Polynomial Equivalence Problems for Sums of Affine Powers

Ignacio García-Marco
Facultad de Ciencias. Universidad de La Laguna
iggarcia@ull.es

Pascal Koiran
LIP, ENS Lyon. Université de Lyon
pascal.koiran@ens-lyon.fr

Timothée Pecatte
LIP, ENS Lyon. Université de Lyon
timothee.pecatte@ens-lyon.fr

ABSTRACT

A sum of affine powers is an expression of the form

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

where ℓ_i is an affine form. We propose polynomial time black-box algorithms that find the decomposition with the smallest value of s for an input polynomial f. Our algorithms work in situations where s is small enough compared to the number of variables or to the exponents e_i. Although quite simple, this model is a generalization of Waring decomposition. This paper extends previous work on Waring decomposition as well as our work on univariate sums of affine powers (ISSAC'17).

ACM Reference Format:
Ignacio García-Marco, Pascal Koiran, and Timothée Pecatte. 2018. Polynomial Equivalence Problems for Sums of Affine Powers. In *ISSAC '18: 2018 ACM International Symposium on Symbolic and Algebraic Computation, July 16–19, 2018, New York, NY, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3208976.3208993

1 INTRODUCTION

Let $\mathbb{F}[X] = \mathbb{F}[x_1, \ldots, x_n]$ be a ring of polynomials in n variables over a characteristic 0 field. This paper studies the multivariate version of the Affine Power model, i.e., we study expressions of a polynomial $f \in \mathbb{F}[X]$ as

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

where $e_i \in \mathbb{N}$, $\alpha_i \in \mathbb{F}$ and ℓ_i is a (non constant) affine form for all i. We denote by $\mathrm{AffPow}_{\mathbb{F}}(f)$ (or $\mathrm{AffPow}(f)$ when $\mathbb{F}$ is clear from the context) the minimum value s such that there exists a representation of the previous form with s terms.

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 $s = \mathrm{AffPow}_{\mathbb{F}}(f)$ and a set of triplets of coefficients, affine forms and exponents $\{(\alpha_i, \ell_i, e_i) \mid 1 \leq i \leq s\} \subseteq \mathbb{F} \times \mathbb{F}[X] \times \mathbb{N}$ such that $f = \sum_{i=1}^{s} \alpha_i \ell_i^{e_i}$.

ISSAC '18, July 16–19, 2018, New York, NY, USA

ACM ISBN 978-1-4503-5550-6/18/07…$15.00
https://doi.org/10.1145/3208976.3208993

The univariate version of this model and several related problems have been extensively studied in [8–10, 14]. These works provide lower bounds, structural results and reconstruction algorithms under certain hypotheses for this model. This paper concerns the reconstruction in the multivariate version of this problem.

Model (1) extends the Waring model, where all the exponents are equal to the degree of the polynomial, i.e., $e_i = \deg(f)$ for all i. For a homogeneous polynomial f of degree d, consider expressions of f of the form:

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

with $\alpha_i \in \mathbb{F}$ and ℓ_i are linear forms. We denote by $\mathrm{Waring}(f)$ 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. We refer to [11] for the historical background. The algorithmic study of the univariate case is often attributed to Sylvester (see [11, section 1.3]). Most of the subsequent work was devoted to the multivariate version[1] with much of the 20th century work focused on the determination of the Waring rank of generic polynomials [1, 4, 11]. A few recent papers [7, 17] have begun to investigate the Waring rank of specific polynomials such as monomials, sums of coprime monomials, the permanent and the determinant.

Waring decomposition has also been studied from an algorithmic point of view, see e.g. [3, 15, 16, 20]. Since Model (1) is more general than the Waring model, the algorithms we provide here can be adapted to find Waring decompositions.

1.1 Our results

In this work we devise algorithms for finding optimal representations of a polynomial $f \in \mathbb{F}[X]$ in Model (1), provided the value of $\mathrm{AffPow}(f)$ is small compared to the number of variables or to the degree of f.

Let us denote by $\mathrm{EssVar}(f)$ the number of *essential variables* of f. This is roughly speaking the number of variables on which f "truly depends" up to a linear change of variables [5, 15]. A first easy remark is that the value $\mathrm{AffPow}(f)$ is at least equal to $\mathrm{EssVar}(f)$. In Section 3 we investigate when this is an equality and provide an algorithm that decides whether $\mathrm{AffPow}(f) = \mathrm{EssVar}(f)$ and, if so, provides an optimal expression in the model.

In Section 4, we generalise the previous results to characterize by means of an algorithm when a polynomial $f \in \mathbb{F}[X]$ can be written as a sum of univariates after an affine change of coordinates. It is plausible that when this is a case, an optimal expression of f

[1] In the literature, Waring rank is usually defined for homogeneous polynomials.

can be built by putting together the optimal expressions of all the univariate polynomials involved. We believe this is true and we give a proof for $n = 2$. The general case ($n \geq 3$) is left as an open problem.

In Section 5, we focus on the reconstruction problem when $\text{AffPow}(f) \leq \binom{n+1}{2}$. In the main result of this section, we provide a randomized algorithm that works when in the optimal decomposition all the e_i's are ≥ 5 and the coefficients of the ℓ_i's are taken uniformly at random from a finite set. In particular, this provides a new algorithm for computing Waring decompositions of "generic polynomials with $\text{Waring}(f) \leq \binom{n+1}{2}$". For comparison, note that the algorithm from [15] can only find Waring decompositions up to size n, and that the Waring decomposition algorithm from [16] is only interesting when d is relatively large compared to s (see Theorem 5 and Remark 6 in that paper). Our main tool in this section is a "4th order Hessian" inspired from the ordinary Hessian determinant used in [15].

Finally, in Section 6 we propose an algorithm that performs random univariate projections, calls our univariate algorithm for sums of affine powers [9] and reconstructs f from this univariate information.

1.2 Model of computation

Throughout this paper, we will 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 $(x_1, \ldots, x_n) \in \mathbb{F}^n$.

Our algorithms handle polynomials with coefficients in an arbitrary field $\mathbb{F}$ 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 $\mathbb{F}$. We will additionally assume that we are able to solve polynomial equations in one variable. When we write that an algorithm runs in polynomial time, we mean that the number of such steps and of calls to the black box is polynomial in the input size.

Whenever $\mathbb{F}$ is an algebraically closed field, then we may assume without loss of generality that all the α_i's equal 1 in a expression in Model (1). For the sake of conciseness, we assume this is the case. However, one can restate all the results in this work for a non algebraically closed field by just adding the α_i's.

1.3 Future work

The problem studied in the present paper is far from completely solved: our algorithms rely on assumption on $\text{AffPow}(f)$ being small and, sometimes, on a random choice of the affine forms involved in the optimal expression. It would be very interesting to weaken these assumptions, or even to remove them entirely, even though the recent NP-hardness result of Waring decomposition [21] and the similarity of both problems seems to indicate that it might be hard to do so.

We prove in Proposition 4.9 that whenever a bivariate polynomial $f(x_1, x_2)$ is a sum of two univariate ones $g_1(x_1), g_2(x_2)$, one can construct an optimal expression of f in Model (1) by gathering the (univariate) optimal expressions of f_1 and f_2 and putting together the terms of degree 1. We do not know if this phenomenon is also true for polynomials in more than two variables that can be written as a sum of univariates. Even more generally, we wonder if whenever $f(X) \in \mathbb{F}[X]$ can be expressed as a sum of two polynomials g_1, g_2 in disjoints set of variables, then an optimal expression for f in Model (1) can be built up from the optimal expressions of g_1 and g_2 by just putting together the terms of degree 1. This could be seen as a analog of the Strassen's conjecture for the symmetric tensor rank, which can be stated as follows: the rank is additive on the sum of forms in different set of variable (see [22]). Our result should be compared with [6, Theorem 5.6], where the authors prove the conjecture for homogeneous polynomials in four variables that can be written as a sum of two bivariate ones.

2 PRELIMINARIES

2.1 Essential variables

We say that a polynomial $f \in \mathbb{F}[X]$ *depends on a variable* if it appears in at least one of the monomials of $f(X)$. The *number of essential variables* of a polynomial f, denoted by $\text{EssVar}(f)$, is the least integer $t \in [\![0, n]\!]$ such that there exists an invertible linear transformation $A \in \text{GL}_n(\mathbb{F})$ such that $f(A \cdot X)$ depends on t variables. The number of essential variables of a polynomial is given by the following result due to Carlini [5, Proposition 1].

