^{\tau}, g_3, \ldots, g_{z+1})}{\mathrm{Wr}(g_2, \ldots, g_{z+1})}$$

.

We observe that

$$\mathrm{Wr}(R(x)(x-b_1)^{\tau}, g_3, \ldots, g_z) = (x-b_1)^{\tau-(z-1)} \prod_{i=3}^{z+1} (x-b_i)^{d_i-(z-1)} W_1,$$

and

$$\mathrm{Wr}(g_2, \ldots, g_z) = \prod_{i=2}^{z+1} (x-b_i)^{d_i-(z-1)} W_2,$$

where W_2 is a polynomial of degree $\leq z\delta + \frac{z(z-1)}{2}$. Thus,

$$(x-b_1)^{\tau-(z-1)} W_1 = (x-b_2)^{e_2-(z-1)} W_2$$

and, the multiplicity of b_1 in both sides of this expression has to be the same. Therefore, $\tau - (z-1) \leq \mathrm{M}_{b_1}\left((x-b_1)^{\tau-(z-1)} W_1\right) = \mathrm{M}_{b_1}\left((x-b_2)^{e_2-(z-1)} W_2\right) \leq$

$\deg(W_2) \leq z\delta + \frac{z(z-1)}{2}$, and $\tau \leq (z-1) + z\delta + \frac{z(z-1)}{2} \leq r - 1 + r\delta + \frac{r(r-1)}{2} \leq \frac{(r+1)^2}{2} + r\delta \leq \frac{(r+1)^2(\delta+1)^2}{2}$, a contradiction.

□

Remark 5.4. *The algorithm* $\texttt{Build}(f, \delta)$ *described above can be slightly modified to not receive* δ *as input as long as* f *satisfies the hypotheses of Theorem 5.3 for some* $t, \delta \in \mathbb{Z}^+$*. That is, we only need to assume that there exists* $\delta \in \mathbb{Z}^+$ *such that*

$$f = \sum_{i=1}^{t} Q_i(x)\,(x - a_i)^{e_i},$$

where $Q_i(x) \in \mathbb{F}[x]$ *has degree* $\leq \delta$*, the* a_i*'s are distinct elements of* $\mathbb{F}$*, and* $e_i \geq \frac{5}{2}t^2(\delta+1)^2$ *for all* i*. Indeed, it suffices to start with* $\delta = 0$ *and execute* $\texttt{Build}(f, \delta)$ *with increasing values of* δ *until the reconstruction of* f *succeeds. The correctness of this algorithm is justified by Corollary 5.2.(b). In fact, once we find* δ *such that the reconstruction is possible, we obtain the optimal expression of* f *in the Affine Power model.*

5.2 Big gaps

This subsection deals with polynomials f such that whenever the terms $(x-a)^e$ and $(x-a)^d$ appear in the optimal expression of f in the Affine Power model, then the difference between d and e is "large". Similarly to Section 5.1, we begin with some results ensuring that whenever f satisfies a SDE, then so do some of its terms in the optimal expression of f in the Affine Power model. The desired algorithm then follows as a consequence of these results.

Proposition 5.5. *Let* $f \in \mathbb{F}[x]$ *be written as*

$$f = (x-a)^m g(x) + \sum_{i=1}^{s} \alpha_i (x-a)^{e_i} + \sum_{i=1}^{p} \beta_i (x - a_i)^{d_i},$$

with $g \in \mathbb{F}[x], a, a_i, \alpha_i, \beta_i \in \mathbb{F}$*,* $m, e_i, d_i \in \mathbb{N}$ *and* $a_i \neq a$ *for all* i*. We set* $e := \max\{e_1, \ldots, e_s\}$ *if* $s \geq 1$ *or* $e := -1$ *if* $s = 0$*. Whenever* f *satisfies a SDE*(k, l) *with* $m - e > p + (k+l)(p+1) + \binom{p}{2}$*, then* $(x-a)^m g$ *satisfies the same SDE.*

Proof. Assume that f satisfies a SDE(k, l)

$$\sum_{i=1}^{k} P_i(x) f^{(i)}(x) = 0$$

By contradiction, we assume that $(x-a)^m g(x)$ does not satisfy this equation. Thus, there exists $T(x) \in \mathbb{F}[x]$ nonzero such that $\sum_{i=0}^{k} P_i(x)\left((x-a)^m g\right)^{(i)} = T(x)(x-a)^{m-k}$. For every $j \in \{1,\dots,s\}$ and every $j \in \{1,\dots,p\}$, we denote by h_j and g_j the polynomials such that

$$h_j = \sum_{i=0}^{k} P_i(x)\left((x-a)^{e_j}\right)^{(i)} \quad \text{and} \quad g_j = \sum_{i=0}^{k} P_i(x)\left((x-a_j)^{d_j}\right)^{(i)}.$$

We observe that $\deg(h_j) \leq e_j + l \leq e + l$ and $\deg(g_j) \leq d_j + l$. Since f satisfies the already mentioned SDE, we get that

$$T(x)(x-a)^{m-k} = \sum_{i=1}^{s} \alpha_i h_i + \sum_{i=1}^{p} \beta_i g_i.$$

If we differentiate $(e+l+1)$ times on both sides of the previous equation, we obtain an equality of the following form

$$U(x)(x-a)^{m-k-e-l-1} = \sum_{i=1}^{p} \beta_i g_i^{(e+l+1)} = \sum_{i=1}^{p} U_i(x)(x-a_i)^{r_i}$$

with $r_i := \max\{0, d_i - k - e - l - 1\}$ and $\deg(U_i(x)) \leq k + l$. If we take a linearly independent family $\{g_i^{(e+l+1)} : i \in I\} \subseteq \{g_i^{(e+l+1)} : i \in \{1,\dots,p\}\}$ and compute the multiplicity of a on both sides of the previous equality using Proposition 2.4, we obtain that

$$m - k - e - l - 1 \leq p - 1 + (k+l)p + (p-1)p - \binom{p}{2},$$

which yields that

$$m - e \leq p + (k+l)(p+1) + \binom{p}{2},$$

a contradiction. □

The following result is a generalization of Proposition 4.1 where we allow repeated nodes provided their corresponding exponents are far enough.

