2018 IEEE 59th Annual Symposium on Foundations of Computer Science

Deterministic Factorization of Sparse Polynomials with Bounded Individual Degree

Vishwas Bhargava
Department of Computer Science, Rutgers University
Piscataway, NJ 08854, USA
Email: vishwas1384@gmail.com

Shubhangi Saraf
Department of Computer Science & Department of Mathematics
Rutgers University
Piscataway, NJ 08854, USA
Email: shubhangi.saraf@gmail.com

Ilya Volkovich
Department of EECS, CSE Division, University of Michigan, Ann Arbor
MI 48109, USA
Email: ilyavol@umich.edu

***Abstract*—In this paper we study the problem of deterministic factorization of sparse polynomials. We show that if f is an n-variate polynomial with s monomials, with individual degrees of its variables bounded by d, then f can be deterministically factored in time $s^{\mathrm{poly}(d)\log n}$. Prior to our work, the only efficient factoring algorithms known for this class of polynomials were randomized, and other than for the cases of $d = 1$ and $d = 2$, only exponential time deterministic factoring algorithms were known.**

A crucial ingredient in our proof is a quasi-polynomial sparsity bound for factors of sparse polynomials of bounded individual degree. In particular we show if f is an s-sparse polynomial in n variables, with individual degrees of its variables bounded by d, then the sparsity of each factor of f is bounded by $s^{\mathcal{O}(d^2 \log n)}$. This is the first nontrivial bound on factor sparsity for $d > 2$. Our sparsity bound uses techniques from convex geometry, such as the theory of Newton polytopes and an approximate version of the classical Carathéodory's Theorem.

Our work addresses and partially answers a question of von zur Gathen and Kaltofen (JCSS 1985) who asked whether a quasi-polynomial bound holds for the sparsity of factors of sparse polynomials.

***Keywords*-Bounds on Factor Sparsity; Multivariate Polynomial Factorization; Sparse polynomials**

I. Introduction

Polynomial factorization is one of the most fundamental questions in computational algebra. The problem of multivariate polynomial factorization asks the following: Given $f \in \mathbb{F}[x_1, x_2 \ldots, x_n]$ a multivariate polynomial over a field $\mathbb{F}$, compute each of the irreducible factors of f. Other than being natural and central, the problem has many applications in areas such as list decoding [1, 2], derandomization [3] and cryptography [4].

There has been a large body of research studying efficient algorithms for this problem (see e.g. [5]) and numerous *randomized* algorithms were designed [6, 7, 8, 9, 5]. However, the question of whether there exist *deterministic* algorithms for this problem remains an important and interesting open question (see [5]).

Another fundamental question in algebraic complexity is the problem of Polynomial Identity Testing (PIT). The problem of PIT asks the following: Given a polynomial $f \in \mathbb{F}[x_1, x_2, \ldots, x_n]$ represented by a small arithmetic circuit, determine if the polynomial is identically 0. In a recent work, Kopparty et al [10] showed that the problem of derandomizing multivariate polynomial factorization is *equivalent* to the problem of derandomizing polynomial identity testing for general arithmetic circuits. They showed this result in both the white-box and the black-box settings. We already know deterministic PIT algorithms for several interesting classes of arithmetic circuits, and this raises the very natural question of whether we can derandomize *polynomial factoring* for these classes. Perhaps the most natural such class of polynomials is the class of *sparse* polynomials.

The sparsity of f, denoted $\|f\|$, is the number of monomials (with non zero coefficients) appearing in f. For instance, the sparsity of the polynomial $x_1 + x_2^3 + x_3x_4 + 20$ is four.

Factoring of sparse polynomials has been studied for over three decades. It was initiated by the work of von zur Gathen and Kaltofen [6] that gives the first *randomized* algorithm for factorization of sparse multivariate polynomials. The runtime of this algorithm has polynomial dependence on the sparsity of the *factors* of the underlying polynomial, and thus, very naturally, this work raised the question of whether one can find efficient bounds on the sparsity of factors of a sparse polynomial.

In this paper, we consider the following two problems:

2575-8454/18/$31.00 ©2018 IEEE
DOI 10.1109/FOCS.2018.00053

(1) Prove efficient bounds on the sparsity of the factors of sparse polynomials. (2) Derandomize polynomial factorization for sparse polynomials.

Indeed, these are extremely natural questions to study. However already for general fields, we know that one cannot hope to prove a strong sparsity bound for the factors of a sparse polynomial.

In this paper, we focus our attention on the class of sparse polynomials with bounded individual degree, i.e. for some parameter d, we limit the degree of each variable x_i to be at most d.

One very interesting such class of polynomials is the class of sparse multilinear polynomials ($d = 1$). This is the simplest case of sparse polynomials with bounded degree. In [11], Shpilka and Volkovich gave a derandomization for the problem of polynomial factorization for this class. Factor sparsity bounds are fairly easy to show for this class of polynomials, and armed with the sparsity bound and a technique for derandomizing a certain PIT problem that arises, they were able to derandomize factoring in this case. This was extended to the case $d = 2$ in the work of Volkovich [12], again by first showing a sparsity bound for the factors of polynomials of individual degree 2, and then showing how to derandomize the polynomial factorization problem. For $d > 2$, the techniques used by the above works for proving sparsity bounds on the factors of a polynomial seem to break down.

In a recent beautiful work, Oliveira [13] showed that the factors of sparse polynomials of bounded individual degree can be computed by small *depth-7* circuits. This again raises the very natural question: What is the size of the best *depth-2* circuit computing the factors of a sparse polynomial of bounded individual degree. This is precisely the problem of proving sparsity bounds for the factors of a sparse polynomial of bounded individual degree, which is a question we study in this paper.

The other question that we address in this work is the problem of deterministically factoring sparse polynomials of bounded individual degree. A bound on the sparsity of the factors of such a polynomial just implies that the factors will have an efficient representation as a sum of monomials. However in order to actually obtain the factors deterministically, there are several additional derandomization hurdles to overcome.

A. Our Results

In this paper we give the first deterministic quasi-polynomial time algorithm for factoring sparse polynomials of bounded individual degree. Prior to our work, only efficient randomized factoring algorithms were known for this class of polynomials, and other than for the cases of $d = 1$ [11] and $d = 2$ [12] only exponential time deterministic factoring algorithms were known.

A crucial ingredient of our proof is a quasi-polynomial size sparsity bound for factors of sparse polynomials of bounded individual degree d. In particular, we show that if f is an s-sparse polynomial in n variables with individual degrees of its variables bounded by d, then f can be deterministic factored in time $s^{\mathrm{poly}(d)\log n}$. This is the first nontrivial bound on factor sparsity for any $d > 2$. Our sparsity bound uses techniques from convex geometry, such as the theory of Newton polytopes and an approximate version of the classical Carathéodory's Theorem.

We say that a polynomial $f \in \mathbb{F}[x_1, x_2, \ldots, x_n]$ has *sparsity* s if it has at most s nonzero monomials. We say that it has individual degree at most d if the maximum degree in each of its variables is bounded above by d.

We formally state below our factor sparsity bound and then our result on deterministic factoring.

Theorem 1 (Factor Sparsity Bound)**.** *Let $\mathbb{F}$ be an arbitrary field (finite or otherwise) and let $f \in \mathbb{F}[x_1, x_2, \ldots, x_n]$ be a polynomial of sparsity s and individual degrees at most d, then the sparsity of every factor of f is bounded by $s^{\mathcal{O}(d^2 \log n)}$.*

Remark I.1. *Note that for $d = \mathrm{polylog}(n)$, we obtain a quasi-polynomial sparsity bound on the factors of f. Indeed when $s = \mathrm{poly}(n)$, for any $d = o(\sqrt{n}/\log^2 n)$, we obtain a nontrivial sparsity bound on the factors of f.*

Given a polynomial $f \in \mathbb{F}[x_1, x_2, \ldots, x_n]$, the *complete factorization* of f is a representation of f as a product $h_1^{e_1} \cdots h_m^{e_m}$, where $h_1, h_2, \ldots, h_m$-s are pairwise co-prime, irreducible polynomials, and $e_1, e_2, \ldots, e_m$ are positive integers. This representation is unique up to reordering of the h_i.