Proposition 2.1. *For a polynomial $f \in \mathbb{F}[X]$, we have*

$$\text{EssVar}(f) = \dim_{\mathbb{F}} \left\langle \frac{\partial f}{\partial x_i} \mid 1 \leq i \leq n \right\rangle.$$

A first easy observation is that if $f(X) = \sum_{i=1}^{s} \ell_i^{e_i}$ with ℓ_i affine forms and $e_i \in \mathbb{N}$; then $\left\langle \frac{\partial f}{\partial x_i} \mid 1 \leq i \leq n \right\rangle \subseteq \left\langle \ell_i^{e_i - 1} \mid 1 \leq i \leq s \right\rangle$ and, hence, $\text{EssVar}(f) \leq s$. In particular, $\text{EssVar}(f) \leq \text{AffPow}(f)$.

A polynomial $f \in \mathbb{F}[X]$ is said *regular* if it has n essential variables. From now on we always assume that the input polynomial $f \in \mathbb{F}[X]$ is regular. This can be achieved through a preliminary step consisting of eliminating the redundant variables using a randomized polynomial time algorithm (see, e.g., [16, Lemma 17] and [15, Theorem 4.1]).

2.2 Algorithmic preliminaries

In the rest of this paper, we will design algorithms that work in the "black box" setting: they have access to the input polynomial only through an oracle so that for any point $a \in \mathbb{F}^n$, we can obtain $f(a)$ in a single step by querying this oracle. This very general model is standard for the study of many problems about multivariate polynomials such as, e.g., factorization [12], sparse interpolation [2, 18], sparsest shift [19] or Waring decomposition [13]. In this section, we describe some useful blackbox subroutines that our algorithms will use.

2.2.1 Polynomial Identity Testing. Given a blackbox access to a polynomial $f \in \mathbb{F}[X]$ of degree d, the Schwartz-Zippel lemma ensures that evaluating f at random points yields a randomized polynomial time algorithm that tests whether f is equal to the zero polynomial.

2.2.2 Obtaining the derivatives.

Proposition 2.2. *[16, Proposition 18] Let $f(X) \in \mathbb{F}[X]$ be an n-variate polynomial of degree d. Given blackbox access to f, in time poly(dn), we obtain blackbox access to any derivative $\frac{\partial f}{\partial x}$ of f.*

2.2.3 Obtaining the homogeneous components. For a polynomial $f(X) \in \mathbb{F}[X]$ we will denote by $[f]_k$ its homogeneous component of degree k. We will also sometimes use the notation $[f]_{\geq k}$ defined as $[f]_{\geq k} := \sum_{i \geq k} [f]_i$.

Proposition 2.3. *[16, Proposition 19] Let $f \in \mathbb{F}[X]$ be a polynomial of degree d. Given blackbox access to f and a point $a \in \mathbb{F}^n$, we can compute $[f]_i(a)$ for each $i \in [0..d]$ in polynomial time.*

2.2.4 Factorization. To factorize a polynomial, we will use the randomized polynomial time algorithm described in [12] which outputs blackbox access to its factors. In order to apply this algorithm, we will assume that we have an effective polynomial factorization algorithm for $\mathbb{F}[x]$. In the following, we will often need to reconstruct the coefficients of a degree 1 factor h from a blackbox access to it. This can be easily done using an additional randomized step in poly(n) time: we evaluate h in $n+2$ random points and then interpolate the coefficients.

3 FROM RECONSTRUCTION TO POLYNOMIAL EQUIVALENCE

Let us first consider $f \in \mathbb{F}[X]$ a regular polynomial such that $\text{AffPow}(f) = n$, i.e. there exists a decomposition $f(X) = \sum_{i=1}^{n} \ell_i^{e_i}$. We construct the matrix A from the linear parts of the ℓ_i's and the vector b from the constant terms. Since f is regular, we have that $A \in \text{GL}_n(\mathbb{F})$. This implies that $f(A^{-1}X - A^{-1}b) = \sum_{i=1}^{n} x_i^{e_i}$ and motivates the following definitions.

Definition 3.1. [16] We will say that two n-variate polynomials f and g are equivalent, denoted $f \sim g$, if there exists an invertible linear transformation $A \in \text{GL}_n(\mathbb{F})$ such that $f(X) = g(A \cdot X)$. Moreover, we will say that f and g are affine equivalent, denoted $f \equiv g$ if there exists a vector $c \in \mathbb{F}^n$ such that $f(X + c) \sim g$, or similarly if $f = g(A \cdot X + b)$ with $A \in \text{GL}_n(\mathbb{F}), b \in \mathbb{F}^n$.

With these notations, for a regular polynomial f, we have that $\text{AffPow}(f) = n$ if and only if $f \equiv g$ where $g = \sum_{i=1}^{n} x_i^{e_i}$ for some $(e_i) \in \mathbb{N}^n$. This restates the problem of checking whether $\text{AffPow}(f) = n$ into a problem of testing affine equivalence. The affine equivalence problem was already investigated in [16]. One major difference of our situation with respect to [16] is that instead of testing affine equivalence to one target polynomial g, we test affine equivalence to a family of polynomials. Another difference is that its author used [16, Theorem 28] as a preliminary step to reduce the affine equivalence problem to an equivalence problem, which cannot be used here since the polynomials we consider are not homogeneous in general. Yet, the techniques used to solve some special cases of the equivalence problem in [15] have been a source of inspiration to design the algorithms of this paper.

3.1 Algorithm overview

Let us fix some notations: unless stated otherwise, f will always denote the input polynomial and g one target polynomial. Whenever $f \equiv g$, we will usually denote by A and b the matrices such that $f(X) = g(A \cdot X + b)$, with $A \in \text{GL}_n(\mathbb{F})$. Moreover, we will define the associated affine and linear forms: $\ell_i = \sum_{j=1}^{n} A_{i,j} x_j + b_i$ and $[\ell_i] = \ell_i - b_i$. The main tool of the algorithms is the Hessian matrix, whose entries are the second order derivatives of the polynomial.

Definition 3.2. For a polynomial $f \in \mathbb{F}[X]$, the Hessian matrix $H_f \in \mathcal{M}_n(\mathbb{F}[X])$ is defined as

$$(H_f)_{i,j} = \frac{\partial^2 f}{\partial x_i \partial x_j}$$

In the following, the most useful property of the Hessian matrix is how affine transformations change the matrix. This Lemma is an affine analogue of [15, Lemma 5.1] and can be proved similarly.

Lemma 3.3. *Let $g \in \mathbb{F}[X]$, $A \in \mathcal{M}_n(\mathbb{F})$ be a linear transformation, and $b \in \mathbb{F}^n$. Consider $f(X) = g(A \cdot X + b)$, then,*

$$H_f(X) = A^T \cdot H_g(A \cdot X + b) \cdot A.$$

In particular we have $\det(H_f(X)) = \det(A)^2 \det(H_g(A \cdot X + b))$.

In particular when $f \equiv g$, the matrix A is invertible and hence the determinant of the Hessian matrix of f can be understood by studying an affine transformation of the determinant of the Hessian matrix of g. For instance, when $g = \sum_{i=1}^{n} x_i^{e_i}$, observe that the matrix H_g is diagonal, and we therefore have

$$\det(H_g(X)) = \prod_{i=1}^{n} e_i(e_i - 1) x_i^{e_i - 2}.$$

In particular, Lemma 3.3 directly implies the following result.

Lemma 3.4. *Let f be a regular polynomial such that $f(X) = \sum_{i=1}^{n} \ell_i(X)^{e_i}$ where $\ell_1, \ldots, \ell_n$ are affine forms and $e_i \geq 2$. Then there exists a nonzero constant $c \in \mathbb{F}$ such that*

$$\det(H_f(X)) = c \cdot \prod_{i=1}^{n} \ell_i(X)^{e_i - 2}.$$

This results yields a blueprint for an algorithm to find a decomposition of f when $\text{AffPow}(f) = n$: factorize $\det(H_f(X))$ to obtain candidates for the affine forms and associated exponents, then try to express f as a linear combination of these affine powers. However, if $\text{AffPow}(f) = n$ and one e_i is ≤ 1 then $\det(H_f(X)) = 0$, and if some of the e_i's are equal to 2 then ℓ_i is not a factor of $\det(H_f(X))$. This makes this idea fail on such scenarios. Therefore, in order to have an algorithm that decides whether $\text{AffPow}(f) = n$, one also needs to handle the case when some of the (e_i)'s are smaller than 3. In the next section, we start tackling this problem by studying the case where f is a quadratic polynomial.