Corollary 5.6. *Let $f \in \mathbb{F}[x]$ be written as*

$$f = \sum_{i=1}^{s} \alpha_i (x-a)^{e_i} + \sum_{i=1}^{p} \beta_i (x-a_i)^{d_i},$$

with $a, a_i, \alpha_i, \beta_i \in \mathbb{F}$, $m, e_i, d_i \in \mathbb{N}$, $a_i \neq a$ for all i and $e_s > \cdots > e_1 > e_0 := -1$. Assume that f satisfies a SDE(k,l) and that $e_{i+1} - e_i > p + (k+l)(p+1) + \binom{p}{2}$ for all i, then $(x-a)^{e_i}$ satisfies the same SDE for all $i \in \{1,\dots,s\}$.

Proof. Assume that there exists an e_i such that $(x-a)^{e_i}$ does not satisfy the SDE(k,l) and we take e the maximum of such e_i. Then, we can write $f(x) = g(x)(x-a)^e + \sum_{e_i<e} \alpha_i (x-a)^{e_i} + \sum_{i=1}^{p} \beta_i (x-a_i)^{d_i}$. By means of Proposition 5.5 we have that $g(x)(x-a)^e$ is a solution of the same SDE. Moreover, for all $e_i > e$, then $(x-a)^{e_i}$ is also a solution of the SDE. But this is not possible since the set of solutions is a vector space, and, hence, $(x-a)^e$ would also be a solution to the same SDE. □

The proof of the following Corollary is similar to that of Corollary 4.2 but makes use of Corollary 5.6 instead of Proposition 4.1.

Corollary 5.7. *Let $f \in \mathbb{F}[x]$ be a polynomial that can be written as*

$$f = \sum_{i=1}^{s} \alpha_i (x-a_i)^{e_i}$$

with $a_i, \alpha_i \in \mathbb{F}$, $e_i > 5s^2/2$ and, whenever $a_i = a_j$ for some $1 \leq i < j \leq s$, then $|e_i - e_j| > 5s^2/2$.

a) $\{(x-a_i)^{e_i} \,|\, 1 \leq i \leq s\}$ are linearly independent,

b) If $f = \sum_{i=1}^{t} \beta_i (x-b_i)^{d_i}$ with $t \leq s$, then $t = s$ and we have the equality $\{(\alpha_i, a_i, e_i) \,|\, 1 \leq i \leq s\} = \{(\beta_i, b_i, d_i) \,|\, 1 \leq i \leq s\}$; in particular, AffPow$_{\mathbb{F}}(f) = s$,

c) f satisfies a SDE$(2s-1, 0)$,

d) if f satisfies a SDE$(k,0)$ with $k \leq 2s-1$, then $(x-a_i)^{e_i}$ also satisfies it for all $i \in \{1, \ldots, s\}$, and

e) f does not satisfy any SDE$(k,0)$ with $k < s$.

From this corollary we get the following result whose proof is similar to that of Theorem 4.3.

Theorem 5.8 (Big gaps)**.** *Let $f \in \mathbb{F}[x]$ be a polynomial that can be written as*

$$f = \sum_{i=1}^{s} \alpha_i (x-a_i)^{e_i}$$

with $a_i, \alpha_i \in \mathbb{F}$, $e_i > 5s^2/2$ and whenever $a_i = a_j$ for some $1 \leq i < j \leq s$, then $|e_i - e_j| > 5s^2/2$. Then, AffPow$_{\mathbb{F}}(f) = s$. Moreover, there is a polynomial time algorithm `Build`(f) *that receives $f = \sum_{i=0}^{d} f_i x^i \in \mathbb{F}[x]$ as input and computes the s-tuples of nodes $N(f) = (a_1, \ldots, a_s)$, coefficients $C(f) = (\alpha_1, \ldots, \alpha_s)$ and exponents $E(f) = (e_1, \ldots, e_s)$. The algorithm* `Build`$(f)$ *works as follows:*

Step 1. *Take r the minimum value such that f satisfies a SDE$(r, 0)$ and compute explicitly one of these SDE.*

Step 2. *Compute $B = \{(x - b_i)^{d_i} \mid 1 \leq i \leq t\}$, the set of all the solutions of the SDE of the form $(x - b)^d$ with $(r + 1)^2/2 \leq e \leq \deg(f) + (r^2/2)$.*

Step 3. *Determine $\alpha_1, \ldots, \alpha_r$ such that $f = \sum_{i=1}^{r} \alpha_i (x - b_i)^{d_i}$*

Step 4. *Output the sets $C(f) = (\alpha_1, \ldots, \alpha_r)$, $N(f) = (b_1, \ldots, b_r)$ and $E(f) = (d_1, \ldots, d_r)$.*