Theorem 2 (Main)**.** *There exists a deterministic algorithm that given a polynomial $f \in \mathbb{F}[x_1, x_2, \ldots, x_n]$ of sparsity s and individual degrees at most d, computes the complete factorization of f, using $s^{\mathcal{O}(d^7 \log n)} \cdot \mathrm{poly}(\mathrm{c}_{\mathbb{F}}(d^2))$ field operations, where $\mathrm{c}_{\mathbb{F}}(d)$ denotes the time of the best known algorithm that factors a univariate polynomial of degree d over $\mathbb{F}$.*

Remark I.2. *A more refined version of Theorem 2 is given in Theorem V.7. The run time for the deterministic factoring algorithm in Theorem V.7 gives the precise*

dependence on the sparsity bound for factors of sparse polynomials. In particular, if one could improve the sparsity bound, then one could plug it into the statement of Theorem V.7 to get an improved run time for the deterministic factoring algorithm.

B. Related Work

The study of sparse polynomial factorization was initiated in [6], where the first *randomized* algorithm for the factorization of sparse polynomials was given. The runtime of this algorithm was polynomial in the sparsity of the factors, and in this work, von zur Gathen and Kaltofen explicitly raised the question of proving improved sparsity bounds for the factors of sparse polynomials.

In [14], Dvir and Oliveira gave an elegant approach for bounding the sparsity of factors of a general sparse polynomial by studying the Newton polytopes of the polynomial and its factors. This approach did not eventually lead to an efficient sparsity bound. However it did inspire our work and our approach of using techniques from convex geometry to bound the factors of sparse polynomials.

In [11], Shpilka and Volkovich gave efficient deterministic factoring algorithms for sparse multilinear polynomials. This result was extended in [15] to the model of sparse polynomials that split into multilinear factors. In [12], Volkovich gave an efficient deterministic factorization algorithm for sparse multiquadratic polynomials. The results [11, 12] correspond to the special case when the individual degree d equals 1 and 2, respectively. For $d \geq 3$, the proof techniques of both these works broke down, and a new approach was needed.

The problem of multivariate polynomial factorization for polynomials of bounded individual degree was also studied in [13]. In this work, among other things, it was shown that if $f \in \mathbb{F}[x_1, x_2, \ldots, x_n]$ is an s-sparse polynomial of individual degree d, where $\mathbb{F}$ is a field of characteristic 0, then any factor of f can be computed by a depth-7 circuit of size $\mathrm{poly}(dn^d, s)$. In particular if d is constant, then this shows that any factor of f can be computed by a depth-7 circuit with only a polynomial blow-up in size. This is in contrast to our work, where we want to bound the number of *monomials* in the factors of f. In other words, we attempt to represent the factors of f by the more natural class of depth-2 circuits and then understand the size complexity (which we show is quasi-polynomial). We also would like to point out that our result holds over any field $\mathbb{F}$.

Another work that is relevant in this context is the work of Kopparty et al [10] which shows an equivalence between the problems of polynomial identity testing (PIT) and polynomial factorization. In particular, it shows that if one can derandomize PIT for the class of general arithmetic circuits, then one can derandomize polynomial factorization for that same class. Since there are several natural examples of classes of polynomials for which we know deterministic PIT algorithms, this naturally raises the question (which was indeed raised in [10]) of whether one can derandomize factoring for the corresponding classes of polynomials. Sparse polynomials are, perhaps, the most natural example of such a class, and our work makes the first significant advance in this direction.

C. Proof overview

Our proof of the deterministic factoring algorithm has two self- contained and independently interesting components. We first prove a sparsity bound on the factors of sparse polynomials with bounded individual degree (Theorem 1). We then show how such a sparsity bound can be used effectively to derandomize factoring of this same class of polynomials (Theorem 2).

We elaborate on both these components below.

1) Proof Overview for the Sparsity Bound: Theorem 1: Our proof uses tools from convex geometry such as the theory of Newton polytopes and an approximate version of Carathéodory's theorem.

Suppose that $f, g, h \in \mathbb{F}[x_1, x_2, \ldots, x_n]$ are polynomials such that $f = g \cdot h$. We want to show that if f is s-sparse and with bounded individual degree d, then g and h are both at most s' sparse, where $s' = s^{\mathcal{O}(d^2 \log n)}$.

We will show this by instead showing the following slightly more general result. For a polynomial f, let $\|f\|$ denote the sparsity (i.e. the number of nonzero monomials) of f. Suppose that g is any polynomial of individual degree d such that $\|g\| = s$, and suppose that $f = g \cdot h$ (with no assumptions on the degrees of f and h), then $\|f\| \geq s^{\frac{1}{\mathcal{O}(d^2 \log n)}}$. In particular, there is no polynomial h that one can multiply g with, so that the product $g \cdot h$ has an overwhelming cancellation of monomials.

Newton Polytopes and Connection to the Sparsity Bound: Let $f \in \mathbb{F}[x_1, x_2, \ldots, x_n]$ be a polynomial such that:

$$f = \sum a_{i_1 i_2 \ldots i_n} x_1^{i_1} x_2^{i_2} \cdots x_n^{i_n}.$$

One can consider the set

$$\mathrm{Supp}(f) = \{(i_1, i_2, \ldots, i_n) \mid a_{i_1 i_2 \ldots i_n} \neq 0\} \subseteq \mathbb{R}^n$$

of exponent vectors of f. One can then associate a polytope $P_f \subseteq \mathbb{R}^n$, called the Newton polytope of f, which is the convex hull of points in $\mathrm{Supp}(f)$.

A classic fact about Newton polytopes that was observed by Ostrowski [16] in 1921 states that if $f = g \cdot h$, then P_f is the Minkowski sum of P_g and P_h, where for two polytopes A and B, their Minkowski sum $A + B$ is defined to be the set of points $\{u + v \mid u \in A \text{ and } v \in B\}$. Minkowski sums of polytopes are extremely well-studied and it is not difficult to show that the Minkowski sum of two polytopes is itself a polytope. Moreover, if we let $V(P)$ denote the set of vertices (equivalently corner points) of a polytope P, then

$$|V(A+B)| \geq \max\{|V(A)|, |V(B)|\}.$$

Once we have these basic facts about Newton polytopes and Minkowski sums, it follows that a lower bound for $\|f\|$ (in terms of $\|g\|$), follows from a lower bound on $|V(P_f)|$, and in particular from a lower bound on $|V(P_g)|$. Thus, via the theory of Newton polytopes and Minkowski sums, we see that the monomials of g that correspond to the vertices of P_g are very *robust*. There is no way of multiplying g with any other polynomial and obtaining a cancellation of monomials that will make these special monomials corresponding to the vertices of P_g "disappear".

Thus for $f = g \cdot h$, our task of lower bounding $\|f\|$ in terms of $\|g\|$ has reduced to lower bounding $|V(P_g)|$, where P_g is the Newton polytope of a polynomial g such that $\|g\| = s$ and g has individual degree bounded by d. Showing a lower bound on $|V(P_g)|$ will be the main technical core of our proof of the sparsity bound.

We note that this connection between Newton polytopes and sparsity bounds was first made in [14] and indeed it inspired the approach taken in this paper.

Sparsity Bound from Carathéodory's Theorem: Note that in general, for an arbitrary polynomial g, there is no good bound on the number of vertices of P_g in terms of the number of monomials of g. For instance one can easily construct examples of polynomials g with exponential in n many monomials, and such that P_g has only n vertices. Here is an example : consider the polynomial $P_g = (x_1 + x_2 + \cdots + x_n)^n$. It clearly has exponentially many monomials. However P_g has only n vertices, which are the scalings of the coordinate vectors by n.

In the case when g has individual degree bounded by d, we will show that a much nicer bound actually holds. Notice that in this case, $\mathrm{Supp}(g) \subseteq \{0, 1, \ldots, d\}^n$. We will show that if $E \subseteq \{0, 1, \ldots, d\}^n$ is an arbitrary subset of size s, then the convex hull of E (denoted CS(E)) has at least $s^{\frac{1}{d^2 \cdot \log n}}$ vertices. This will immediately imply our sparsity bound.