3.2 Quadratic polynomials

The goal of this subsection is to describe how to obtain an optimal expression in the Affine Powers model for every polynomial of degree 2. In particular, we are going to generalize the following classical result concerning homogeneous polynomials of degree 2.

Proposition 3.5. *Let $f, g \in \mathbb{F}[X]$ be homogeneous quadratic polynomials. Then $f \sim g \Leftrightarrow \text{EssVar}(f) = \text{EssVar}(g)$.*

As a consequence, for a quadratic homogeneous regular polynomial f, we have $\text{AffPow}(f) = \text{EssVar}(f) = n$ since $f \sim \sum_{i=1}^{n} x_i^2$. Now we can proceed with the classification of degree 2 polynomials.

Theorem 3.6. *Let $f \in \mathbb{F}[X]$ be a polynomial of degree at most 2. Then, there exists a polynomial time algorithm that obtains an expression of f as either*

(a) $\sum_{i=1}^{s} \ell_i^2$, *(b)* $\sum_{i=1}^{s} \ell_i^2 + c$ *with* $c \in \mathbb{F}^*$, *or (c)* $\sum_{i=1}^{s-1} \ell_i^2 + \ell_s$, *with* $r \in [\![0, n]\!]$, *and ℓ_i linear forms.*

Proof. We propose a greedy algorithm showing how to write f in one of the three forms. We proceed by induction on the number of variables of f. If f has 0 or 1 variables or f has degree one, it is trivial to write f in one of the desired forms. Assume now that f has $n \geq 2$ variables and that f has degree 2. If there exists a variable x such that the monomial x^2 appears in f, then after multiplying f by a constant if necessary, we write $f = x^2 + xt + g$, where t is a linear form in $n-1$ variables and g is a polynomial of degree ≤ 2 in $n-1$ variables. Thus setting $\ell_1 = x + (t/2)$, we have that $f = \ell_1^2 + g - (t^2/4)$ and proceeding by induction on $g - (t^2/4)$ we are done. If for every variable there is no monomial of the form x^2 in f, then we take two variables x, y such that the monomial xy appears in f. After multiplying f by a constant if necessary, we have that $f = xy + xt_1 + yt_2 + g$, where t_1, t_2 are linear forms in $n-2$ variables and g is a polynomial of degree ≤ 2 in $n-2$ variables. So we set $\ell_1 = (x + y + t_1 + t_2)/2$ and $\ell_2 = (x - y - t_1 + t_2)/2$ and we have that $f = \ell_1^2 - \ell_2^2 + g - t_1 t_2$ and we proceed by induction on $g - t_1 t_2$. We also observe that by construction the linear parts of the affine forms $\ell_1, \ldots, \ell_s$ we obtain are linearly independent. Thus, $s = \mathrm{EssVar}(f)$.

□

As a consequence we have the following result, which shows that the greedy algorithm of Theorem 3.6 provides an effective method to compute the exact value of $\mathrm{AffPow}(f)$ for any degree 2 polynomial f. In particular, it implies that a quadratic polynomial always has an optimal decomposition with exponents at most 2.

Corollary 3.7. *Let $f \in \mathbb{F}[X]$ be a regular polynomial of degree at most 2. Then, $\mathrm{AffPow}(f) = n+1$ if we have $f \equiv \sum_{i=1}^{n} x_i^2 + c$ with $c \in \mathbb{F}^*$; and $\mathrm{AffPow}(f) = n$ otherwise.*

Proof. By Theorem 3.6 we know that f is equivalent to

(a) $\sum_{i=1}^{n} x_i^2$, (b) $\sum_{i=1}^{n} x_i^2 + c$ with $c \in \mathbb{F}^*$, or (c) $\sum_{i=1}^{n-1} x_i^2 + x_n$.

Let us prove now that these scenarios are disjoint. First, in (a) or (b) we have that $\det(H_f(X)) \neq 0$, whereas $\det(H_f(X)) = 0$ in (c). For any polynomial g, we denote by $g^h \in \mathbb{F}[X, z]$ its homogenization with respect to a new variable z. In (a) we have that $f^h = \sum_{i=1}^{n} (\ell_i^h)^2$, whereas $f^h = \sum_{i=1}^{n-1} \ell_i^2 + cz^2$ in (b). By Proposition 3.5, in (a) we have that $\mathrm{EssVar}(f^h) = n$ whereas in (b) we have that $\mathrm{EssVar}(f^h) = n+1$; showing the disjointness.

If $f \equiv \sum_{i=1}^{n} x_i^2$ or $f \equiv \sum_{i=1}^{n-1} x_i^2 + x_n$, then $\mathrm{AffPow}(f) \leq n$ and equality holds because $\mathrm{AffPow}(f) \geq \mathrm{EssVar}(f) = n$. It only remains to consider when $f \equiv \sum_{i=1}^{n} x_i^2 + c$ with $c \in \mathbb{F}^*$. In this case we clearly have that $\mathrm{AffPow}(f) \leq n+1$, hence to prove equality we just need to prove that $\mathrm{AffPow}(f) \neq n$. Assume for contradiction that $f = \sum_{i=1}^{n} \ell_i^{e_i}$ for some affine forms ℓ_i and some $e_i \in \mathbb{N}$. Since we have neither $f \equiv \sum_{i=1}^{n} x_i^2$ nor $f \equiv \sum_{i=1}^{n-1} x_i^2 + x_n$, there exists some exponent $e_i \geq 3$. By Lemma 3.4, we have that $\det(H_f)$ is a non-constant polynomial or zero, a contradiction.

3.3 Linear terms in an optimal expression

We now investigate the case where $f \equiv g$ with $g = \sum_{i=1}^{n} x_i^{e_i}$ and $\min(e_i) = 1$. Notice first that $e_i = 1$ can only hold for one $i \in [\![1, n]\!]$ since otherwise $\mathrm{EssVar}(f) = \mathrm{EssVar}(g) < n$. Up to renaming the variables, we can therefore write g as $g = \sum_{i=1}^{n-1} x_i^{e_i} + x_n$. We define $h = g - x_n$ and we decompose A along its last line l so that the equality of Lemma 3.3 can be rewritten as

$$H_f(X) = (B^T \; l^T) \cdot \begin{pmatrix} H_h(A \cdot X + b) & 0 \\ 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} B \\ l \end{pmatrix}$$

If we denote by $[H_f]_{k,k}$ the submatrix of H_f obtained by deleting the k^{th} row and the k^{th} column of H_f, and by $[B]_k$ the square submatrix of B obtained by deleting the k^{th} column, then we have:

$$[H_f(X)]_{k,k} = ([B]_k)^T \cdot H_h(A \cdot X + b) \cdot [B]_k$$

Since $A \in \mathrm{GL}_n(\mathbb{F})$, we have $\operatorname{rank} B = n-1$ and therefore there exists $k \in [\![1, n]\!]$ such that $[B]_k \in \mathrm{GL}_{n-1}(\mathbb{F})$. Finally, since $\det(H_h(X)) = \prod_{i=1}^{n-1} e_i(e_i - 1) x_i^{e_i - 2}$, we get the following result.

Lemma 3.8. *Let f be a regular polynomial such that $f(X) = \sum_{i=1}^{n-1} \ell_i(X)^{e_i} + \ell_n(X)$ where $\ell_1, \ldots, \ell_n$ are affine forms. Then there exists an integer $k \in [\![1, n]\!]$ and a nonzero constant $c \in \mathbb{F}$ such that*

$$\det([H_f(X)]_{k,k}) = c \cdot \prod_{i=1}^{n-1} \ell_i(X)^{e_i - 2}$$

3.4 Wrapping up : the algorithm

The goal of this subsection is to design a polynomial-time randomized algorithm that receives as input a blackbox access to a regular polynomial $f \in \mathbb{F}[X]$, and decides whether $\mathrm{AffPow}(f) = n$ and, in such a case, provides an optimal expression of f in Model (1).