6 The multivariate case

This section concerns the study of the multivariate version of the Affine Power model, i.e., we study expressions of a polynomial $f \in \mathbb{F}[x_1, \ldots, x_n]$ as

$$f = \sum_{i=1}^{s} \alpha_i \ell_i^{e_i}, \tag{10}$$

where $e_i \in \mathbb{N}$, $\alpha_i \in \mathbb{F}$ and ℓ_i is a (non constant) linear form for all i. We denote by $\text{AffPow}_{\mathbb{F}}(f)$ the minimum value s such that there exists a representation of the previous form with s terms. We will study the uniqueness of optimal representations and propose an algorithm for finding such representations. *In this section only, we work in the black box model:* we assume that our algorithm has access to f only through a "black box" that outputs $f(x_1, \ldots, x_n)$ when queried on an input $(x_1, \ldots, x_n) \in \mathbb{F}^n$. This very general model is standard for the study of many problems about multivariate polynomials such as, e.g., factorization [16], sparse interpolation [2, 11], sparsest shift [10] or Waring decomposition [18]. We also assume that our algorithm has access to $d = \deg(f)$; the knowledge of an upper bound on $\deg(f)$ would in fact suffice. As explained in the introduction, our algorithm proceeds by reduction to the univariate case: we solve n univariate projections of the multivariate problem, and then "lift" them to a solution of the multivariate problem. One (very) minor difficulty is that our univariate algorithms are presented for polynomials given in dense representation rather than in black box representation. But it is easy to convert from black box to dense representation:

Remark 6.1. *Suppose that we have black-box access to a polynomial $f(x_1, \ldots, x_n)$ of degree d. We can obtain the dense representation of the univariate polynomial $f_1(x_1) = f(x_1, 0, 0, \ldots, 0)$ by querying f on $d + 1$ distinct inputs of the form $(a_i, 0, \ldots, 0)$ and interpolating f_1 from its values at $a_0, \ldots, a_d$.*

In our algorithm we perform a random change of coordinates before projecting to a univariate problem. Converting to dense representation in this case is hardly more difficult:

Remark 6.2. *Suppose that we have black-box access to a polynomial $f(x_1, \ldots, x_n)$ of degree d. Let $g(x) = f(\Lambda.x + \lambda)$, where $\lambda = (\lambda_1, \ldots, \lambda_n) \in \mathbb{F}^n$ and $\Lambda = (\lambda_{ij})$ is an $n \times n$ matrix.*

We can obtain the dense representation of the univariate polynomial $g_1(x_1) = g(x_1, 0, 0, \ldots, 0) = f(\lambda_{11}x_1 + \lambda_1, \lambda_{21}x_1 + \lambda_2, \ldots, \lambda_{n1}x_1 + \lambda_n)$ by evaluating g_1 at $d + 1$ points and interpolating from those values. Equivalently, we can observe that a black-box for g can be constructed from the black box for f, and we can therefore apply Remark 6.1 to g.

Having recalled these well-known facts, we proceed with uniqueness considerations. Strictly speaking the optimal expressions in model (10) are never unique since for all $\lambda \in \mathbb{F} \setminus \{0\}$ we have $\alpha_i \ell_i^{e_i} = \beta_i t_i^{e_i}$ with $\beta_i := \alpha_i \lambda^{e_i}$ and $t_i := \ell_i / \lambda$. To deal with this ambiguity, we use the notion of *essentially equal* expressions. Given f we say that two expressions of $f = \sum_{i=1}^{s} \alpha_i \ell_i^{e_i} = \sum_{i=1}^{r} \beta_i t_i^{d_i}$ are essentially equal if $r = s$ and there exists a permutation σ of $\{1, \ldots, s\}$ such that $\alpha_i \ell_i^{e_i} = \beta_{\sigma(i)} t_{\sigma(i)}^{d_{\sigma(i)}}$ for all $i \in \{1, \ldots, s\}$. Likewise, we say that f has an *essentially unique* optimal decomposition in the multivariate Affine Powers model if two optimal decompositions of f are always essentially equal.

If the representation of $f = \sum_{i=1}^{s} \alpha_i \ell_i^{e_i}$ is optimal, ℓ_i and ℓ_j cannot be proportional whenever $e_i = e_j$. Otherwise if $\ell_i = \lambda \ell_j$ with $\lambda \in \mathbb{F}$, we can rewrite $\alpha_i \ell_i^{e_i} + \alpha_j \ell_j^{e_j} = (\lambda^{e_i} \alpha_i + \alpha_j) \ell_j^{e_j}$

The following result provides a sufficient condition for f to have an essentially unique optimal decomposition in the multivariate Affine Powers model. Indeed, it is an extension to the multivariate setting of Corollary 3.14.

Proposition 6.3. *Let $f \in \mathbb{F}[x_1, \ldots, x_n]$ be a polynomial of the form:*

$$f = \sum_{i=1}^{s} \alpha_i \ell_i^{e_i}$$

where $\alpha_i \in \mathbb{F} \setminus \{0\}$, the ℓ_i are non constant linear forms, and ℓ_i is not proportional to ℓ_j whenever $e_i = e_j$. For every $e \in \mathbb{N}$ we denote by n_e the number of exponents smaller than e, i.e., $n_e = \#\{i : e_i \leq e\}$. If $n_e \leq \sqrt{\frac{e+1}{2}}$ for all $e \in \mathbb{N}$, then $\mathit{AffPow}_{\mathbb{F}}(f) = s$ and the optimal representation of f is essentially unique.

Proof. Let $r := \mathrm{AffPow}_{\mathbb{F}}(f) \leq s$ and let $f = \sum_{i=s+1}^{s+r} \alpha_i \ell_i^{e_i}$ be an optimal representation of f. We write $\ell_i = \sum_{j=1}^{n} a_{ij} x_j + a_{i0}$ for all $i \in \{1, \ldots, s + r\}$.