To show this bound, we will use an approximate version of Carathéodory's theorem. The classic version of Carathéodory's theorem is a fundamental result in convex geometry and it states that if a point $\mu \in \mathbb{R}^n$ lies in the convex hull of a set V, then μ can be written as the convex combination of at most $n + 1$ points of V. A suitable "approximate" version of Carathéodory's theorem suitably applied implies the following: For $E \subseteq \{0, 1, \ldots, d\}^n$, every point $\mu \in E$ can be ε-*approximated* by a k-uniform convex combination of elements of $V(E)$, where $k = \mathcal{O}(d^2 \log n)$. Again, (and this time truly) one can conclude that $|E| \leq |V(E)|^k$, since each subset of $V(E)$ of size k could "approximately recover" at most one element of E via a k-uniform convex combination (by the triangle inequality the same point cannot approximate two different points of E), and each element of E must be approximately recovered by some subset of $V(E)$ of size k. See Theorem III.5 for the statement of the approximate Carathéodory theorem that we use.

2) Proof Overview for the Factoring Algorithm: Theorem 2: Let $f \in \mathbb{F}[y, x_1, x_2, \ldots, x_n]$ be a multivariate polynomial with individual degrees at most d. While in general f could have as many as $d(n+1)$ factors, our starting point is an observation that if f is monic in y, then every factor of y must also be monic in y. Consequently, f has at most d factors (total). This makes the monic case much easier to handle, and we first show how to factorize f when f is monic, and then we show how to extend our algorithm to the general non-monic case.

In the monic case, there are at most d factors. How would we identify these factors? The traditional approach [6, 8, 9] suggests projecting the polynomial into a low-dimensional space, where the factorization problem is easy. Yet, in order to recover the original factors, the factorization "pattern" of f should stay the same upon the projection. That is, every irreducible factor should remain irreducible upon the projection. This is typically achieved by the Hilbert Irreducibility Theorem, which shows that a random projection would achieve this goal. Nonetheless, derandomizing the Irreducibility Theorem appears to be a challenging task. Instead, we take a somewhat different approach.

Finding a "good" Projection: First, we relax the requirement of maintaining the same factorization "pat-

tern" to a requirement that different irreducible factors do not "overlap" upon projection (i.e. have no non-trivial gcd). This is a standard processing step in many factorization algorithms and it is usually taken care of by hitting the Discriminant of the polynomial f (i.e. $\Delta_y(f)$). Yet this approach for obtaining our deterministic algorithm presents its challenges, and it is particularly tricky in the case that the characteristic of the ambient field $\mathbb{F}$ is finite (i.e. $\mathrm{char}(\mathbb{F}) > 0$). We show how to go around these problems.

Formally, let $f \in \mathbb{F}[y, x_1, x_2, \ldots, x_n]$ be monic in y and let $f(y, \overline{x}) = h_1^{e_1}(y, \overline{x}) \ldots h_k^{e_k}(y, \overline{x})$ be the factorization of $f(y, \overline{x})$. We will project f to a univariate polynomial in y by setting all the variables in $\overline{x}$ to elements of $\mathbb{F}$. In order to guarantee that different irreducible factors have no non-trivial gcd after projection, it suffices to find an assignment $\overline{a} \in \mathbb{F}^n$ such that

$$\forall i \neq j : \gcd\left(h_i(y, \overline{a}), h_j(y, \overline{a})\right) = 1.$$

This condition translates into finding a single assignment $\overline{a}$ that hits (i.e. is a nonzero assignment for) the *Resultant*, $\mathrm{Res}_y(h_i, h_j)$, for all $i \neq j$ (see Section II-D for more details). As f is an s-sparse polynomial, by Theorem 1, each h_i is an $s^{\mathcal{O}(d^2 \log n)}$-sparse polynomial. Hence, by the properties of the Resultant (Lemma II.7), $\mathrm{Res}_y(h_i, h_j)$ is $s^{\mathcal{O}(d^3 \log n)}$-sparse polynomial. Consequently, hitting all the pairwise resultants corresponds to hitting their product, which is a (somewhat) sparse polynomial. We handle this in a "black-box" fashion. That is, we iterate over all the points in a hitting set for (somewhat) sparse polynomials (for example using the hitting set of [17]).

Finding the "right" Partition: As the projection we obtain is no longer required to maintain the same factorization "pattern", irreducible factors could split into "pieces" (i.e. further factorize upon projection) in a way that the same set of "pieces" can emerge from different polynomials. For example, consider the polynomials $f(y, x) = (y^2 - x)y$ and $g = y(y - x)(y + x)$. These two polynomials have different factorization patterns. However observe that $f(y, 1) = g(y, 1) = y(y - 1)(y + 1)$. While in both cases, the different "pieces" of the irreducible factors of f and g do not overlap (i.e. no nontrivial gcd), it is not clear how to group the pieces together to recover the factorization pattern of the original polynomial. I.e. just by examining the pieces, we cannot determine what the right partition of the set of factors of $f(y, 1)$ and $g(y, 1)$ should be.

We address this problem by recalling and taking advantage of the fact that a monic polynomial of degree d can split into at most d pieces! Therefore, the are at most $d^{\mathcal{O}(d)}$ possible partitions. We find the "right" partition by iterating over all of them till we find the right one.

Reconstructing the Factors: As before, let $f \in \mathbb{F}[y, x_1, x_2, \ldots, x_n]$ be monic in y and let $f(y, \overline{x}) = h_1^{e_1}(y, \overline{x}) \ldots h_k^{e_k}(y, \overline{x})$. Given a "good" projection $\overline{a}$ and the "right" partition, we will show how to obtain oracle (i.e. "black-box") access the polynomials $h_1, \ldots h_k$. Once we can do this, as Theorem 1 provides us an upper bound on the sparsity of h_i-s, we can use a reconstruction algorithm for sparse polynomials to reconstruct $h_1, \ldots h_k$, given via an oracle access.

We obtain oracle access to $h_1, \ldots h_k$ by mirroring the factorization algorithm of [9]. Given an input point $\overline{b} \in \mathbb{F}^n$ at which we want to compute $h_1(y, \overline{b}), \ldots h_k(y, \overline{b})$, the algorithm uses $\overline{a}$ as an anchor point and draws a line to $\overline{b}$. We then obtain a problem of bi-variate factorization, which we know how to solve efficiently. The non-overlapping property of the "pieces" makes it possible to group the pieces together in the same consistent way for every choice of $\overline{b}$. Once we can do this, this allows us to evaluate the individual factors at $\overline{b}$.

Testing the Purported Factors: As was discussed earlier, given a polynomial f, the algorithm will proceeding by trying to reconstruct the factors of f for every projection and every partition. Some of these projections and partitions will return valid factorizations of f and some might return garbage. We need to prune out the garbage solutions, which we can do as follows: As each factor of f is "somewhat" sparse (Theorem 1) and there are at most d of them, given a purported factorization, we can test if it is a good and valid factorization it by explicitly multiplying out the polynomials.

Clearly, this algorithm will pick up *any* valid factorization of f (not just the irreducible one). We will select the irreducible factorization using the simple characterization given in Lemma II.4.

Factoring General Sparse Polynomials: In order to the extend the above algorithm that works in the monic case to the more general case of non-monic polynomials, we use a standard reduction that transforms a general polynomial $f \in \mathbb{F}[x_1, x_2, \ldots, x_n]$ into a monic one $\hat{f}$.

Organization of Paper: In the next section, we recall some algebraic tools and algebraic algorithms that will be useful for us. In Section III, we discuss properties of polytopes and their relation to factor sparsity. Section IV contains the proof of the sparsity bound along with a discussion on its tightness. We present and analyze the deterministic factoring algorithm in Section V. We conclude with some open questions in Section VI. Due

to space constraints, several proofs are omitted.

II. Preliminaries

A. Algebraic Tool Kit

Let $\mathbb{F}$ denote a field, finite or otherwise, and let $\overline{\mathbb{F}}$ denote its algebraic closure.

A polynomial $f \in \mathbb{F}[x_1, x_2, \ldots, x_n]$ *depends* on a variable x_i if there are two inputs $\overline{\alpha}, \overline{\beta} \in \overline{\mathbb{F}}^n$ differing only in the i^{th} coordinate for which $f(\overline{\alpha}) \neq f(\overline{\beta})$. We denote by $\mathrm{var}(f)$ the set of variables that f depends on.

For a polynomial $f(x_1, \ldots, x_n)$, a variable x_i and a field element α, we denote with $f|_{x_i=\alpha}$ the polynomial resulting from substituting α to x_i. Similarly given a subset $I \subseteq [n]$ and an assignment $\overline{a} \in \mathbb{F}^n$, we define $f|_{\overline{x}_I=\overline{a}_I}$ to be the polynomial resulting from substituting a_i to x_i for every $i \in I$.