Theorem 3.9. *There exists a polynomial-time randomized algorithm* `Build1` *that receives as input a blackbox access to a regular polynomial $f \in \mathbb{F}[X]$ and finds an optimal decomposition of f in the Affine Powers model if $\mathrm{AffPow}(f) = n$, or rejects otherwise.*

Proof. We obtain blackbox access to $D(X) = \det(H_f(X))$ and distinguish two cases depending on whether it vanishes or not.

Case $D \neq 0$: if D does not split into degree 1 factors, we reject. Otherwise we write $D = c \cdot \prod_{i=1}^{t} \ell_i^{m_i}$ with $c \in \mathbb{F}^*$ and $\ell_1, \ldots, \ell_t$ affine forms. If $t > n$, we reject. Consider the $t \times s$ matrix A whose rows are the $[\ell_i]$'s, and the matrix b whose entries are the constant terms. If the system $A \cdot X = -b$ has no solution, we reject. Otherwise, let X_0 be one solution, and consider $h(X) = g(X + X_0)$ so that $(\ell_i(X - X_0))^{m_i+2} = [\ell_i]^{m_i+2}$ is a homogeneous polynomial of degree $m_i + 2 \geq 3$. By Lemma 3.4, these are the only terms of degree ≥ 3 in an expression of f as a combination of n affine powers. Therefore, if $[h]_{\geq 3} \notin \left\langle [\ell_i]^{m_i+2} \right\rangle$ (see Section 2.2.3), then we reject. Otherwise, let (α_i) be such that $h = \sum_{i=1}^{t} \alpha_i [\ell_i]^{m_i+2} + [h]_{\leq 2}$. We express $[h]_{\leq 2} = \sum_{i=1}^{r} \beta_i t_i^{e_i}$ as in Theorem 3.6. If $r + t \neq n$, then reject. Otherwise output the optimal expression of $f(X) = g(B^{-1} \cdot X) = h(B^{-1} \cdot X - X_0)$.

Case $D = 0$: for all k such that $\det([H_f(X)]_{k,k}) \neq 0$, we repeat the previous procedure. If no such k exists, or if we reject for all such k, then we reject; otherwise we output the optimal expression.

Correctness of the algorithm is justified by Lemma 3.4 and Lemma 3.8. □

In Sections 4 and 5, we generalize this algorithm in two natural ways: by allowing the affine forms to be repeated, or by allowing more than $\mathrm{EssVar}(f)$ different affine forms.

4 REPEATED AFFINE FORMS

In this section, we investigate the case where there exists a decomposition of a regular polynomial f with n different affine forms that

can be used possibly several times in the decomposition. Since f is regular, the n affine forms are necessarily linearly independent. In other words, we want to test if $f \equiv g$ with $g = \sum_{i=1}^{n} \sum_{j=1}^{t_i} \alpha_{i,j} x_i^{e_{i,j}}$. In such a scenario, we can write f as a sum of univariate polynomials: $f = \sum_{i=1}^{n} g_i(\ell_i(X))$ with $g_i(x) = \sum_{j=1}^{t_i} \alpha_{i,j} x^{e_{i,j}}$ and ℓ_i an affine form. Conversely, if f can be written in this way, we can obtain a decomposition with n linearly independent affine forms by taking a decomposition for each univariate polynomial g_i. This motivates the study of the following problem of *univariate decomposition*:

Problem 4.1. *Given $f \in \mathbb{F}[X]$, is $f \equiv g$ with $g = \sum_{i=1}^{n} g_i(x_i)$?*

Yet, this problem does not completely capture the problem of finding an optimal decomposition in the AffPow model: indeed, even if a polynomial has a univariate decomposition $f(X) = \sum_{i=1}^{n} g_i(\ell_i(X))$, we have no guarantee that taking an optimal AffPow decomposition for each g_i will yield an optimal decomposition of f in Model (1). In the following, we first study Problem 4.1 on its own, and then solve the bivariate case by proving that indeed an optimal univariate decomposition is optimal in Model (1).

4.1 Decomposing a polynomial as sum of univariates

The goal of this section is to design an algorithm in Theorem 4.4 that receives as input a regular polynomial f and computes a univariate decomposition if there is one. Notice first that Problem 4.1 is equivalent to testing if there exists univariate polynomials $(g_i(x))$ such that $f \sim g_1(x_1) + \cdots + g_n(x_n)$. A more general version of this problem has been already studied in Appendix C of [15] where the following result is proved:

Theorem 4.2. *[15, Theorem C.2] Given an n-variate polynomial $f(X) \in \mathbb{F}[X]$, there exists an algorithm that finds a decomposition of f as $f(A \cdot X) = p(x_1, \ldots, x_t) + q(x_{t+1}, \ldots, x_n)$, with A invertible, if it exists, in randomized polynomial time provided $\det(H_f)$ is a regular polynomial, i.e. it has n essential variables.*

In the following, we will see how to find a univariate decomposition even if the determinant of the Hessian is not regular. The following result both provides the main ideas and justifies the correctness of the algorithm we propose.

Proposition 4.3. *Let $f \in \mathbb{F}[X]$, and let g_i's be univariate polynomials sorted by decreasing degree. Let $d_i := \deg(g_i)$ and $k := \max\{i : d_i \geq 3\}$. Let $\ell_1, \ldots, \ell_n$ be linear forms such that $f = \sum_{i=1}^{n} g_i(\ell_i)$. Then,*

$$\det(H_f(X)) = c \cdot \prod_{i=1}^{k} \prod_{j=1}^{d_i - 2} (\ell_i - \alpha_{i,j}),$$

where $c \in \mathbb{F}$, and $\alpha_{i,j}$ are the roots of $g_i''(x)$ for $1 \leq i \leq k$.

Moreover, if $\ell_1, \ldots, \ell_n$ are linearly independent, for any solution $X_0 \in \mathbb{F}^n$ to the system $B \cdot X_0 = (\alpha_{1,1}, \ldots, \alpha_{k,1})^T$, where B is the $k \times n$ matrix whose rows are the coefficients of the $\ell_1, \ldots, \ell_k$, we have that

(a) $[f(X + X_0)]_{\geq 3} = \sum_{i=1}^{k} h_i(\ell_i)$ for some unique $h_i \in \mathbb{F}[x]$, and
(b) $\mathit{EssVar}([f(X + X_0)]_2) = |\{i \mid \deg(g_i) = 2\}|$.

Proof. By Lemma 3.3, $\det(H_f(X)) = (\det(A))^2 \prod_{i=1}^{n} g_i''(\ell_i)$, where A is the matrix whose i-th row corresponds to the coefficients of ℓ_i. It suffices to write $g_i''(x) = c_i \prod_{i=1}^{d_i - 2}(x - \alpha_{i,j})$ for all $i \in [\![1, k]\!]$ to get the first part of the result.

We assume now that $\ell_1, \ldots, \ell_n$ are linearly independent. To prove (a), we observe that

$$[f(X + X_0)]_{\geq 3} = \sum_{i=1}^{k} [g_i(\ell_i(X + X_0))]_{\geq 3} = \sum_{i=1}^{k} [g_i(\ell_i + \alpha_{i,1})]_{\geq 3};$$

so it suffices to take $h_i(x) := [g_i(x + \alpha_{i,1})]_{\geq 3}$ for $i = 1, \ldots, k$. Uniqueness of h_i comes directly from the fact that $\ell_1, \ldots, \ell_k$ are linearly independent.

To prove (b) we observe first that $[g_i(x + \alpha_{i,1})]_2 = 0$ because $g_i''(\alpha_{i,1}) = 0$ for $i = 1, \ldots k$. Since ℓ_i is a linear form this implies that $[g_i(\ell_i + \alpha_{i,1})]_2 = 0$ for all $i \in [\![1, k]\!]$, and then

$$[f(X + X_0)]_2 = \sum_{d_i = 2} [g_i(\ell_i(X + X_0))]_2 = \sum_{d_i = 2} \gamma_i \ell_i^2,$$

for some $\gamma_i \neq 0$ and, thus, (b) follows from Proposition 3.5. □

Theorem 4.4. *There exists a polynomial-time randomized algorithm that receives as input a blackbox access to a regular polynomial $f \in \mathbb{F}[X]$ and finds a univariate decomposition of f if such a decomposition exists, or rejects otherwise.*

Proof. The algorithm works as follows. We first compute $D(X) = \det(H_f(X))$ and separate two cases.