Consider the ring homomorphism $\varphi : \mathbb{F}[x_1, \ldots, x_n] \to \mathbb{F}[x]$ induced by $x_i \mapsto \omega_i x + \lambda_i$ where $\omega = (\omega_1, \ldots, \omega_n), \lambda = (\lambda_1, \ldots, \lambda_n) \in \mathbb{F}^n$. If we write $\varphi(\ell_i) = b_i x + c_i$, we choose ω and λ such that

(1.a) $b_i \neq 0$ and $c_i \neq 0$ for all $i \in \{1, \ldots, r+s\}$, and

(1.b) for all $1 \leq i < j \leq s+r$, $\varphi(\ell_i) = \mu\varphi(\ell_j)$ with $\mu \in \mathbb{F}$ if and only if $\ell_i = \mu\ell_j$.

It is important to observe that a generic choice of $\omega, \lambda \in \mathbb{F}^n$ fulfils these two conditions. Then

$$\begin{array}{rclcl} \varphi(f) & = & \sum_{i=1}^{s} \alpha_i \varphi(\ell_i)^{\,e_i} & = & \sum_{i=1}^{s} \alpha_i b_i^{e_i} (x + c_i/b_i)^{\,e_i} \\ & = & \sum_{i=s+1}^{s+r} \alpha_i \varphi(\ell_i)^{\,e_i} & = & \sum_{i=s+1}^{s+r} \alpha_i b_i^{e_i} (x + c_i/b_i)^{\,e_i}. \end{array}$$

We consider the expression $\varphi(f) = \sum_{i=1}^{s} \alpha_i b_i^{e_i} (x + c_i/b_i)^{\,e_i}$ in the univariate Affine Power model. By (1.b), whenever $e_i = e_j$ then $c_i/b_i \neq c_j/b_j$. Moreover it satisfies that $\{i \in \{1, \ldots, s\} : e_i \leq e\} = n_e \leq \sqrt{\frac{e+1}{2}}$ for all $e \in \mathbb{N}$. Hence we apply Corollary 3.14 to get that $r \geq \text{AffPow}_{\mathbb{F}}(\varphi(f)) = s \geq r$ and that both expressions for $\varphi(f)$ are the same. After reindexing if necessary we get that

(2.a) $\alpha_i b_i^{e_i} = \alpha_{i+s} b_{i+s}^{e_{i+s}}$,

(2.b) $c_i/b_i = c_{i+s}/b_{i+s}$,

(2.c) and $e_i = e_{i+s}$ for all $i \in \{1, \ldots, s\}$.

By (2.b) we have that $b_i x + c_i = \mu(b_{i+s} x + c_{i+s})$ with $\mu := b_i/b_{i+s}$. By (1.b) we have that $\ell_i = \mu\ell_{i+s}$. Finally, by (2.a) and (2.c), we conclude that

$$\alpha_i \ell_i^{e_i} = \alpha_i \mu^{\,e_i} \ell_{i+s}^{e_i} = \alpha_{i+s} \ell_{i+s}^{e_{i+s}},$$

proving that the optimal representation of f is essentially unique. □

Our next goal is to provide algorithms that, given black-box access to a polynomial $f \in \mathbb{F}[x_1, \ldots, x_n]$, compute $s = \text{AffPow}_{\mathbb{F}}(f)$ and the terms $\alpha_i \ell_i^{e_i}$ for $i \in \{1, \ldots, s\}$ such that $f = \sum_{i=1}^{s} \alpha_i \ell_i^{e_i}$. We are going to prove a multivariate analogue of Theorem 4.5 where the condition of "distinct nodes" is replaced by "the ℓ_i's in the decomposition are not proportional". The same strategy that we are going to exhibit in the proof also applies to obtain similar results for the other algorithms of sections 4 and 5.

Theorem 6.4. *Let $f \in \mathbb{F}[x_1, \ldots, x_n]$ be a polynomial that can be written as*

$$f = \sum_{i=1}^{s} \alpha_i \ell_i^{e_i},$$

where ℓ_i are nonconstant linear forms such that $\ell_i \neq \lambda \ell_j$ for all $\lambda \in \mathbb{F}$, $1 \leq i < j \leq s$, $\alpha_i \in \mathbb{F} \setminus \{0\}$, and $e_i \in \mathbb{N}$. Assume that $n_i \leq (3i/4)^{1/3} - 1$ for all $i \geq 2$, where n_i denotes the number of indices j such that $e_j \leq i$. Then, AffPow$_{\mathbb{F}}(f) = s$.

Moreover, there is a randomized algorithm `MultiBuild`(f) *that, given access to a black box for f and to $d = \deg(f)$, computes the set of terms $T(f) = \{\alpha_i \ell_i^{e_i} \mid 1 \leq i \leq s\}$. The algorithm* `MultiBuild`$(f)$ *runs in time polynomial in n and d, and works as follows:*

Step 1. *We define $g := \phi(f)$ where ϕ is a random affine change of coordinates ($x_i \mapsto \sum_{j=1}^{n} \lambda_{ij} x_j + \lambda_i$ for all i).*

Step 2. *For each $j \in \{1, \ldots, n\}$, we set $g_j := \pi_j(g)$ where $\pi_j : \mathbb{F}[x_1, \ldots, x_n] \longrightarrow \mathbb{F}[x]$ is induced by $x_k \mapsto 0$ if $k \neq j$ and $x_j \mapsto x$.*

We apply the algorithm `Build`(g_j) *from Theorem 4.5 to obtain $s_j :=$ AffPow$_{\mathbb{F}}(g_j)$ and the triplets $(\beta_{ij}, b_{ij}, e_{ij}) \in \mathbb{F} \times \mathbb{F} \times \mathbb{N}$ such that $g_j = \sum_{i=1}^{s_j} \beta_{ij}(x + b_{ij})^{e_{ij}}$.*