Definition II.1 (Line)**.** *Given $\overline{a}, \overline{b} \in \mathbb{F}^n$ we define a* line *passing through $\overline{a}$ and $\overline{b}$ as $\ell_{\overline{a},\overline{b}} : \mathbb{F} \to \mathbb{F}^n$, $\ell_{\overline{a},\overline{b}}(t) \triangleq (1-t)\cdot\overline{a}+t\cdot\overline{b}$. In particular, $\ell_{\overline{a},\overline{b}}(0) = \overline{a}$ and $\ell_{\overline{a},\overline{b}}(1) = \overline{b}$.*

Definition II.2 (Degrees, Leading Coefficients)**.** *Let $x_i \in \mathrm{var}(f)$. We can write: $f = \sum_{j=0}^{d} f_j \cdot x_i^j$ such that $\forall j : x_i \notin \mathrm{var}(f_j)$ and $f_d \not\equiv 0$. The* leading coefficient *of f w.r.t to x_i is defined as $\mathrm{lc}_{x_i}(f) \triangleq f_d$. The* individual degree *of x_i in f is defined as $\deg_{x_i}(f) \triangleq d$. We say that f is* monic *in a variable x_i if $\mathrm{lc}_{x_i}(f) = 1$. We say that f is monic if it is monic in some variable.*

It easy to see that for every $f, g \in \mathbb{F}[x_1, x_2, \ldots, x_n]$ and $i \in [n]$ it holds: $\mathrm{lc}_{x_i}(f \cdot g) = \mathrm{lc}_{x_i}(f) \cdot \mathrm{lc}_{x_i}(g)$.

B. Factors and Divisibility

Let $f, g \in \mathbb{F}[x_1, x_2, \ldots, x_n]$ be polynomials. We say that *g divides f*, or equivalently g is a factor of f, and denote it by $g \mid f$ if there exists a polynomial $h \in \mathbb{F}[x_1, x_2, \ldots, x_n]$ such that $f = g \cdot h$. We say that f is *irreducible* if f is non-constant and cannot be written as a product of two non-constant polynomials.

Given the notion of divisibility, we define the gcd of a set of polynomials in the natural way: we define it to be the highest degree polynomial dividing them all (suitably scaled). Given the notion of irreducibility we can state the important property of the uniqueness of factorization.

Lemma II.3 (Uniqueness of Factorization)**.** *Let $h_1^{e_1} \cdot \ldots \cdot h_k^{e_k} = g_1^{e'_1} \cdot \ldots \cdot g_{k'}^{e'_{k'}}$ be two factorizations of the same non-zero polynomial into irreducible, pairwise coprime factors. Then $k = k'$ and there exists a permutation $\sigma : [k] \to [k]$ such that $h_i \sim g_{\sigma(i)}$ and $e_i = e'_{\sigma(i)}$ for $i \in [k]$.*

Suppose that f is monic in x_i. It is easy to see f can be written as a product of monic factors. Therefore, we can specialize Lemma II.3 to consider the *unique monic factorization* of f as: $f = h_1^{e_1} \cdot \ldots \cdot h_k^{e_k}$ where h_i-s are irreducible, monic, pairwise coprime factors.

The following lemma provides a characterization of all irreducible, pairwise coprime factorizations of any polynomial.

Lemma II.4. *Consider the function $\Phi : \mathbb{N}^* \to \mathbb{N}$: given $\overline{e} = (e_1, \ldots, e_k)$, $\Phi(\overline{e}) \triangleq 2 \cdot \sum_{i=1}^{k} e_i - k$. Let $f \in \mathbb{F}[x_1, x_2, \ldots, x_n]$ be a polynomial and let $f = h_1^{e_1} \cdot \ldots \cdot h_k^{e_k}$ a factorization of f (not necessarily irreducible or coprime), where h_i-s are non-constant and $e_i \geq 1$. Then all irreducible, pairwise coprime factorizations of f correspond to those that maximize $\Phi(\overline{e})$.*

C. Sparse Polynomials

In this section we discuss sparse polynomials, their properties and some related efficient algorithms which leverage these properties.

An *s-sparse polynomial* is polynomial with at most s (non-zero) monomials. We denote by $\|f\|$ the *sparsity* of f. In this section we list several results related to sparse polynomials. We begin with an efficient reconstruction algorithm for sparse polynomials.

Lemma II.5 ([17])**.** *Let $n, s, d \in \mathbb{N}$. There exists a deterministic algorithm that given n, s, d and an oracle access to an s-sparse polynomial $f \in \mathbb{F}[x_1, x_2, \ldots, x_n]$ of degree d, uses $\mathrm{poly}(n, s, d, \log|\mathbb{F}|)$ field operations and outputs f (in its monomial representation).*

Notice that, the above lemma also shows the existence of an efficient hitting set for sparse polynomials.

As another simple corollary of [17], we obtain an efficient algorithm for sparse polynomial division, given an upper bound on the sparsity of the quotient polynomial. In other words, if f, g are sparse polynomials such that $f = g \cdot h$, then given black-box access to f and g, one can recover h (as long as it is also sparse). This is because given black-box access to f and g, one can simulate black-box access to h. One can then use [17] to interpolate and recover h. If h ends up being not sparse, then this algorithm would just reject. Moreover, given a candidate sparse polynomial h, it is easy to verify whether it is indeed the quotient polynomial of f and

g, but just multiplying out $h \cdot g$ and comparing with f.

Lemma II.6 (Corollary of [17]). *Let $n, s, d, t \in \mathbb{N}$. Let $f, g \in \mathbb{F}[x_1, x_2, \ldots, x_n]$ be s-sparse polynomials of degree at most d. Then there exists an algorithm that given f, g and t uses $\mathrm{poly}(n, d, s, t)$ field operations and computes the quotient polynomial of f and g, if it is a t-sparse polynomial. That is, if $f = gh$ for some $h \in \mathbb{F}[x_1, x_2, \ldots, x_n]$, $\|h\| \leq t$, then the algorithm outputs h. Otherwise, the algorithm rejects.*

D. GCD and Resultants

Let $f = a_d y^d + a_{d-1} y^{d-1} + \cdots + a_0$ and $g = b_e y^e + b_{e-1} y^{e-1} + \cdots + b_0$ be polynomials of y-degree exactly d and e, respectively. Resultant is the determinant of Sylvester matrix of order $d + e$. This representation of resultant ensures that if f and g are sparse polynomials in $\mathbb{F}[x_1, x_2, \ldots, x_n]$ with small individual degree (in y), then sparsity of $\mathrm{Res}_y(f, g)$ is bounded. We will use the following properties of resultant (For more info, see [18, Chap. 7])

Lemma II.7 (Resultant Properties). *Let $f, g \in \mathbb{F}[y, x_1, x_2, \ldots, x_n]$ be monic in y, s-sparse polynomial with individual degrees at most d. Then:*

1) *$\mathrm{Res}_y(f, g)(\overline{x})$ is an $(2ds)^{2d}$-sparse polynomial over $\mathbb{F}[x_1, x_2, \ldots, x_n]$ with individual degrees at most $2d^2$.*
2) *For every $\overline{a} \in \mathbb{F}^n$: $\mathrm{Res}_y\left(f|_{\overline{x}=\overline{a}}, g|_{\overline{x}=\overline{a}}\right) = \mathrm{Res}_y(f, g)(\overline{a})$.*
3) *$\gcd(f, g) \neq 1$ iff $\mathrm{Res}_y(f, g) \equiv 0$.*

Definition II.8. *For a field $\mathbb{F}$ we denote by $\mathrm{c}_{\mathbb{F}}(d)$ the time of the best known algorithm that factors a univariate polynomial of degree d over $\mathbb{F}$.*

Lemma II.9 (Univariate factoring). *Let $f(x) \in \mathbb{F}[x]$ be a univariate polynomial of degree d then by the well known algorithms of Lenstra-Lenstra-Lovasz [19] and Berlekamp [20], f can be factorized in time $\mathrm{c}_{\mathbb{F}}(d)$. where:*

1) *$\mathrm{c}_{\mathbb{F}}(d) = \mathrm{poly}(\ell \cdot p, d)$, if $\mathbb{F} = \mathbb{F}_{p^\ell}$.*
2) *$\mathrm{c}_{\mathbb{F}}(d) = \mathrm{poly}(d, t)$, where t is maximum bit-complexity of the coefficients of f, if $\mathbb{F} = \mathbb{Q}$.*

III. Polytopes and Polynomials

In this section we will discuss various properties of polytopes, in particular the Newton polytope. These will be crucial ingredients in our proof of the sparsity bound for factors of sparse polynomials. The main results that we will discuss and develop are:

1) If f, g, h are polynomials such that $f = g \cdot h$ then the sparsity of f is lower bounded by $\max\{|V(P_g)|, |V(P_h)|\}$, where P_g and P_h are the Newton polytopes of g and h respectively, and where for a polytope P, $V(P)$ denotes the set of vertices of P.
2) The convex hull of any subset of $\{0, 1, \ldots, d\}^n$ must have "many" vertices (i.e. corner points). We will prove this as a corollary of an approximate version of Carathéodory's theorem.