Case $D \neq 0$: if $D(X)$ does not split into polynomials of degree 1, we reject. Otherwise we take $\ell_1, \ldots, \ell_k$ all the non-proportional linear parts of the factors and build the associated $k \times n$ matrix B. If $\mathrm{rank}(B) \neq k$, we reject. Otherwise, we gather the factors to write

$$D(X) = c' \cdot \prod_{i=1}^{k} p_i(\ell_i(X)) \quad \text{with} \quad p_i(x) = \prod_{j=1}^{d_i} (x - \alpha_{ij})$$

where c' are nonzero constants. Now we take X_0 a solution of $B \cdot X_0 = (\alpha_{1,1}, \ldots, \alpha_{k,1})^T$ and consider $g(X) = f(X + X_0)$. Let $h_1, \ldots, h_k \in \mathbb{F}[x]$ be the only polynomials so that $[g]_{\geq 3} = \sum_{i=1}^{k} h_i(\ell_i)$ (or reject if they do not exist). Then we use the greedy algorithm of Section 3.2 to write $[g]_2$ as $\sum_{i=k+1}^{m} \gamma_i \ell_i^2$ for some new linear forms $\ell_{k+1}, \ldots, \ell_m$. If $m \neq n$ or $\ell_1, \ldots, \ell_n$ are not linearly independent, we reject. Otherwise, we express $[g]_{\leq 1} = \sum_{i=1}^{n} \delta_i \ell_i + b$ for some $\delta_1, \ldots, \delta_n, b \in \mathbb{F}$. Putting all together we have that g can be written as $\sum_{i=1}^{k} (h_i(\ell_i) + \delta_i \ell_i) + \sum_{i=k+1}^{n} \left(\gamma_i \ell_i^2 + \delta_i \ell_i\right) + b$, and we finally get an univariate decomposition of f as $f(X) = \sum_{i=1}^{n} q_i(t_i)$ with

$$q_i(x) := \begin{cases} h_1(x) + \delta_1 x + b & \text{for } i = 1 \\ h_i(x) + \delta_i x & \text{for } i = 2, \ldots, k \\ \gamma_i x^2 + \delta_i x & \text{for } i = k+1, \ldots, n \end{cases}$$

and $t_i(X) := \ell_i(X - X_0)$ an affine form for all $i \in [\![1, n]\!]$.

Case $D = 0$: this case happens whenever f is equivalent to a sum of univariate polynomials where one of the g_i's is of degree 1. To handle this situation we proceed similarly to Section 3.3. Again we can have at most one g_i of degree 1 since otherwise the number of essential variables of f would not be n. In this case we use the following more general version of Lemma 3.8 which can be proved using similar techniques.

Lemma 4.5. *Let $f(X)$ be a regular polynomial such that $f(X) = \sum_{i=1}^{n-1} g_i(\ell_i(X)) + \ell_n(X)$ where $\ell_1, \ldots, \ell_n$ are affine forms, and g_i is a univariate polynomial of degree ≥ 2 for all i. Then there exists an*

integer $k \in [\![1, n]\!]$ *and* $c \neq 0$ *such that*

$$\det([H_f(X)]_{k,k}) = c \cdot \prod_{i=1}^{n-1} g_i''(\ell_i(X))$$

Hence, for all k such that $D_k := \det([H_f(X)]_{k,k}) \neq 0$, we proceed as before with D_k and try to express $[f]_{\geq 2}$ as $[\sum_{i=1}^{n-1} q_i(t_i)]_{\geq 2}$. If we succeed, we set $t_n := f - \sum_{i=1}^{n-1} q_i(t_i)$, $q_n := x$ and output the optimal expression. If there is no k with $D_k \neq 0$, or if we reject for all such k, then we reject. □

4.2 The bivariate case

Let $f \in \mathbb{F}[x_1, x_2]$ be a bivariate polynomial that admits a univariate decomposition $f = f_1(\ell_1) + f_2(\ell_2)$. In this case, we are going to describe how the optimal expression of f can be obtained from the (univariate) optimal expressions of f_1 and f_2, by putting together, if possible, the terms of degree ≤ 1 in one bivariate polynomial. More precisely, write $f_i = \sum_{j=1}^{s_i} \alpha_{i,j}(x_i + a_{i,j})^{e_{i,j}}$, with $s_i := \mathrm{AffPow}(f_i)$ and $e_{i,1} \leq \cdots \leq e_{i,s_i}$. We separate two cases: if there exist optimal expressions of f_1 and f_2 with $e_{1,1} \leq 1$ and $e_{2,1} \leq 1$; we define $\mathrm{UnivAffPow}(f) := s_1 + s_2 - 1$. Otherwise, we define $\mathrm{UnivAffPow}(f) := s_1 + s_2$.

We prove in Proposition 4.9 $\mathrm{AffPow}(f) = \mathrm{UnivAffPow}(f)$. Notice first that every univariate polynomial g of degree d satisfies that $\mathrm{AffPow}(g) \leq r := \lceil \frac{d+1}{2} \rceil$. Moreover, if $\mathrm{AffPow}(g) = r$, then g admits an expression as $\sum_{i=1}^{r} \alpha_i(x + a_i)^{e_i}$ with $d = e_1$ and $e_i - e_{i+1} \geq 2$ for all i and, thus, $e_r \in \{0, 1\}$ (see [8, Proposition 18]).

Lemma 4.6. *Let* $f_i \in \mathbb{F}[x_i]$ *polynomials of degree* d_i *for* $i = 1, 2$. *Then,*

$$\mathit{UnivAffPow}(f_1 + f_2) \leq \left\lceil \frac{d_1 + 1}{2} \right\rceil + \left\lceil \frac{d_2 + 1}{2} \right\rceil - 1.$$

Proof. Let $s_i := \mathrm{AffPow}(f_i)$ for $i = 1, 2$. If $s_i < \lceil \frac{d_i+1}{2} \rceil$ for some i, the result follows directly since $\mathrm{UnivAffPow}(f_1 + f_2) \leq s_1 + s_2$. Otherwise, $s_i = \lceil \frac{d_i+1}{2} \rceil$ for $i = 1, 2$; in this case both f_i can be written in an optimal way that uses a term of degree ≤ 1; hence, $\mathrm{UnivAffPow}(f_1 + f_2) = s_1 + s_2 - 1$, proving the result. □

Lemma 4.7. *Let* $s, d \in \mathbb{Z}^+$ *and* $b \in (\mathbb{F}^*)^s$ *such that* $b_i \neq b_j$. *If*

$$\lambda_1 x_1^d + \lambda_2 x_2^d = \sum_{i=1}^{s} \gamma_i (x_1 + b_i x_2)^d, \tag{2}$$

with $\lambda_1, \lambda_2 \in \mathbb{F}$ *and* $\gamma_i \in \mathbb{F}$ *not all zero, then* $s \geq d$. *Moreover, if* $\lambda_1 = 0$ *or* $\lambda_2 = 0$ *then* $s \geq d + 1$; *and if* $\lambda_1 = \lambda_2 = 0$, *then* $s \geq d + 2$.

Proof. Consider $c \in \mathbb{F}^*$ different from b_i for all i and the evaluation map φ induced by $x_1 \mapsto -cx$, $x_2 \mapsto x + 1$. Then, $\varphi(x_1 + b_i x_2) = (b_i - c)(x + \frac{b_i}{b_i - c})$. Thus, by applying φ in (2) we get that $\{x^d, (x + 1)^d, (x + \frac{b_i}{b_i - c})^d \mid 1 \leq i \leq s\}$ is linearly dependent. Hence, the result follows from the well-known fact that for every $r < d + 2$ and $c_1, \ldots, c_r$ different elements of $\mathbb{F}$, then the set $\{(x + c_i)^d \mid 1 \leq i \leq r\}$ is $\mathbb{F}$-linearly independent. □

Lemma 4.8. *Let* $f = \sum_{i=1}^{s} \alpha_i(x_1 + b_i x_2 + c_i)^{e_i} \in \mathbb{F}[x_1, x_2]$ *be a polynomial of degree* ≥ 2, *with* $\alpha_i, b_i \in \mathbb{F}^*$ *for all* $i \in [\![1, s]\!]$. *If* $f = f_1 + f_2$ *with* $f_i \in \mathbb{F}[x_i]$, *then* $s \geq \mathit{UnivAffPow}(f_1 + f_2)$.