If there exist i, j such that $b_{ij} = 0$, then output 'It is not possible to reconstruct f'. Otherwise, for all j we define the set of triplets

$$P_j := \{(c_{ij}, p_{ij}, e_{i,j}) \mid c_{ij} := \beta_{ij} b_{ij}^{e_{ij}},\ p_{ij} := 1/b_{ij},\ 1 \leq i \leq s_i\}.$$

Step 3. *If one of these conditions holds:*

(a) there exist $j_1 \neq j_2$ such that $s_{j_1} \neq s_{j_2}$,

(b) there exist $i_1 \neq i_2$ and j such that $c_{i_1 j} = c_{i_2 j}$, or

(c) there exist i, j such that for all i', $c_{i1} \neq c_{i'j}$ or $e_{i1} \neq e_{i'j}$;

then output: 'It is not possible to reconstruct f'. Otherwise we set $s := s_1 = s_2 = \cdots = s_r$ and reorder the elements of $P_2, \ldots, P_n$ so that $c_i := c_{i1} = c_{i2} = \cdots = c_{in}$ and $e_i := e_{i1} = e_{i2} = \cdots = e_{in}$ for all $i \in \{1, \ldots, s\}$.

Step 4. *$g = \sum_{i=1}^{s} c_i (1 + \sum_{j=1}^{n} p_{ij} x_j)^{e_i}$, so we output $f = \sum_{i=1}^{s} c_i (\phi^{-1}(1 + \sum_{j=1}^{n} p_{ij} x_j))^{e_i}$*

If the λ_i's and the λ_{ij}'s needed to define ϕ are chosen uniformly at random from a finite set S, then the probability of success of the algorithm is at least

$$1 - \frac{d^{2/3}(2n + d)}{|S|}.$$

Proof. The input polynomial f satisfies the hypotheses of Proposition 6.3, so $\mathrm{AffPow}_{\mathbb{F}}(f) = s$ and the optimal representation of f is essentially unique.

After applying a random ϕ as described in **Step 1**, with high probability[8] we have that ϕ is invertible and $g = \sum_{i=1}^{s} \alpha_i t_i^{e_i}$ with $t_i = \sum_{j=1}^{n} a_{ij} x_j + a_{i0}$ satisfies the following properties:

(i) $a_{ij} \neq 0$ for all i, j.

(ii) for all $j \neq 0$, then $a_{ij}/a_{i0} \neq a_{i'j}/a_{i'0}$ for all i, i', and

(iii) $\alpha_i a_{i0}^{e_i} \neq \alpha_{i'} a_{i'0}^{e_{i'}}$ for all $i \neq i'$.

It is important to observe that for a generic choice of the λ_i's and λ_{ij}'s involved in the definition of ϕ, these conditions will be fulfilled. The goal of the algorithm is to recover f via the following expression of g:

$$g = \sum_{i=1}^{s} \alpha_i a_{i0}^{e_i} \left(1 + \sum_{j=1}^{n} \frac{a_{ij}}{a_{i0}} x_j\right)^{e_i};$$

so we are interested in computing the values

- $\alpha_i a_{i0}^{e_i}$ for all i
- a_{ij}/a_{i0} for all i, j
- e_i for all i

In **Step 2**, for all $j \in \{1, \ldots, n\}$ we consider

$$\pi_j(g) = \sum_{i=1}^{s} \alpha_i a_{i0}^{e_i} \left(1 + \frac{a_{ij}}{a_{i0}}\, x\right)^{e_i} = \sum_{i=1}^{s} \alpha_i a_{ij}^{e_i} \left(x + \frac{a_{i0}}{a_{ij}}\right)^{e_i}.$$

Since $\pi_j(g)$ satisfies the hypotheses of Theorem 4.5 $\texttt{Build}(\pi_j(g))$ outputs the values

$$\left\{(\alpha_i a_{ij}^{e_i}, \frac{a_{i0}}{a_{ij}}, e_i) \,|\, 1 \leq i \leq s\right\}.$$

From these values we obtain in the sets

$$P_j = \left\{(\alpha_i a_{i0}^{e_i}, \frac{a_{ij}}{a_{i0}}, e_i) \,|\, 1 \leq i \leq s\right\}.$$

[8] A detailed probabilistic analysis is performed at the end of this proof.

Before calling $\texttt{Build}(\pi_j(g))$, we compute the dense representation of $\pi_j(g)$ using Remarks 6.1 and 6.2.

Thanks to the unique expression of g_j for all j and to (iii) we have that none of the conditions of **Step 3** is satisfied and we obtain g in **Step 4**.

If we see the values of λ_i, λ_{ij} used to define ϕ as variables, the invertibility of ϕ is equivalent to the nonvanishing of a degree n polynomial. Moreover, the a_{ij} are degree one polynomials in these variables. Thus, the conditions $a_{ij} \neq 0$ consist in the nonvanishing of $s(n+1)$ polynomials of degree 1. The conditions $a_{ij}/a_{i0} \neq a_{i'j}/a_{i'0}$ for all i, i', j with $j \neq 0$ can be seen as the nonvanishing of $s(s-1)n/2$ polynomials of degree 2. The conditions $\alpha_i a_{i0}^{e_i} \neq \alpha_{i'} a_{i'0}^{e_{i'}}$ can be seen as the nonvanishing of $s(s-1)/2$ polynomials of degree at most $\max(e_i)$, which, by Corollary 3.16, is upper bounded by $d + (s^2/2)$. Hence, all the conditions to be satisfied can be codified in a nonzero polynomial ψ of degree

$$n + s(n+1) + s(s-1)n + (s(s-1)(2d+s^2)/4) \leq \frac{8s^2n + 2s^2d + s^4}{4}.$$

Moreover, if we set $e := \max(e_i)$, then

- $e \leq d + (s^2/2)$, and
- $s = n_e \leq (3e/4)^{1/3}$,

form where we deduce that $s \leq d^{1/3}$ and the degree of ψ is upper bounded by $d^{2/3}(2n+d)$. Hence, by the Schwartz-Zippel lemma, if we assume the λ_i, λ_{ij} are taken uniformly at random from a finite set S, the probability of satisfying all these constraints is at least

$$1 - \frac{d^{2/3}(2n+d)}{|S|}.$$

and the result follows. □

Acknowledgments

The reconstruction problem for sums of affine powers was suggested to one of us (P.K.) by Erich Kaltofen at a Dagstuhl workshop where P.K. gave a talk on lower bounds for this model.

