On the Complexity of Factoring Bivariate Supersparse (Lacunary) Polynomials*

Erich Kaltofen
Dept. of Mathematics
North Carolina State University
Raleigh, North Carolina
27695-8205, USA
kaltofen@math.ncsu.edu

Pascal Koiran
Laboratoire LIP
École Normale Supérieure de Lyon
46, Allée d'Italie,
69364 Lyon Cedex 07, France
Pascal.Koiran@ens-lyon.fr

ABSTRACT

We present algorithms that compute the linear and quadratic factors of supersparse (lacunary) bivariate polynomials over the rational numbers in polynomial-time in the input size. In supersparse polynomials, the term degrees can have hundreds of digits as binary numbers. Our algorithms are Monte Carlo randomized for quadratic factors and deterministic for linear factors. Our approach relies on the results by H. W. Lenstra, Jr., on computing factors of univariate supersparse polynomials over the rational numbers. Furthermore, we show that the problem of determining the irreducibility of a supersparse bivariate polynomial over a large finite field of any characteristic is co-NP-hard via randomized reductions.

Categories and Subject Descriptors

I.1.2 [**Computing Methodologies**]: Symbolic and Algebraic Manipulation—*Algorithms*; F.1.2 [**Theory of Computation**]: Analysis of Algorithms and Problem Complexity—*Nonnumerical Algorithms and Problems*

General Terms

algorithms, theory

Keywords

sparse polynomials, lacunary polynomials, multivariate polynomials, polynomial factorizations, polynomial-time complexity, NP-hardness

1. INTRODUCTION

The algorithms in this paper take as inputs "super"sparse polynomials, which A. Schinzel and H. W. Lenstra, Jr., call *lacunary*† polynomials. A *supersparse* polynomial

$$f(X_1, \dots, X_n) = \sum_{i=1}^{t} a_i\, X_1^{\alpha_{i,1}} \cdots X_n^{\alpha_{i,n}}$$

is input by a list of its coefficients and corresponding term degree vectors. One defines the size of f as

$$\mathrm{size}(f) = \sum_{i=1}^{t} \Big(\mathrm{size}(a_i) + \lceil \log_2(\alpha_{i,1} \cdots \alpha_{i,n} + 2)\rceil\Big), \quad (1)$$

where $\mathrm{size}(a_i)$ is the bit-size of the scalar coefficients. One thus allows very high degrees, say with hundreds of digits as binary numbers, in distinction to the usual sparse representation [28, 16]. If the coefficients are integers, one cannot evaluate a supersparse polynomial at integer values in polynomial-time in its size, because the value of the polynomial can have exponential size, say 2^{100} digits. Important exceptions are evaluating at 0 or ± 1. A supersparse polynomial can be represented by a straight-line program [13] of size $O(\mathrm{size} f)$ via evaluating its terms with repeated squaring. It is NP-hard to test if two integral univariate supersparse polynomials have a non-trivial greatest common divisor [22].

A breakthrough polynomial-time result is in [3]. Any integral root of a univariate supersparse polynomial with integral coefficients can be found in $(\mathrm{size} f)^{O(1)}$ bit operations. H. W. Lenstra, Jr., [19, 20] has generalized the result to computing factors of fixed degree in an algebraic extension of fixed degree, in particular to computing rational roots in polynomial-time. Using interpolation and divisibility testing à la [1] in connection with Lenstra's algorithm, in section 3 we present an algorithm for computing linear and quadratic rational factors of integral bivariate ($n = 2$) supersparse polynomials in $(\mathrm{size} f)^{O(1)}$ bit operations. Our algorithm is randomized of the Monte Carlo kind, and in section 4 we show how the linear bivariate factors can be found deterministically.

Several hardness results for supersparse polynomials over finite fields have been derived from Plaisted's approach [7, 17]. For example, Plaisted's hardness of GCD $\neq 1$ extends to polynomials over $\mathbb{Z}_p$ [7] and can be used to show NP-hardness (via randomized reduction) of the irreducibility of supersparse bivariate polynomials for sufficiently large p (cf. [17, Proof of Theorem 1]). In section 5 we summarize those

*This material is based on work supported in part by the National Science Foundation under Grant No. CCR-0305314 (Kaltofen).

ISSAC'05, July 24–27, 2005, Beijing, China.

†A lacuna is a hole as in the word 'lake;' the polynomials have, so to speak, "lagoons of zero coefficients."

results and generalize them to finite fields of any characteristic.

For all problems that we consider there are deterministic and/or probabilistic algorithms whose bit complexity is of order $(\mathrm{size}(f) + \deg(f))^{O(1)}$ [14, 15]. We remark that our representation of the coefficients of f and the modulus p is by dense vectors of digits, not by supersparse lists of non-zero digits and their positions in the integers (cf. [25]).