Proof. Let $\ell_i := x_1 + b_i x_2 + c_i$ for $i = 1, \ldots, s$, and $d_j := \deg(f_j)$ for $j = 1, 2$. We know that $s_i := \mathrm{AffPow}(f_i) \leq \lceil (d_i + 1)/2 \rceil$ and we assume that $d_1 \geq d_2$.

For all $e \in \mathbb{N}$, consider $[f]_e$, the homogeneous component of degree e of f. We have:

$$[f_1]_e + [f_2]_e = \sum_{e_i \geq e} \alpha_i [\ell_i^{e_i}]_e \in \langle (x_1 + b_i x_2)^e \mid e_i \geq e \rangle.$$

Case 1: $d_1 > d_2$. We have that $0 \neq [f_1]_{d_1} = \sum_{e_i \geq d_1} \gamma_i (x_1 + b_i x_2)^{d_1}$ with $\gamma_i \in \mathbb{F}$, hence by Lemma 4.7 there are at least $d_1 + 1$ exponents e_i that are $\geq d_1$. So, by Lemma 4.6 we get

$$s \geq d_1 + 1 \geq \left\lceil \frac{d_1 + 1}{2} \right\rceil + \left\lceil \frac{d_2 + 1}{2} \right\rceil > \mathrm{UnivAffPow}(f_1 + f_2).$$

Case 2: $d_1 = d_2$. We have that $[f_1]_{d_1} + [f_2]_{d_2} = \sum_{e_i \geq d_1} \gamma_i (x_1 + b_i x_2)^{d_1}$ with $\gamma_i \in \mathbb{F}$, hence by Lemma 4.7 there are at least d_1 exponents bigger than or equal to d_1. So, by Lemma 4.6 we get

$$s \geq |\{i : e_i \geq d_1\}| \geq d_1 \geq 2 \left\lceil \frac{d_1 + 1}{2} \right\rceil - 2 \geq \mathrm{UnivAffPow}(f_1 + f_2) - 1.$$

If one of these inequalities is strict, the result is proved; so assume by contradiction that they are all equalities. In particular, we have that $b_i \neq b_j$ for all $i \neq j$. We claim that $e_i = d_1$ for all i. Otherwise, taking $e := \max(e_i) > d_1$ and observing the homogeneous component of degree e, we get that $0 = \sum_{e_i = e} \alpha_i (x_1 + b_i x_2)^e$; but again by Lemma 4.7, this implies that the number of ℓ_i with $e_i = e$ is at least $e + 2 \geq d_1 + 3 > s$, a contradiction. Hence,

$$f_1 + f_2 = \sum_{i=1}^{s} \alpha_i (x_1 + b_i x_2 + c_i)^{d_1}.$$

Now set $\beta_i \in \mathbb{F}$ the (only) root of the derivative of order $d_1 - 1$ of f_i and consider $g_i(x_i) := f_i(x_i + \beta_i)$. We have that $g_1 + g_2 = \sum_{i=1}^{s} \alpha_i (x_1 + b_i x_2 + c_i')^{d_1}$. Since $[g_1]_{d_1 - 1} = [g_2]_{d_1 - 1} = 0$; the homogeneous component of degree $d_1 - 1$ in this expression is

$$0 = \sum_{i=1}^{s} d_1 \alpha_i c_i' (x_1 + b_i x_2)^{d_1 - 1}.$$

Since $s < d_1 + 2$, Lemma 4.7 yields that $c_i' = 0$ for all i. Since $f_1(x_1 + \beta_1), f_2(x_2 + \beta_2)$ are univariate polynomials, then $f_1(x_1 + \beta_1) + f_2(x_2 + \beta_2) = \gamma_1 x_1^{d_1} + \gamma_2 x_2^{d_2}$. However, this implies that $\mathrm{AffPow}(f_1) = \mathrm{AffPow}(f_2) = 1$ and, then, $1 \geq \mathrm{UnivAffPow}(f) - 1 = d_1 = d_2$, a contradiction. □

As a consequence of Lemma 4.8, we obtain the main result of this subsection:

Proposition 4.9. *Let* $f_1 \in \mathbb{F}[x_1]$ *and* $f_2 \in \mathbb{F}[x_2]$, *then*

$$\mathit{AffPow}(f_1 + f_2) = \mathit{UnivAffPow}(f_1 + f_2).$$

Proof. It is obvious that $\mathrm{AffPow}(f_1 + f_2) \leq \mathrm{UnivAffPow}(f_1 + f_2)$. Let $s := \mathrm{AffPow}(f_1 + f_2)$ and consider $f_1 + f_2 = \sum_{i=1}^{s} \ell_i^{e_i}$ an optimal expression of $f_1 + f_2$ in Model (1). We write $\ell_i = a_i x_1 + b_i x_2 + c_i$ with $a_i, b_i, c_i \in \mathbb{F}$. Set $g := f_1 + f_2 - \sum_{\substack{b_i = 0 \\ \text{or } c_i = 0}} \ell_i^{e_i}$. Clearly, g is a sum of two univariate polynomials and can be written as

$$g = \sum_{\substack{a_i \neq 0 \\ b_i \neq 0}} \ell_i^{e_i} = \sum_{\substack{a_i \neq 0 \\ b_i \neq 0}} a_i^{e_i} \left(x_1 + \frac{b_i}{a_i} x_2 + \frac{c_i}{a_i} \right)^{e_i}.$$

Setting $r := |\{i : a_i \neq 0 \text{ and } b_i \neq 0\}|$, by Lemma 4.8 we have $\mathrm{UnivAffPow}(g) \leq r$. Hence we can rewrite g as $g = \sum_{i=1}^{r'} (\alpha_i x + \beta_i y + \gamma_i)^{d_i}$ with either $\alpha_i = 0$, $\beta_i = 0$ or $d_i = 1$, and $r' \leq r$. As a consequence, $f = \sum_{i=1}^{r'} (\alpha_i x + \beta_i y + \gamma_i)^{d_i} + \sum_{\substack{b_i = 0 \\ \text{or } c_i = 0}} \ell_i^{e_i}$ is an

expression of f with $s-r+r'$ terms. Since $s-r+r' \leq s$, this shows that $\mathrm{UnivAffPow}(f_1+f_2) \leq s = \mathrm{AffPow}(f_1+f_2)$.

5 ALLOWING MORE AFFINE FORMS

In what follows we investigate the case where the number of affine forms used to express f in Model (1) is greater than the number of essential variables. The most basic such case is when $f \equiv g$ with $g = \sum_{i=1}^{n} x_i^{e_i} + \ell^e$, where ℓ is an affine form and $e \in \mathbb{N}^*$. Let us first see why the algorithm of Section 3 cannot be straightforwardly generalised to recover the optimal expression of f. We set $h := g - \ell^e$ so that we have $H_g = H_h + H_{\ell^e}$ by linearity of differentiation. Notice that $H_{\ell^e} = e^{\underline{2}}\, \ell^{e-2} \beta\beta^T$, where β is the column vector associated to the coefficients of ℓ and $e^{\underline{i}} := e \cdots (e-i+1)$. In order to compute $\det(H_g)$, we use the *matrix determinant lemma*. We therefore have $\det(H_g) = \det(H_h) + e^{\underline{2}}\, \ell^{e-2} \beta^T \operatorname{adj}(H_h)\beta$. Hence, if $f = g(A \cdot X + b)$, Lemma 3.4 implies that

$$\det(H_f) = \det(A)^2 \left(\prod_{i=1}^{n} e_i^{\underline{2}}\, \ell_i(X)^{e_i-2} + e^{\underline{2}}\, \ell(A \cdot X + b)^{e-2}\, P(X) \right)$$

with $P(X) = \sum_{i=1}^{n} \beta_i^2 \left(\prod_{j \neq i} e_j^{\underline{2}}\, \ell_j(X)^{e_i-2} \right) \in \mathbb{F}[X]$. In most cases neither the ℓ_i's nor ℓ are factors of $\det(H_f)$, which makes the (straightforward generalization of) algorithm of Section 3 fail.