Our approach to bounding the sparsity of factors of a polynomial using the theory of polytopes, and in particular Item 1 (as stated above) was inspired by a connection of the theory of polytopes to sparsity bounds that was observed by Dvir and Oliveira [14].

For a finite set of points $v_1, v_2, \ldots, v_k \in \mathbb{R}^n$, their *convex span*, which we denote by $CS(v_1, \ldots, v_k)$ is the set defined by

$$CS(v_1, \ldots, v_k) = \left\{ \sum_{i=1}^{k} \lambda_i v_i \;\middle|\; \lambda_i \geq 0,\ \sum_{i=1}^{k} \lambda_i = 1 \right\}.$$

A set $P \subseteq \mathbb{R}^n$ is a called a *polytope* if there is a finite set of points $v_1, v_2, \ldots, v_k \in \mathbb{R}^n$ such that $P = CS(v_1, v_2, \ldots, v_k)$. For a polytope P, and a point $a \in P$, we say that a is a *vertex* of P if it **cannot** be written as $a = \lambda u + (1 - \lambda)v$ for any $u, v \in P \setminus \{a\}$ and $\lambda \in [0, 1]$. Alternatively, a vertex of P is face of dimension 0. We let $V(P)$ denote the set of vertices of P.

It is an easy to verify, and a basic fact about polytopes, that if P is a polytope, then $P = CS(V(P))$. Moreover, is $P = CS(v_1, v_2, \ldots, v_k)$ then $V(P) \subseteq \{v_1, v_2, \ldots, v_k\}$.

(For more details see [21] Propositions 2.2 and 2.3)

A. The Newton Polytope and Minkowski Sum

Definition III.1. *Given two polytopes P_1 and P_2 in $\mathbb{R}^n$, we define their* Minkowski Sum $P_1 + P_2$ *to be the set of points given by*

$$P_1 + P_2 = \{v_1 + v_2 \mid v_1 \in P_1 \text{ and } v_2 \in P_2\}.$$

The following is a classic fact about the Minkowski sum of two polytopes. It basically says that the Minkowski sum of two polytopes is itself a polytope, and the number of vertices of each of the original polytopes is a lower bound on the number of vertices of the Minkowski sum. See [14] (Theorem 3.12, Corollary 3.13), and [22] for the formal details of a proof. For

an informal proof sketch see the full version of this paper [23][Prop. 3.2].

Proposition III.2. *Let P_1 and P_2 be polytopes in $\mathbb{R}^n$. Then their Minkowski sum $P_1 + P_2$ is a polytope and*

$$|V(P_1 + P_2)| \geq \max\{|V(P_1)|, |V(P_2)|\}.$$

For a polynomial $f \in \mathbb{F}[x_1, x_2, \ldots, x_n]$, suppose that

$$f = \sum a_{i_1 i_2 \ldots i_n} x_1^{i_1} x_2^{i_2} \cdots x_n^{i_n}.$$

For each coefficient $a_{i_1 i_2 \ldots i_n} \neq 0$, we say that the exponent vector $(i_1, i_2, \ldots, i_n)$ is in the *support* of f, when viewed as a vector in $\mathbb{R}^n$. We define $\mathrm{Supp}(f)$ to be the set of all support vectors of f, i.e.

$$\mathrm{Supp}(f) = \{(i_1, i_2, \ldots, i_n) \mid a_{i_1 i_2 \ldots i_n} \neq 0\}.$$

The convex hull of the set $\mathrm{Supp}(f)$ is defined to be the *Newton polytope* of f, which we denote by P_f.

The following classic fact was observed by Ostrowski [16] in 1921. It states that if a polynomial f factors as $g \cdot h$, then the Newton polytope of f is the Minkowski sum of the Newton polytopes of g and h. (See also [14] (Proposition 3.16) for a proof.)

Proposition III.3. *Let $f, g, h \in \mathbb{F}[x_1, x_2, \ldots, x_n]$ be polynomials such that $f = g \cdot h$. Then*

$$P_f = P_g + P_h.$$

As an immediate corollary of the above two propositions, we easily recover the following basic bound relating the sparsity of polynomials to the Newton polytopes of its factors. (This bound was observed by Dvir and Oliveira in [14]).

Corollary III.4. *Let $f, g, h \in \mathbb{F}[x_1, x_2, \ldots, x_n]$ be polynomials such that $f = g \cdot h$. Then*

$$\|f\| \geq |V(P_f)| \geq \max\{|V(P_g)|, |V(P_h)|\}.$$

Proof: By Proposition III.3, $P_f = P_g + P_h$, and hence by Proposition III.2,

$$|V(P_f)| \geq \max\{|V(P_g)|, |V(P_h)|\}.$$

Since $P_f = CS(\mathrm{Supp}(f))$, thus $V(P_f) \subseteq \mathrm{Supp}(f)$. Hence $|V(P_f)| \leq \|f\|$ and the result follows. ∎

It is worth noting that if $d = 1$, that is when g (or h) is multilinear, then every point in P_g is a corner point. Hence, $|V(P_g)| = \|g\|$ and by Prop. III.2 $\|f\| \geq \|g\|$.

B. An approximate Carathéodory's Theorem

Carathéodory's theorem is a fundamental result in convex geometry, and it states that if a point $\mu \in \mathbb{R}^n$ lies in the convex hull of a set U, then μ can be written as the convex combination of at most $n + 1$ points of U.

In order to prove our sparsity bound, we will be using an "approximate" version of Carathéodory's theorem. The version that we use appears in [24][1]. It essentially states that if a point $\mu \in \mathbb{R}^n$ lies in the convex hull of a set U, then μ can be *uniformly* ε-approximated in the ℓ_∞ norm by a vector that is the convex combination of only $\frac{\log n}{\varepsilon^2}$ points of U.

We first introduce some notation that we will use. For a set of vectors $U = \{u_1, u_2, \ldots, u_m\} \subseteq \mathbb{R}^n$, let $CS(U)$ denote the convex hull of U. (Note that for a finite set, the convex span of a set of vectors is the same as the convex hull of the vectors. Since in the rest of the paper we will only be dealing with finite sets, we will use the terms convex span and convex hull interchangeably). A vector $\mu \in CS(U)$ is defined to be k-uniform with respect to U if there exists a multiset S of $[m]$ of size at most k such that $\mu = \frac{1}{k}\sum_{i \in S} u_i$.

We now state the approximate Carathéodory theorem for completeness this version below is for the ℓ_∞ norm, and its proof is fairly straightforward.

Theorem III.5 ([24], Theorem 3)**.** *Given a set of vectors $U = \{u_1, u_2, \ldots, u_m\} \subseteq \mathbb{R}^n$ with $\max_{u \in U} \|u\|_\infty \leq 1$, and $\varepsilon > 0$. For every $\mu \in CS(U)$ there exists an $\mathcal{O}\left(\frac{\log n}{\varepsilon^2}\right)$ uniform vector $\mu' \in CS(U)$ such that $\|\mu - \mu'\|_\infty \leq \varepsilon$.*

IV. Sparsity Bound

In this section we prove the sparsity bound. We will first show how to apply the approximate Carathéordory's Theorem III.5 to show that the convex hull of any subset of $\{0, 1, \ldots, d\}^n$ must have *many* vertices (i.e. corner points).