We note that Barvinok's representation by short rational generating functions [2] is related to our supersparse representation, and short rational functions have been successfully employed to solve combinatorial counting problems [4].

2. THE RESULTS BY CUCKER ET AL. AND LENSTRA

In [3] it is shown how to compute an integer root of a supersparse polynomial $f(X) = a_1 + a_2X^{\alpha_2} + \cdots + a_tX^{\alpha_t} \in \mathbb{Z}[X]$ in polynomial time in the size of the polynomial. The result has a short proof based on finding gaps: suppose that $f(X) = g(X) + X^u h(X)$ with $g \neq 0$, $h \neq 0$, $\deg(g) \leq k$ and let $u - k \geq \delta = \log_2 \|f\|_1 = \log_2(|a_1| + \cdots + |a_t|)$. For an integer $a \neq \pm 1$, we have $f(a) = 0 \Longrightarrow g(a) = h(a) = 0$. Assume the contrary, namely that $a \neq 0, \pm 1$ and $h(a) \neq 0$. Then

$$|g(a)| < \|f\|_1 \cdot |a|^k \leq 2^{u-k} \cdot |a|^k \leq |a|^u \leq |a^u h(a)|, \quad (2)$$

thus $|f(a)| \geq |a^u h(a)| - |g(a)| > 0$. Note the similarity of (2) with the proof of Cauchy's root bound. The estimate for δ can be sharpened [3, Proposition 2]. The polynomial time algorithm can now proceed by computing the integer roots of those polynomial segments $a_iX^{\alpha_i} + \cdots + a_jX^{\alpha_j}$ in f whose terms have degree differences $\alpha_l - \alpha_{l-1} < \delta$, for all $i < l \leq j$. After dividing out X^{α_i}, we have polynomials of degree $\leq (t-1)(\delta - 1)$, whose common integer roots are found by p-adic lifting [21]. In section 4 we give a variant of the gap technique for high degree sums of linear forms.

H. W. Lenstra, Jr. has used the gap method to computing rational roots and low degree factors of supersparse rational polynomials via the height of an algebraic number (see section 4). The algorithm presented in [19] receives as input a supersparse polynomial $f(X) = \sum_{i=1}^{t} a_iX^{\alpha_i} \in K[X]$, where the algebraic number field K is represented as $K = \mathbb{Q}[\zeta]/(\varphi(\zeta))$ with a monic irreducible minimum polynomial $\varphi(\zeta) \in \mathbb{Z}[\zeta]$. Furthermore, a factor degree bound d is input. The algorithm produces a list of all irreducible factors of f over K of degree $\leq d$ and their multiplicities. Let $D = d \cdot \deg(\varphi)$. There are at most

$$O(t^2 \cdot 2^D \cdot D \cdot \log(2D\,t)) \quad (3)$$

irreducible factors of degree $\leq d$ [20, Theorem 1], each of which, with the exception of the possible factor X, has multiplicity at most t [19, Proposition 3.2]. The algorithm finishes in

$$\big(t + \log(\deg f) + \log\|f\| + \log\|\varphi\|\,\big)^{O(D)} \quad (4)$$

bit operations. Here $\|\varphi\|$ is the (infinity) norm of the coefficient vector of φ and $\|f\|$ is the norm of the vector of norms of the coefficients $a_i(\zeta)$. We assume that a common denominator has been multiplied through and all coefficients of the $a_i(\zeta)$ are integers. We note that by standard factor coefficient bound techniques [6], all factors have coefficients of size $\big(t + \log\|f\| + \log\|\varphi\|\,\big)^{O(D)}$, which is independent of $\deg(f)$.

For example, for $\varphi = \zeta - 1$, that is, $K = \mathbb{Q}$, and $d = 1 = D$, Lenstra's algorithm finds all rational roots of a supersparse integral polynomial f in polynomial-time in size(f).

3. LINEAR AND QUADRATIC FACTORS

We now present our randomized algorithm for computing linear and quadratic factors (and their multiplicities) of bivariate supersparse polynomials. For simplicity, we shall consider polynomials with rational coefficients only, although our method would allow coefficients in an algebraic number field. Our algorithm calls the univariate algorithm by H. W. Lenstra, Jr. [19]. For simplicity (see Remark 1 below), we assume that the input polynomial is monic in X.