The main idea we propose to generalize the algorithm is to consider an extension of the Hessian by looking at higher order derivatives. We will therefore consider the *symmetric 4-th order Hessian* $\overline{H}_f \in \mathcal{M}_{\binom{n+1}{2}}(\mathbb{F}[X])$ whose entries are:

$$\forall a \leq b,\ i \leq j, \quad (\overline{H}_f)_{(a,b),(i,j)} = \frac{\partial^4 f}{\partial x_a \partial x_b \partial x_i \partial x_j}.$$

In this section, we will design a randomized algorithm that can reconstruct a decomposition in Model (1) that uses up to $\binom{n+1}{2}$ distinct affine forms. However, it will not work for all input polynomials of such type. Indeed, it will work whenever all the exponents involved in the optimal expression of f are ≥ 5 and a certain matrix U, which depends on the affine forms involved, is invertible. We will conduct a randomized analysis to show that our method is correct with high probability (over the choice of the input polynomial and of the internal coin tosses of the algorithm). We begin by proving an analogue of Lemma 3.4 for the symmetric 4-th order Hessian.

Proposition 5.1. *Let $n \in \mathbb{N}^*$, $m := \binom{n+1}{2}$ and $f = \sum_{i=1}^{m} \ell_i^{e_i}$, with $\ell_i = \sum_{j=1}^{n} b_{i,j} x_j + b_{i,0}$ affine forms and $e_i \geq 4$ for all i. Let U be the square $m \times m$ matrix with entries $U_{(i,j),k} := b_{k,i}\, b_{k,j}$ for all $1 \leq k \leq m$, $1 \leq i \leq j \leq n$. If $\det(U) \neq 0$, there exists $c \neq 0$ such that*

$$\det(\overline{H}_f(X)) = c \cdot \prod_{i=1}^{m} \ell_i^{e_i-4},$$

Proof. By linearity of the symmetric 4-th order Hessian, we have

$$\overline{H}_f(X) = \sum_{k=1}^{m} \overline{H}_{\ell_k}(X) = \sum_{k=1}^{m} e_k^{\underline{4}}\, \ell_k^{e_k-4} (u_k \cdot u_k^T) = U \cdot D \cdot U^T,$$

where $D = \mathrm{Diag}(e_1^{\underline{4}}\, \ell_1^{e_1-4}, \ldots, e_m^{\underline{4}}\, \ell_m^{e_m-4})$, and u_k is the column vector whose (i,j)-th entry is $b_{k,i} b_{k,j}$ with $1 \leq i \leq j \leq n$. Thus, $\det(\overline{H}_f(X)) = \det(U)^2 \prod_{k=1}^{m} e_k^{\underline{4}}\, \ell_k^{e_k-4}$. □

Now, we are going to prove that if the coefficients of the ℓ_i are chosen uniformly at random, then with a high probability we have $\det(U) \neq 0$. Thus, whenever $e_i \geq 5$ for all i, one can find ℓ_i as a factor of $\det(\overline{H}_f(X))$ of multiplicity $e_i - 4$.

Lemma 5.2. *Let $n \in \mathbb{N}^*$ and $m := \binom{n+1}{2}$, and consider the set of variables $\mathcal{V} := \{y_{(k,l),i} \mid 1 \leq k \leq l \leq n,\ 1 \leq i \leq n\}$. Let U be the $m \times m$ square matrix with entries $U_{(i,j),(k,l)} := y_{(k,l),i}\, y_{(k,l),j}$, where $1 \leq i \leq j \leq n$, $1 \leq k \leq l \leq n$. Then, $\det(U) \in \mathbb{Z}[\mathcal{V}]$ is a nonzero polynomial of degree $2m$.*

Proof. Since all the entries of the matrix are homogeneous polynomials of degree 2, it is clear that $\det(U)$ is either zero or a polynomial of degree $2m$. To prove that $\det(U) \neq 0$ it suffices to exhibit a nonzero evaluation of $\det(U)$. We consider the matrix $\tilde{U}$ given by the evaluation $y_{(k,l),i} \mapsto 1$ if $i \in \{k,l\}$; or $y_{(k,l),i} \mapsto 0$ otherwise. By ordering pairs (i,j) with $i = j$ first, we obtain the following shape

$$\tilde{U} = \begin{array}{c} \\ i=j \\ i<j \end{array} \begin{array}{c} \begin{array}{cc} k=l & k<l \end{array} \\ \begin{pmatrix} \mathrm{Id}_n & (*) \\ 0 & \mathrm{Id}_{m-n} \end{pmatrix} \end{array},$$

proving that $\det(\tilde{U}) = 1$ and therefore that $\det(U) \neq 0$. □

Theorem 5.3. *Let $n \geq 2$ and $m := \binom{n+1}{2}$. Let $\ell_i = \sum_{j=1}^{n} b_{i,j} x_j + b_{i,0} : 1 \leq i \leq m$ whose coefficients $b_{i,j}$ are taken uniformly at random from a finite set S and take $f := \sum_{i=1}^{m} \ell_i^{e_i} \in \mathbb{F}[X]$ with $e_i \geq 4$ for all i. Then, $\det(\overline{H}_f(X)) \neq 0$ with probability at least $1 - \frac{2m}{|S|}$.*

Proof. By Proposition 5.1, it is enough to show that $\det(U) \neq 0$, where U is the matrix defined by $U_{(i,j),k} = b_{k,i} b_{k,j}$. By Schwartz-Zippel lemma and Lemma 5.2, the probability that $\det(U) \neq 0$ is at least $1 - \frac{2m}{|S|}$. □

This theorem suggests a polynomial time algorithm for finding an optimal expression of a polynomial f with high probability when $\mathrm{AffPow}(f) \leq m = \binom{n+1}{2}$, the affine forms in optimal expression of f are chosen at random from a finite set and all the exponents involved are ≥ 5. It is enough to start with $k = m-1$, choose randomly k affine forms $t_1, \ldots, t_k$ with exponents $d_i = 4$ and denote $g := f + \sum_{i=1}^{k} t_i^{d_i}$. If $D := \det(\overline{H}_f(X)) = 0$, we decrease the value of k by one unit and repeat the argument, or we reject if $k = 0$. If $D \neq 0$, we factorize it. If D splits into linear factors $l_1, \ldots, l_{m-k}$ of multiplicities $r_1, \ldots, r_{m-k}$ and $f \in \langle l_i^{r_i+4} \mid 1 \leq i \leq m-k \rangle$, then $\mathrm{AffPow}(f) = m-k$ and we output the optimal expression. Otherwise, we reject.

6 UNIVARIATE PROJECTIONS

We denote by n_e the number of exponents smaller than $e \in \mathbb{N}$, i.e., $n_e = \#\{i : e_i \leq e\}$. The main result of this section is an algorithm that finds the optimal reconstruction under the condition on n_e being small. We will proceed by reduction to the univariate case: we solve n univariate projections of the multivariate problem using algorithms from [9], and then "lift" them to a solution of the multivariate problem.

6.1 Essentially unique optimal expressions

The notion of *essentially equal* expressions was introduced in [16]. We say that $\sum_{i=1}^{s} \ell_i^{e_i} = \sum_{i=1}^{r} t_i^{d_i}$ are essentially equal if $r = s$ and after a permutation $\ell_i^{e_i} = t_i^{d_i}$ for all i. Likewise, we say that f has

an *essentially unique* optimal decomposition in Model (1) if any two optimal decompositions of f are essentially equal. The following result extends [9, Corollary 3.14], providing a sufficient condition to have an essentially unique optimal decomposition.