Theorem IV.1. *Let $E \subseteq \{0, 1, \ldots, d\}^n$. Let $t = |V(CS(E))|$. Then there exists an absolute constant C such that Then $t^{Cd^2 \log n} \geq |E|$.*

Remark IV.2. *In fact, the dependence on $\log n$ in the theorem above is necessary. In particular, there is a set $E \subseteq \{-1, 0, 1\}^n$ such that the number of corner points in the convex hull of E is n, but $|E| = n^{\Omega(\log n)}$. However, it is not clear if such polytopes yield a*

[1]There is actually a small typo in the version of the theorem in [24], and the statement below fixes it.

polynomial with $\xi(n,s,d) = s^{\Theta(d^2 \log n)}$. An example of such a set (and the resulting polytope) was shared with us in [25], for details see full version [23][Claim 4.4].

Theorem IV.3 (The Bound of Factor Sparsity)**.** *There exists an non-decreasing function $\xi(n,s,d) \leq s^{\mathcal{O}(d^2 \log n)}$ such that if $f \in \mathbb{F}[x_1, x_2, \ldots, x_n]$ is a polynomial of sparsity s and individual degrees at most d, and if $f = g \cdot h$, for $g, h \in \mathbb{F}[x_1, x_2, \ldots, x_n]$, then the sparsity of g is upper bounded by $\xi(n,s,d)$.*

Proof: Let $\|g\|$ denote the sparsity of g. Thus g has $\|g\|$ monomials. Let $\mathrm{Supp}(f), \mathrm{Supp}(g) \subseteq \{0,1,\ldots,d\}^n$ denote the sets of exponent vectors of f and g, respectively.

Let $t_g = |V(CS(\mathrm{Supp}(g)))|$. Thus t_g denotes the number of vertices of the polytope which is the convex span of $\mathrm{Supp}(g)$. Similarly let $t_f = |V(CS(\mathrm{Supp}(f)))|$. By Theorem IV.1,

$$t_g \geq \|g\|^{\frac{1}{Cd^2 \log n}}.$$

Now, by Corollary III.4, $t_g \leq t_f$. Moreover, since $V(CS(\mathrm{Supp}(f))) \subseteq E_f$, thus $t_f \leq |E_f|$, which equals the sparsity of f, $\|f\|$. Hence

$$\|f\| \geq \|g\|^{\frac{1}{Cd^2 \log n}}$$

and the theorem follows.

■

V. Factoring Algorithm

In this section, we give our deterministic factorization algorithm for sparse polynomials with small individual degree, thus proving Theorem 2. The runtime of the algorithm strongly depends on the bound in Theorem IV.3. To emphasize this dependence, we state our results in terms of $\xi(n,s,d)$. Theorem 1 follows by instating the upper bound.

As outlined in Section I-C2, we first focus on monic polynomials. Then we show how to extend the algorithm to general polynomials. Due to space constraints, all proofs are omitted. One can refer to the full version [23][Section 5] for them.

A. Black-box Factoring of Sparse Monic Polynomials (given some advice)

In this section we give an algorithm that takes as input a sparse monic polynomial $f(y, \overline{x})$ of bounded individual degree, as well as some additional information about its factorization pattern, and then outputs (in some sense) blackbox access to its factors.

The algorithm mirrors that black-box factorization algorithm of [9].

The algorithm assumes that it is given an assignment $\overline{a} \in \mathbb{F}^n$ for which no two distinct coprime factors of $f(y, \overline{x})$ have non-trivial gcd, when we set $\overline{x} = \overline{a}$, and it is given the correct partition of the factors of $f(y, \overline{a})$ (i.e. the partition gives the grouping of the factors of $f(y, \overline{a})$ that will correspond to the factors of f). The algorithm outputs evaluations of the irreducible factors of f at any input $(y_0, \overline{b}) \in \mathbb{F}^{n+1}$. More precisely, for any $\overline{b} \in \mathbb{F}^n$, and any irreducible factor $h_i(y, \overline{x})$ of f, the algorithm will output the univariate polynomial $h_i(y, \overline{b})$ which can then be evaluated at y_0.

Given an input point $\overline{b} \in \mathbb{F}^n$, the algorithm uses $\overline{a}$ as an anchor point and draws a line to $\overline{b}$. Next, the algorithm computes a bi-variate factorization of the polynomial $f(y, \ell_{\overline{a},\overline{b}}(t))$ (see Definition II.1). Finally, the algorithm outputs the black-boxes for each factor of f by matching the factors of $f(y, \ell_{\overline{a},\overline{b}}(t))$ to the factors of $f(y, \overline{a})$. We will describe our black-box factoring algorithm below:

Input: s-sparse monic (in y) polynomial $f \in \mathbb{F}[y, x_1, x_2, \ldots, x_n]$ of individual degree at most d.
Assignments: $\overline{a}, \overline{b} \in \mathbb{F}^n$
Univariate Polynomials: $g_1(y), g_2(y), \ldots, g_r(y)$
Partition: $A_1 \bigcup A_2 \bigcup \cdots \bigcup A_m = [r]$
Exponent Vector: $\overline{e} = (e_1, e_2, \ldots, e_m) \in [d]^m$
Output: Univariate Polynomials: $\varphi_1(y), \varphi_2(y), \ldots, \varphi_m(y)$

$\tilde{f}(y,t) \leftarrow f(y, \ell_{\overline{a},\overline{b}}(t))$;
Compute the bi-variate factorization of $\tilde{f}(y,t) = f_1^{v_1}(y,t) \cdot f_2^{v_2}(y,t) \cdots f_{r'}^{v_{r'}}(y,t)$;
for $i \leftarrow 1$ **to** m **do**
  $\tilde{h}_i(y,t) \leftarrow 1$;
  for $k \leftarrow 1$ **to** r' **do**
    if *there exists* $j \in A_i$ *s.t* $g_j(y) \mid f_k(y,0)$ **then**
      $\tilde{h}_i(y,t) \leftarrow \tilde{h}_i(y,t) \cdot f_k^{v_k/e_i}(y,t)$;
  end
end
return $\tilde{h}_1(y,1), \tilde{h}_2(y,1), \ldots, \tilde{h}_m(y,1)$;

Algorithm 1: Black-Box Evaluation of Factors

Lemma V.1 (Black-box Factorization)**.** *Let $f(y, \overline{x}) \in \mathbb{F}[y, x_1, x_2, \ldots, x_n]$ be a polynomial monic in y with individual degrees at most d. Suppose $f(y, \overline{x})$ can be writ-*

ten as $f(y,\overline{x}) = \prod_{i=1}^{m} h_i^{e_i}(y,\overline{x})$ *such that* $\gcd(h_i, h_{i'}) = 1$ *for* $i \neq i'$. *Then given:*

1) *a point* $\overline{a} \in \mathbb{F}^n$ *such that* $\forall i \neq i' : \mathrm{Res}_y(h_i, h_{i'})(\overline{a}) \neq 0$
2) *monic irreducible polynomials* $g_1(y), \ldots, g_r(y)$
3) *a partition* $A_1 \dot{\bigcup} A_2 \dot{\bigcup} \cdots \dot{\bigcup} A_m = [r]$ *such that for all* $i \in [m] : h_i(y,\overline{a}) = \prod_{j \in A_i} g_j(y)$
4) *exponent vector* $\overline{e} = (e_1, e_2, \ldots, e_m) \in [d]^m$

and a point $\overline{b} \in \mathbb{F}^n$, *Algorithm 1 computes* $h_1(y,\overline{b}), h_2(y,\overline{b}), \ldots, h_m(y,\overline{b})$, *using* $\mathrm{poly}(n, \mathrm{c}_{\mathbb{F}}(d))$ *field operations.*

B. Factoring Sparse Monic Polynomials (without advice)

With the black-box factoring algorithm of the previous subsection, we get blackbox access to the irreducible factors of the input monic sparse polynomial, and we can use a reconstruction algorithm to reconstruct the actual factors. The caveat is that black-box factorization algorithm of the previous section assumes that it is given some additional information: an assignment $\overline{a} \in \mathbb{F}^n$ for which no two distinct factors of $f(y,\overline{x})$ have non-trivial gcd, when we set $\overline{x} = \overline{a}$, and the correct partition of the factors of $f(y,\overline{a})$.