References

[1] James Alexander and André Hirschowitz. Polynomial interpolation in several variables. *Journal of Algebraic Geometry*, 4(2):201–222, 1995.

[2] M. Ben-Or and P. Tiwari. A deterministic algorithm for sparse multivariate polynomial interpolation. In *Proc. 20th annual ACM Symposium on Theory of Computing*. ACM, 1988.

[3] M. Bocher. The theory of linear dependence. *Annals of Mathematics*, 2(1/4): 81–96, 1900-1901.

[4] A. Borodin and P. Tiwari. On the decidability of sparse univariate polynomial interpolation. *Computational Complexity*, 1(1):67-90, 1991.

[5] M. Boij, E. Carlini, A. V. Geramita. Monomials as sums of powers: the real binary case. *Proc. Amer. Math. Soc.* 139(9):3039–3043, 2011.

[6] A. Bostan, F. Chyzak, M. Giusti, R. Lebreton, G. Lecerf, B. Salvy, and É. Schost. Algorithmes efficaces en calcul formel. Preprint version, 2017, 686 pages.

[7] Maria Chiara Brambilla and Giorgio Ottaviani. On the Alexander –Hirschowitz theorem. *Journal of Pure and Applied Algebra*, 212(5):1229–1251, 2008.

[8] D. A. Cox, Galois theory. Second edition. Pure and Applied Mathematics (Hoboken). John Wiley & Sons, Inc., 2012.

[9] I. García-Marco, and P. Koiran. Lower bounds by Birkhoff interpolation. Submitted, arXiv:1507.02015 [cs.CC].

[10] M. Giesbrecht, E. Kaltofen, W. Lee. Algorithms for computing sparsest shifts of polynomials in power, Chebyshev and Pochhammer bases. *International Symposium on Symbolic and Algebraic Computation* (ISSAC'2002) (Lille). *Journal of Symbolic Computation* 36(3-4):401–424, 2003.

[11] M. Giesbrecht, G. Labahn and W.-S. Lee. Symbolic-numeric sparse interpolation of multivariate polynomials. *Journal of Symbolic Computation* 44(8):943–959, 2009.

[12] M. Giesbrecht, D. S. Roche. Interpolation of shifted-lacunary polynomials. *Computational Complexity* 19(3):333–354, 2010.

[13] D. Grigoriev and M. Karpinski. A zero-test and an interpolation algorithm for the shifted sparse polynomials. In *Proc. Applied Algebra, Algebraic Algorithms and Error-Correcting Codes, 10th International Symposium (AAECC-10)*. LNCS 673, pp. 162-169, Springer, 1993.

[14] J. H. Grace, A. Young. The algebra of invariants. Cambridge University Press, 1903.

[15] A. Iarrobino, V. Kanev, Power sums, Gorenstein algebras, and determinantal loci. Appendix C by Iarrobino and Steven L. Kleiman. Lecture Notes in Mathematics, 1721. Springer-Verlag, Berlin, 1999.

[16] E. Kaltofen and B. Trager. Computing with polynomials given by black boxes for their evaluations: Greatest common divisors, factorization, separation of numerators and denominators. *Journal of Symbolic Computation* 9(3):301-320, 1990.

[17] J. Kleppe. Representing a Homogenous Polynomial as a Sum of Powers of Linear Forms. *Thesis for the degree of Candidatus Scientiarum (University of Oslo)*, 1999. Available at `http://folk.uio.no/johannkl/kleppe-master.pdf`.

[18] N. Kayal. Affine projections of polynomials. In *Proc. 44th annual ACM Symposium on Theory of Computing (STOC 2012)*, pp. 643-662. ACM, 2012.

[19] N. Kayal, P. Koiran, T. Pecatte, and C. Saha. Lower bounds for sums of powers of low degree univariates. In *Proc. 42nd International Colloquium on Automata, Languages and Programming (ICALP 2015), part I*, LNCS 9134, pages 810–821. Springer, 2015. Available from `http://perso.ens-lyon.fr/pascal.koiran`.

[20] Y. N. Lakshman, B. D. Saunders, Sparse shifts for univariate polynomials. *Appl. Algebra Engrg. Comm. Comput.* 7 (1996), no. 5, 351–364.

[21] Joseph M Landsberg and Zach Teitler. On the ranks and border ranks of symmetric tensors. *Foundations of Computational Mathematics*, 10(3):339–366, 2010.

[22] A. K. Lenstra, H. W. Lenstra, L. Lovász. Factoring polynomials with rational coefficients. Math. Ann. 261 (1982), no. 4, 515–534.

[23] G. Pólya, and G. Szegö. Problems and theorems in analysis. Vol. II. Theory of functions, zeros, polynomials, determinants, number theory, geometry. Revised and enlarged translation by C. E. Billigheimer of the fourth German edition. Springer Study Edition. Springer-Verlag, New York-Heidelberg, 1976. xi+391 pp.

[24] A. Schrijver Theory of linear and integer programming. John Wiley & Sons, 1986. xii+471 pp.