Proposition 6.1. *Let $f = \sum_{i=1}^{s} \ell_i^{e_i} \in \mathbb{F}[X]$, where the ℓ_i are affine forms, and ℓ_i is not proportional to ℓ_j whenever $e_i = e_j$. If $n_e \leq \sqrt{\frac{e+1}{2}}$ for all $e \in \mathbb{N}$, then AffPow$(f) = s$ and the optimal representation of f is essentially unique.*

Proof. Let $r := \text{AffPow}(f) \leq s$ and let $f = \sum_{i=s+1}^{s+r} \ell_i^{e_i}$ 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 $\varphi : \mathbb{F}[X] \to \mathbb{F}[x]$ induced by $x_i \mapsto \omega_i x + \lambda_i$ where $\omega, \lambda \in \mathbb{F}^n$. We denote $\varphi(\ell_i) = b_i x + c_i$ and choose ω and λ (a generic choice would suffice) so that

(1.a) $\varphi(\ell_i)^{e_i} = \varphi(\ell_j)^{e_j}$ if and only if $\ell_i^{e_i} = \ell_j^{e_j}$, and
(1.b) whenever $e_i = e_j$ with $1 \leq i < j \leq s$, then $c_i/b_i \neq c_j/b_j$.

$$\text{Then, } \varphi(f) = \sum_{i=1}^{s} \varphi(\ell_i)^{e_i} = \sum_{i=1}^{s} b_i^{e_i}(x + c_i/b_i)^{e_i}$$
$$= \sum_{i=s+1}^{s+r} \varphi(\ell_i)^{e_i} = \sum_{i=s+1}^{s+r} b_i^{e_i}(x + c_i/b_i)^{e_i}.$$

We consider the expression $\varphi(f)$ in the univariate Affine Power model. Since (1.b) holds and $n_e \leq \sqrt{\frac{e+1}{2}}$ for all $e \in \mathbb{N}$, by [9, Corollary 3.14] we get that $r = s$ and that both expressions for $\varphi(f)$ are the same. By (1.a) we obtain the result. □

6.2 Projection and recovery

Our goal is to provide an algorithm that, given blackbox access to $f \in \mathbb{F}[X]$, computes $s = \text{AffPow}(f)$ and an optimal expression for f. It is a multivariate analogue of [9, Theorem 4.5] where the condition of "distinct nodes" is replaced by "the ℓ_i's in the decomposition are not proportional". The idea of the algorithm is to perform a random change of coordinates and then project to n univariate problems that we solve using [9, Theorem 4.5]. One minor difficulty is that the univariate algorithms of [9] are presented for polynomials given in dense representation rather than in black box representation, but we can obtain the dense representation of a univariate polynomial by random evaluations and, then, interpolation.

Theorem 6.2. *Let $f = \sum_{i=1}^{s} \ell_i^{e_i} \in \mathbb{F}[X]$, where the ℓ_i are pairwise non-propoportional linear forms, and $e_i \in \mathbb{N}$. Assume that $n_i \leq (3i/4)^{1/3} - 1$ for all $i \geq 2$. Then, AffPow$(f) = s$ and there is a randomized algorithm* `MultiBuild`(f) *that, given access to a black box for f, computes the set of terms $T(f) = \{\ell_i^{e_i} \mid 1 \leq i \leq s\}$. The algorithm runs in time polynomial in n and d, and works as follows:*

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).*
2. *For each $j \in [\![1, n]\!]$, we set $g_j := \pi_j(g)$ where π_j is induced by $x_k \mapsto 0$ if $k \neq j$ and $x_j \mapsto x$.*
3. *Apply* `Build`(g_j) *from [9, Theorem 4.5] to obtain $s_j := \text{AffPow}(g_j)$ and the triplets $(\beta_{ij}, b_{ij}, e_{ij})$ such that $g_j = \sum_{i=1}^{s_j} \beta_{ij}(x + b_{ij})^{e_{ij}}$.*
4. *We define $P_j := \{(c_{ij}, 1/b_{ij}, e_{ij}) \mid c_{ij} := \beta_{ij}b_{ij}^{e_{ij}},\ 1 \leq i \leq s_i\}$.*
5. *We 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 $i \in \{1, \ldots, s_1\}$.*
6. *If $g = \sum_{i=1}^{s} c_i(1 + \sum_{j=1}^{n} x_j/b_{ij})^{e_i}$, we output $f = \phi^{-1}(g)$.*

Or we reject if any of these steps is not feasible.

Proof. We observe that AffPow$(f) = s$ and the optimal representation of f is essentially unique by Proposition 6.1.

With high probability we have that ϕ is invertible and $g = \sum_{i=1}^{s} t_i^{e_i}$ with $t_i = \sum_{j=1}^{n} a_{ij}x_j + a_{i0}$ satisfies that:

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

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

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

Since $\pi_j(g)$ satisfies the hypotheses of [9, Theorem 4.5], `Build`$(\pi_j(g))$ outputs $\left\{(a_{ij}^{e_i}, \frac{a_{i0}}{a_{ij}}, e_i) \mid 1 \leq i \leq s\right\}$. From these values we obtain $P_j = \left\{(a_{i0}^{e_i}, \frac{a_{ij}}{a_{i0}}, e_i) \mid 1 \leq i \leq s\right\}$. The uniqueness of the expression of g_j for all j and (iii) guarantee that we recover g in **Step 6**.

REFERENCES

[1] J. Alexander, A. Hirschowitz. Polynomial interpolation in several variables. *Journal of Algebraic Geometry*, 4(2):201–222, 1995.
[2] M. Ben-Or, P. Tiwari. A deterministic algorithm for sparse multivariate polynomial interpolation. In *Proc. 20th annual ACM Symposium on Theory of Computing*, 1988.
[3] J. Brachat, P. Comon, B. Mourrain, E. Tsigaridas. Symmetric tensor decomposition. Linear Algebra and its Applications 433, 11:1851-1872, 2010.
[4] M. C. Brambilla, G. Ottaviani. On the Alexander –Hirschowitz theorem. *Journal of Pure and Applied Algebra*, 212(5):1229–1251, 2008.
[5] E. Carlini. Reducing the number of variables of a polynomial. In *Algebraic geometry and geometric modeling*, Math. Vis., 237–247. Springer, Berlin, 2006.
[6] E. Carlini, M. V. Catalisano, L. Chiantini. Progress on the symmetric Strassen conjecture. *J. Pure Appl. Algebra*, 219(8):3149–3157, 2015.
[7] E. Carlini, M. V. Catalisano, A. V. Geramita. The solution to the Waring problem for monomials and the sum of coprime monomials. *J. of Algebra*, 370:5–14, 2012.
[8] I. García-Marco, P. Koiran. Lower bounds by Birkhoff interpolation. *J. Complexity*, 39:38–50, 2017.
[9] I. García-Marco, P. Koiran, T. Pecatte. Reconstruction algorithms for sums of affine powers. *arXiv preprint arXiv:1607.05420.* Conference version in: *Proceedings of the ISSAC '17*, pages 317–324, 2017.
[10] I. García-Marco, P. Koiran, T. Pecatte. On the linear independence of shifted powers. J. Complexity 45 (2018), 67–82.
[11] A. Iarrobino, V. Kanev. *Power Sums, Gorenstein Algebras, and Determinantal Loci.* Springer, 1999.
[12] E. Kaltofen, B. M. Trager. Computing with polynomials given by black boxes for their evaluations: greatest common divisors, factorization, separation of numerators and denominators. *J. Symbolic Comput.*, 9(3):301–320, 1990.
[13] N. Kayal. An exponential lower bound for the sum of powers of bounded degree polynomials. *Electronic Colloquium on Computational Complexity (ECCC)*, 19, 2012.
[14] 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.
[15] N. Kayal. Efficient algorithms for some special cases of the polynomial equivalence problem. In *Symposium on Discrete Algorithms (SODA)*. January 2011.
[16] N. Kayal. Affine projections of polynomials. In *Proceedings of the 44th Annual ACM Symposium on Theory of Computing*, pages 643–662. ACM, 2012.
[17] J. M. Landsberg, Z. Teitler. On the ranks and border ranks of symmetric tensors. *Foundations of Computational Mathematics*, 10(3):339–366, 2010.
[18] G. Labahn, M. Giesbrecht, W.-S. Lee. Symbolic-numeric sparse interpolation of multivariate polynomials. *J. Symbolic Computation*, 44(8):943–959, 2009.
[19] W. Lee, M. Giesbrecht, E. Kaltofen. Algorithms for computing sparsest shifts of polynomials in power, chebyshev and pochhammer bases. *J. Symbolic Computation*, 36(3-4):401–424, 2003.
[20] L. Oeding, G. Ottaviani. Eigenvectors of tensors and algorithms for Waring decomposition. *J. of Symbolic Computation*, 54:9–35,2013.
[21] Y. Shitov. How hard is the tensor rank?. CoRR, arXiv:1611.01559, 2016.
[22] V. Strassen. Vermeidung von Divisionen. *J. Reine Angew. Math.*, 264:184–202, 1973.