In this section we show that the advice is actually a member of a small set that can be computed, and hence one can just "guess" the advice! Since $f(y,\overline{a})$ has at most d factors, the number of possible partition is $d^{\mathcal{O}(d)}$. Hence we can "guess" the correct partition by trying out all the possibilities. In terms of finding $\overline{a}$ as above, the following lemma shows that there exists a small set of points $S \subseteq \mathbb{F}^n$ that contain a point $\overline{a}$ with the required properties for every monic sparse polynomial of degree d.

Lemma V.2. *Let* $f \in \mathbb{F}[y, x_1, x_2, \ldots, x_n]$ *be monic in* y, s*-sparse polynomial with individual degrees at most* d *and let* $f(y,\overline{x}) = h_1^{e_1}(y,\overline{x}) \ldots h_k^{e_k}(y,\overline{x})$ *be the unique monic factorization of* $f(y,\overline{x})$. *Then there exists a set* S *of size* $|S| = (n \cdot \xi(n,d,s))^{\mathcal{O}(d)}$ *such that for any* f *as above there exists an assignment* $\overline{a} \in \mathbb{F}^n$ *satisfying* $\forall i \neq i' : \mathrm{Res}_y(h_i, h_{i'})(\overline{a}) \neq 0$.

Given a polynomial f, the algorithm will proceeding by trying to reconstruct the factors of f for every projection in S and every partition. Given a purported factorization, we can test it by explicitly multiplying out the polynomials. Clearly, the algorithm will pick up *any* valid factorization of f (not just the irreducible one). We will select the irreducible factorization using the simple characterization given in Lemma II.4.

Input: s-sparse polynomial $f \in \mathbb{F}[y, x_1, x_2, \ldots, x_n]$, monic in y, of individual degree at most d.
Output: monic irreducible factors $h_1, h_2, \ldots, h_m$, and $e_1, e_2, \ldots, e_m$ such that $f = h_1^{e_1} \cdots h_m^{e_m}$

1) For each $\overline{a} \in S$ (from Lemma V.2), subset $I \subseteq [d]$, $m' \in [d]$, a non-empty *partition* of I: $A_1 \dot{\bigcup} A_2 \dot{\bigcup} \cdots \dot{\bigcup} A_{m'} = I$, and exponent vector $\overline{e}' = (e'_1, e'_2, \ldots, e'_{m'}) \in [d]^{m'}$:
   a) Compute the monic univariate factorization $f(y,\overline{a}) = \prod_{j=1}^{r} g_j(y)$ (Using Lemma II.9)
   b) Call Algorithm 1 with $f, \overline{a}, \{A_i\}_{i \in [m']}, \overline{e}$ and $\{g_j(y)\}_{j \in I}$.
   c) Invoke the reconstruction algorithm from Lemma II.5 with $n' = n, s' = \xi(n,d,s), d' = d$ using the above as an oracle to reconstruct the polynomials $h'_1(y,\overline{x}), \ldots h'_m(y,\overline{x})$.
   d) Test that $f \equiv {h'_1}^{e'_1} \cdot {h'_2}^{e'_2} \cdots {h'_{m'}}^{e'_{m'}}$ factorization.
2) Return a factorization that maximizes the expression $\Phi(\overline{e}) \triangleq 2 \cdot \sum_{i=1}^{m'} e'_i - m'$. `/* Pick the most ''refined'' factorization */`

Algorithm 2: Sparse Monic Polynomial Factorization

Lemma V.3. *Let* $f(y,\overline{x}) \in \mathbb{F}[y, x_1, x_2, \ldots, x_n]$ *be a polynomial, monic in* y, *with individual degrees at most* d. *Given* f, *Algorithm 2 computes the unique monic factorization of* f. *That is, the algorithm outputs coprime, monic irreducible polynomials* $h_1, h_2, \ldots, h_m$, *and* $e_1, e_2, \ldots, e_m$ *such that* $f = h_1^{e_1} \cdots h_m^{e_m}$, *using at most* $(n \cdot \xi(n,d,s))^{\mathcal{O}(d)} \cdot \mathrm{poly}(\mathrm{c}_{\mathbb{F}}(d))$ *field operations.*

C. Factoring General Sparse Polynomials

In this section we show how to extend the factorization algorithm for monic sparse polynomials to general sparse polynomials. We begin by showing how to convert a (general) sparse polynomial with "small" individual degrees into a "somewhat" sparse monic polynomial of a "slightly larger" individual degrees.

Definition V.4. *Let* $f(x_1, \ldots, x_n, x_{n+1}) \in \mathbb{F}[x_1, x_2, \ldots, x_{n+1}]$ *and let* $k \leq d$ *denote the degree of* x_{n+1} *in* f. *Let* $f_k \triangleq \mathrm{lc}_{x_{n+1}}(f)$. *We define:* $\hat{f}(y, x_1, \ldots, x_n) \triangleq f_k^{k-1} \cdot f(x_1, \ldots, x_n, \frac{y}{f_k})$.

Lemma V.5. *Suppose f is an s-sparse polynomial with individual degrees at most d. Then function $\hat{f}$ is an (s^d)-sparse polynomial in $\mathbb{F}[y, x_1, x_2, \ldots, x_n]$, monic in y with individual degrees at most d^2.*

In addition to the question regarding the sparsity of the polynomial $\hat{f}$, there are two follow-up questions we need to address:

1) How are the factors of $\hat{f}$ related the original factors of f?
2) As the degree of y in $\hat{f}$ is at most d, we can recover at most d factors, while f could potentially have dn factors! How can we recover the remaining factors?

The following lemma provides the answers to both questions.

Lemma V.6. *Let $f(\overline{x}, x_{n+1}) = \prod_{i=1}^{m'} h_i^{e_i}(\overline{x}, x_{n+1}) \cdot \prod_{l=m'+1}^{m} h_l^{e_l}(\overline{x})$ and $f_k(\overline{x}) = \prod_{j=1}^{r} w_j^{\beta_j}(\overline{x})$ be pair-wise coprime, irreducible factorizations of f and f_k, respectively such that $x_{n+1} \in \mathrm{var}(h_i)$ iff $i \in [m']$. Furthermore, let $\hat{f}(y, \overline{x}) = \prod_{j=1}^{\hat{m}} \hat{h}_j^{\hat{e}_j}(y, \overline{x})$ be the unique monic factorization of $\hat{f}$. Then*

1) *$\hat{m} = m'$ and there exist polynomials $u_1(\overline{x}), \ldots, u_{m'}(\overline{x}) \in \mathbb{F}[x_1, x_2, \ldots, x_n]$ and a permutation $\sigma : [m'] \to [m']$ such that: $\hat{h}_i(f_k \cdot x_{n+1}, \overline{x}) = h_{\sigma(i)}(\overline{x}, x_{n+1}) \cdot u_i(\overline{x})$ and $\hat{e}_i = e_{\sigma(i)}$ for $i \in [m']$.*
2) *$m - m' \leq r$. Moreover, there exists an injective map $\tau : \{m'+1, \ldots, m\} \to [r]$ such that h_l and $w_{\tau(l)}$ are nonzero scalar multiples of each other (i.e. $h_l \sim w_{\tau(l)}$), for $l \in \{m'+1, \ldots, m\}$.*

In light of the above, the algorithm proceeds by first converting a given polynomial f into a monic polynomial $\hat{f}$ to recover the factors that depend on x_{n+1}. Next, the algorithm recursively factors f_k (that does not depend on x_{n+1}) to recover the factors that *do not* depend on x_{n+1} (if any).

Theorem V.7. *Let $f(\overline{x}) \in \mathbb{F}[x_1, x_2, \ldots, x_n]$ be a polynomial with individual degrees at most d. Given f, Algorithm 3 outputs pairwise coprime, irreducible polynomials $h_1, h_2, \ldots, h_m$, and $e_1, e_2, \ldots, e_m$ such that $f = h_1^{e_1} \cdots h_m^{e_m}$, using $\left(n \cdot \xi(n, d^2, s^d)\right)^{\mathcal{O}(d^2)} \cdot \mathrm{poly}(\mathrm{c}_{\mathbb{F}}(d^2))$ field operations.*

VI. Open Questions

We conclude by listing some open problems.