[25] M. Voorhoeve, and A.J. Van Der Poorten. Wronskian determinants and the zeros of certain functions. Indagationes Mathematicae, 37 (1975), no. 5, 417–424.

A Appendix: Algorithms for Sparsest Shift and Waring decomposition

In this appendix we apply the techniques from the previous sections to study optimal decompositions of polynomials in the Waring and Sparsest Shift models. As explained in the introduction, these two models have been extensively studied in the literature. We do not claim that the algorithms proposed in this appendix improve on the existing methods. Rather, we present them for the sole purpose of illustrating on these two classical models the techniques developed for the more general model of sums of affine powers.

A.1 Waring decompositions

In Proposition 2.6 we saw that if f has an expression in the AffPow model with s terms, then f satisfies a SDE$(2s-1,0)$. We begin this section by proving that an expression of f with s terms in the Waring model yields a SDE satisfied by f of order s and shift 0.

Proposition A.1. *Let $f \in \mathbb{F}[x]$ be written as*

$$f = \sum_{i=1}^{s} \alpha_i (x-a_i)^d,$$

Then f satisfies a SDE$(s,0)$ that is also satisfied by the $(x-a_i)^d$'s.

Proof. We consider the SDE in the unknown g given by the Wronskian:

$$\mathrm{Wr}(g, (x-a_1)^d, \ldots, (x-a_s)^d)(x) = 0 \tag{11}$$

After factoring out $(x-a_i)^{d-s}$ for all i, we get the reduced SDE:

$$\sum_{i=0}^{s} R_i(x) g^{(i)}(x) = 0,$$

where

$$R_i = \begin{vmatrix} (x-a_1)^s & \ldots & (x-a_s)^s \\ d^{\underline{1}}(x-a_1)^{s-1} & \ldots & d^{\underline{1}}(x-a_s)^{s-1} \\ \vdots & \ddots & \vdots \\ d^{\underline{i-1}}(x-a_1)^{s-i+1} & \ldots & d^{\underline{i-1}}(x-a_s)^{s-i+1} \\ d^{\underline{i+1}}(x-a_1)^{s-i-1} & \ldots & d^{\underline{i+1}}(x-a_s)^{s-i-1} \\ \vdots & \ddots & \vdots \\ d^{\underline{s}} & \ldots & d^{\underline{s}} \end{vmatrix}$$

and $d^{\underline{k}} := \prod_{j=1}^{k}(d-j+1)$. Because of the nice structure induced by all the exponents being equal to d, we have that $R'_{i+1} = R_i$, and hence $\deg(R_i) = \deg(R_s) - (s-i)$. If we factor out the constants on each row in R_s we get that

$$R_s = \prod_{i=1}^{s} d^{\underline{i}} \cdot \begin{vmatrix} (x-a_1)^s & \ldots & (x-a_s)^s \\ (x-a_1)^{s-1} & \ldots & (x-a_s)^{s-1} \\ \vdots & \ddots & \vdots \\ (x-a_1) & \ldots & (x-a_s) \end{vmatrix}$$

We factor $(x - a_i)$ on each row and use the known formula for the determinant of a Vandermonde matrix to obtain:

$$R_s = \prod_{i=1}^{s} d^{\underline{i}} \cdot \prod_{i=1}^{s} (x - a_i) \cdot \prod_{i<j} (a_i - a_j)$$

We have $\deg(R_s) = s$, and hence $\deg(R_i) = i$, which shows that the reduced SDE has in fact a zero shift. □

Moreover, when $\text{Waring}_{\mathbb{F}}(f)$ is small enough we have that the SDE provided in Proposition A.1 is the only SDE$(s, 0)$ satisfied by f.

Corollary A.2. *Let f be a polynomial such that $s = \text{Waring}_{\mathbb{F}}(f) \leq \sqrt{2d/3}$. Then f satisfies a unique SDE$(s, 0)$.*

Proof. Consider a SDE$(s, 0)$ satisfied by f:

$$\sum_{i=0}^{s} P_i(x) f^{(i)}(x) = 0$$

Since $\text{Waring}_{\mathbb{F}}(f) = s$, f can be expressed as $f = \sum_{i=1}^{s} \alpha_i (x - a_i)^d$ and $\{(x - a_i)^d : 1 \leq i \leq d\}$ are linearly independent. By Proposition 4.1 we get that any term $(x - a_i)^d$ satisfies this SDE because

$$d \geq \frac{3}{2} s^2 \geq s + s(s-1) + \binom{s}{2}.$$

Thus, we apply Lemma 2.9 to conclude the equation given by the Proposition A.1 is the unique SDE$(s, 0)$ satisfied by f. □

As a direct consequence of this result, we have the following algorithm.

Algorithm A.3. *Let f be a polynomial of degree d. There is a polynomial time algorithm* `WaringDec`(f) *that receives $f = \sum_{i=0}^{d} f_i x^i \in \mathbb{F}[x]$ as input decides if Waring$(f) \leq \sqrt{2d/3}$. Moreover, whenever $\text{Waring}_{\mathbb{F}}(f) \leq \sqrt{2d/3}$, with the optimal decomposition being*

$$f = \sum_{i=1}^{s} \alpha_i (x - a_i)^d,$$

the algorithm computes the s-tuples of shifts $S(f) = (a_1, \ldots, a_s)$ and coefficients $C(f) = (\alpha_1, \ldots, \alpha_s)$. The algorithm works as follows:

Step 1. *Find the minimum k such that there exists an SDE$(k, 0)$ satisfied by f and compute explicitly one of these SDE.*

Step 2. *If $k > \sqrt{2d/3}$, then Waring$_{\mathbb{F}}(f) > \sqrt{2d/3}$.*

Step 3. *Compute the set $\mathcal{B} = \{(x - b_i)^d \,|\, 1 \leq i \leq t\}$ of solutions of the form $(x - a)^d$ of this SDE.*

Step 4. *If $t < k$, then Waring$_{\mathbb{F}}(f) > \sqrt{2d/3}$.*

Step 5. *Write $f = \sum_{i=1}^{t} \beta_i (x - b_i)^d$, and output $S(f) = (b_1, \ldots, b_t)$ and $C(f) = (\beta_1, \ldots, \beta_t)$.*

Note that if we reach Step 5 of the algorithm, we have $t = k = s \leq \sqrt{2d/3}$.