Input: s-sparse polynomial $f \in \mathbb{F}[x_1, x_2, \ldots, x_{n+1}]$ with individual degrees at most d.
Output: irreducible factors $h_1, h_2, \ldots, h_m$, and $e_1, e_2, \ldots, e_m$ such that $f = h_1^{e_1} \cdots h_m^{e_m}$

1) **if** $n \leq 1$ **then** Return the bi-variate factorization of f;
2) $k = \deg_{x_{n+1}}(f)$; $f_k \leftarrow \mathrm{lc}_{x_{n+1}}(f)$;
3) Compute $\hat{f}(y, \overline{x})$ (Using Definition V.4)
4) Compute the unique monic factorization $\hat{f}(y, \overline{x}) = \prod_{i=1}^{\hat{m}} \hat{h}_i^{e_i}(y, \overline{x})$ (Using Algorithm 2)
5) **foreach** $i \in [\hat{m}]$ **do**
   $h_i(x_1, \ldots, x_n, x_{n+1}) \leftarrow \hat{h}_i(f_k \cdot x_{n+1}, \overline{x})$;
6) Recursively compute a factorization of $f_k(x_1, \ldots, x_n) = \prod_{j=1}^{r} w_j^{\beta_j}(x_1, \ldots, x_n)$
7) **for** $j \leftarrow 1$ **to** r **do**
   $\alpha_j \leftarrow -\beta_j \cdot (k-1)$;
   for $i \leftarrow 1$ **to** $\hat{m}$ **do**
      Find the maximal d_{ij} such that $w_j^{d_{ij}} \mid h_i$;
      `/* By iteratively applying Lemma II.6 with` $t = \xi(n, d^2, s^d)$ `*/`
      $\alpha_j \leftarrow \alpha_j + d_{ij} \cdot e_i$; $h_i \leftarrow h_i / w_j^{d_{ij}}$;
   end
   end
8) **return** $h_1, \ldots, h_{\hat{m}}, w_1, \ldots, w_r$ and $e_1, \ldots, e_{\hat{m}}, \alpha_1, \ldots, \alpha_r$; `/* Return only those where` $\alpha_j > 0$ `*/`

Algorithm 3: Main Algorithm: overview

Perhaps the most immediate and natural question left open by this work is to understand whether one can obtain an improved sparsity bound on the factors of s-sparse polynomials of bounded individual degree. The best lower bound for we know for the sparsity of factors of s-sparse polynomials of individual degree d is $s^{\log d}$ over fields of characteristic 0 and about s^d over general fields; see [23][Section 4.1]. Thus there is a considerable gap between these lower bounds and the upper bound that we prove, and it is a very interesting question to close to gap.

Another more ambitious goal is to obtain a non trivial sparsity bound with no restriction on individual degree. Such a result would not be possible for all fields, and any such proof would have to use the properties of the underlying field to obtain a better bound.

One could also study the algorithmic implications of a

general sparsity bound. It seems challenging to derandomize polynomial factoring, even if we assume that factors of a given sparse polynomial are sparse (without assuming any individual degree bound). We leave this as an interesting open problem.

Given the result of [10] which shows an equivalence between the problems of polynomial identity testing and polynomial factorization, this also naturally raises the question (and indeed it was raised in [10]) of whether one can derandomize factoring for the classes of polynomials for which we know how to derandomize PIT. Sparse polynomials are a natural example of such a class, but there are several other natural classes that one could consider.

ACKNOWLEDGMENT

The authors would like to thank Ramprasad Saptharishi [25] for sharing the tight example for Thm IV.1 with us. In addition, the authors would like to thank the anonymous referees for useful comments that improved the presentation of the results.

REFERENCES

[1] M. Sudan, "Decoding of reed solomon codes beyond the error-correction bound." *Journal of Complexity*, vol. 13(1), pp. 180–193, 1997.

[2] V. Guruswami and M. Sudan, "Improved decoding of reed-solomon codes and algebraic-geometry codes." *IEEE Transactions on Information Theory*, vol. 45(6), pp. 1757–1767, 1999.

[3] V. Kabanets and R. Impagliazzo, "Derandomizing polynomial identity tests means proving circuit lower bounds." *Computational Complexity*, vol. 13, no. 1-2, pp. 1–46, 2004.

[4] B. Chor and R. L. Rivest, "A knapsack-type public key cryptosystem based on arithmetic in finite fields," *IEEE Transactions on Information Theory*, vol. 34, no. 5, pp. 901–909, 1988.

[5] J. v. z. Gathen and J. Gerhard, *Modern computer algebra*. Cambridge University Press, 1999.

[6] J. v. z. Gathen and E. Kaltofen, "Factoring sparse multivariate polynomials," *Journal of Computer and System Sciences*, vol. 31, no. 2, pp. 265–287, 1985. [Online]. Available: http://dx.doi.org/10.1016/0022-0000(85)90044-3

[7] E. Kaltofen, "Single-factor hensel lifting and its application to the straight-line complexity of certain polynomials," in *Proceedings of the 19th Annual ACM Symposium on Theory of Computing (STOC)*, 1987, pp. 443–452. [Online]. Available: http://doi.acm.org/10.1145/28395.28443

[8] ——, "Factorization of polynomials given by straight-line programs," in *Randomness in Computation*, ser. Advances in Computing Research, S. Micali, Ed. JAI Press Inc., Greenwhich, Connecticut, 1989, vol. 5, pp. 375–412.

[9] E. Kaltofen and B. M. Trager, "Computing with polynomials given by black boxes for their evaluations: Greatest common divisors, factorization, separation of numerators and denominators," *J. of Symbolic Computation*, vol. 9, no. 3, pp. 301–320, 1990.

[10] S. Kopparty, S. Saraf, and A. Shpilka, "Equivalence of polynomial identity testing and deterministic multivariate polynomial factorization," in *Proceedings of the 29th Annual IEEE Conference on Computational Complexity (CCC)*, 2014, pp. 169–180.

[11] A. Shpilka and I. Volkovich, "On the relation between polynomial identity testing and finding variable disjoint factors," in *Automata, Languages and Programming, 37th International Colloquium (ICALP)*, 2010, pp. 408–419, full version at https://eccc.weizmann.ac.il/report/2010/036.

[12] I. Volkovich, "On some computations on sparse polynomials," in *APPROX-RANDOM*, 2017, pp. 48:1–4:21.

[13] R. M. de Oliveira, "Factors of low individual degree polynomials," in *Proceedings of the 30th Conference on Computational Complexity (CCC)*, 2015, pp. 198–216. [Online]. Available: https://doi.org/10.4230/LIPIcs.CCC.2015.198

[14] Z. Dvir and R. M. de Oliveira, "Factors of sparse polynomials are sparse," *CoRR*, vol. abs/1404.4834, 2014.

[15] I. Volkovich, "Deterministically factoring sparse polynomials into multilinear factors and sums of univariate polynomials," in *APPROX-RANDOM*, 2015, pp. 943–958.

[16] A. Ostrowski, " "U on the meaning of the theory of convex polyhedra for the formal algebra," *Annual Reports German Math. Association*, vol. 20, pp. 98–99, 1921.

[17] A. Klivans and D. Spielman, "Randomness efficient identity testing of multivariate polynomials," in *Proceedings of the 33rd Annual ACM Symposium on Theory of Computing (STOC)*, 2001, pp. 216–223.

[18] K. O. Geddes, S. R. Czapor, and G. Labahn, *Algorithms for computer algebra*. Kluwer, 1992.

[19] A. Lenstra, H. Lenstr, and L. Lovász, "Factoring polynomials with rational coefficients," *Mathematische Annalen,*, vol. 261, no. 4, pp. 515–534, 1982.

[20] E. Berlekamp, "Factoring polynomials over large finite fields," *Mathematics of Computation*, vol. 24, no. 111, pp. 713–335, 1970.

[21] G. M. Ziegler, *Lectures on polytopes*. Springer Science & Business Media, 2012, vol. 152.

[22] A. Schinzel, *Polynomials with special regard to reducibility*. Cambridge University Press, 2000, vol. 77.

[23] V. Bhargava, S. Saraf, and I. Volkovich, "Deterministic factorization of sparse polynomials with bounded individual degree," *Electronic Colloquium on Computational Complexity (ECCC)*. [Online]. Available: https://eccc.weizmann.ac.il/report/2018/130/

[24] S. Barman, "Approximating nash equilibria and dense bipartite subgraphs via an approximate version of caratheodory's theorem," in *Proceedings of the forty-seventh Annual ACM Symposium on Theory of Computing (STOC)*, 2015, pp. 361–369.

[25] R. Saptharishi, "Private communication," 2018.