A.2 Sparsest Shift decompositions

We saw in Section A.1 that a polynomial with a Waring decomposition of size s satisfies a SDE of order s and shift 0. The same is true for the Sparsest Shift model:

Proposition A.4. *Let $f \in \mathbb{F}[x]$ be written as*

$$f = \sum_{i=1}^{s} \alpha_i (x - a)^{e_i},$$

Then f satisfies a SDE$(s, 0)$ that is also satisfied by the $(x - a)^{e_i}$'s.

Proof. We will prove something stronger, namely that f satisfies an SDE$(s, 0)$ of the following form

$$\sum_{i=0}^{s} \gamma_i (x - a)^i g^{(i)}(x) = 0,$$

where $\gamma_0, \ldots, \gamma_s \in \mathbb{F}$. We take the original SDE given by the Wronskian of an unknown polynomial g and $(x - a)^{e_i}$ for all $i \in \{1, \ldots, s\}$:

$$\sum_{i=0}^{s} (-1)^i \, P_i(x) \, g^{(i)}(x) = 0.$$

Because of the stepped sequence of degrees in the determinant defining P_i, there exists an integer Δ_i such that every permutation σ corresponds to a term $c_\sigma (x - a)^{\Delta_i}$. More precisely, we have

$$\Delta_i = \left(\sum_{j=1}^{s} e_j \right) - \binom{s}{2} + i$$

Thus, the determinant is either 0, or some constant times $(x-a)^{\Delta_i}$. Moreover, we have $\Delta_{i+1} = \Delta_i + 1$ and hence we can rewrite the SDE as

$$\sum_{i=0}^{s} c_i (x-a)^{\Delta_0 + i} g^{(i)}(x) = 0$$

with $c_i \in \mathbb{F}$. We factorize this equation by $(x-a)^{\Delta_0}$ to obtain the wanted SDE$(s, 0)$. Notice that the factorization again doesn't change the space of solutions, hence f and $(x-a)^{e_i}$ are still solutions of this SDE. □

Corollary A.5. *Let f be a polynomial of degree d such that $s = Sparsest_{\mathbb{F}}(f) < \sqrt{d}$, and let $a \in \mathbb{F}$ be the corresponding sparsest shift. Let $\sum_{i=1}^{k} P_i(x) f^{(i)}(x) = 0$ be any SDE$(k, 0)$ satisfied by f. If $k \leq s$ we must have $P_k(a) = 0$.*

Proof. Assume that f satisfies an SDE$(k, 0) : \sum_{i=1}^{k} P_i(x) f^{(i)}(x) = 0$ with $k \leq s$. The sparsest shift decomposition of f is

$$f = \sum_{i=1}^{s} \alpha_i (x-a)^{e_i}.$$

Assume that $e_1 > e_2 > \cdots > e_s > e_{s+1} := -1$. We take $t \in \{1, \ldots, s\}$ the maximum value such that $e_t - e_{t+1} - 1 \geq s$. Such a value t exists, otherwise $d \leq e_1 < \sum_{i=1}^{s} (e_i - e_{i+1}) < s^2$, a contradiction. We rewrite f as

$$f = (x-a)^{e_t} g(x) + \sum_{i=t+1}^{s} \alpha_i (x-a)^{e_i}.$$

Now we apply Proposition 5.5 with $p = 0$ to get that $(x-a)^{e_t} g$ satisfies the same SDE because $e_t - e_{t+1} > s \geq k$. By the same argument as in Proposition 5.1, we conclude that $P_k(a) = 0$. □

Algorithm A.6. *Let f be a polynomial of degree d. There is a polynomial time algorithm* `SparsestShift`(f) *that receives $f = \sum_{i=0}^{d} f_i x^i \in \mathbb{F}[x]$ as input decides if $Sparsest_{\mathbb{F}}(f) \leq \sqrt{d}$; moreover, whenever $Sparsest_{\mathbb{F}}(f) \leq \sqrt{d}$, with the optimal decomposition being*

$$f = \sum_{i=1}^{s} \alpha_i (x-a)^{e_i},$$

the algorithm computes the shift $a \in \mathbb{F}$, and the s-tuples of coefficients $C(f) = (\alpha_1, \ldots, \alpha_s)$ and exponents $E(f) = (e_1, \ldots, e_s)$. The algorithm works as follows:

Step 1. *Find the minimum k such that there exists an SDE$(k, 0)$ satisfied by f and compute explicitly one of these SDE. Namely,*

$$\sum_{i=0}^{k} P_i(x) f^{(i)}(x) = 0$$

Step 2. *Factorize the last coefficient of this SDE, i.e., write:*

$$P_k = c \cdot \prod_{i=1}^{k} (x - a_i).$$

Step 3. *For each a_i, decompose f in the shifted basis $((x - a_i)^j)_{0 \leq j \leq d}$.*

Step 4. *If the decomposition with smallest number of terms has $\leq \sqrt{d}$ terms, we output this decomposition; otherwise, Sparsest$_{\mathbb{F}}(f) > \sqrt{d}$.*