Algorithm *Supersparse Factorization*
Input: a supersparse $f(X,Y) = \sum_{i=1}^{t} a_i\, X^{\alpha_i} Y^{\beta_i} \in \mathbb{Z}[X,Y]$ that is monic in X and an error probability $\epsilon = 1/2^l$.
Output: a list of polynomials $g_j(X,Y)$ with $\deg_X(g_j) \leq 2$ and $\deg_Y(g_j) \leq 2$ and corresponding multiplicities, which with probability no less than $1-\epsilon$ are all linear and quadratic irreducible factors of f over $\mathbb{Q}$ together with their true multiplicities.

Step 0. Factor out the maximum powers of X and Y that divide f. The non-zero coefficients of f do not change. Compute all linear and quadratic irreducible factors of f that are in $\mathbb{Q}[Y]$ by applying Lenstra's method to the coefficients of X^{α_i}. The multiplicities are also provided by Lenstra's algorithm.

Step 1. Compute all linear and quadratic irreducible factors in $\mathbb{Q}[X]$ of $f(X,0)$, $f(X,1)$ and $f(X,-1)$ by Lenstra's method. The algorithm will also provide the multiplicities of those factors.

Step 2. Interpolate all factor combinations.
Test if a factor candidate $g(X,Y)^\mu$ of candidate multiplicities μ divides $f(X,Y)$ by testing if $0 \equiv f(X,a) \bmod (g(X,a)^\mu, p)$ where $a \in S \subset \mathbb{Z}$, and $p \leq B$ a prime integer are randomly selected. The cardinality $|S|$ of S and the bound B are chosen in dependence of f and the input error probability ϵ (see below). The algorithm may fail to sample a prime $p \leq B$ and return "failure," which is interpreted as an incorrect answer in the output specification of the probability of correctness. □

We now show that our algorithm Supersparse Factorization can be implemented as to run in

$$\big(t + \log(\deg f) + \log\|f\| + \log 1/\epsilon\big)^{O(1)} \quad (5)$$

bit operations. Note that the measure (5) is polynomial in size(f) and $l = -\log\epsilon$.

By (3) in section 2 and our restriction to $D \leq 2$, the polynomials $f(X,0)$, $f(X,-1)$ and $f(X,1)$ each have no more than $O(t^2 \log t)$ linear or irreducible quadratic factors. In Step 2, one interpolates factors that are monic in X and whose coefficients have size $(t + \log\|f\|)^{O(1)}$. There are $O(t^4(\log t)^2)$ combinations of linear factors and $O(t^{12}(\log t)^6)$ combinations of quadratic factors, the latter because we must also consider products of univariate linear factors as images of bivariate quadratic factors. In practice, of course, the number of combinations can be much smaller. At least one of the univariate factors in each combination is $\neq X$ in the linear case and $\neq X^2$ in the quadratic case, because the interpolated bivariate factor cannot be X or X^2. Therefore the multiplicity m of one of the univariate factors satisfies $m \leq t$, and we need to check all $\mu \leq m$.

For each candidate factor $G(X,Y) = g(X,Y)^\mu$ we consider the division with remainder in X,

$$f(X,Y) - q(X,Y)G(X,Y) = h(X,Y), \quad \text{where } \deg_X(h) < \deg_X(G). \tag{6}$$

By considering (6) as a (unimodular) linear system over $\mathbb{Q}(Y)$ with $\deg_X(f)+1$ equations and variables, we obtain bounds for $\deg_Y(h)$ and $\|h\|$ [8]:

$$\deg_Y(h) \leq \deg_Y(f) + \deg_Y(G)\times (\deg_X(f) + 1 - \deg_X(G)) = O(t \deg(f)) \tag{7}$$

and

$$\|h\|_\infty^2 \leq t \cdot \|f\|_1^2 \times ((\deg_X(G)+1)\cdot \|G\|_1^2)^{\deg_X(f)+1-\deg_X(G)}. \tag{8}$$

From (7) and ϵ we derive a bound for $|S|$ in Step 2, and from (8) and ϵ a bound for B in Step 2. Suppose G does not divide f, that is there is a coefficient $h_i(Y) \neq 0$ of X^i in h.

First, we wish to have $0 \neq h_i(a)$ with probability $\geq 1-\eta/3$, where $\eta = \epsilon/A$ with $A = O(t^{13}(\log t)^6)$ being the number of factor combinations and multiplicities that have to be tested. The probability to pick a root of $h_i(Y)$ among the elements in $S \subset \mathbb{Z}$ is no more than $\deg_Y(h)/|S|$. By (7), for a set S of cardinality

$$|S| = (t + \deg f + 1/\epsilon)^{O(1)} \tag{9}$$

we can succeed with probability $\geq 1-\eta/3$. Let $H = h_i(a)$ for $a \in S$. We get by (9) and again by (7) and (8) that $H = (t + \deg f + \|f\| + 1/\epsilon)^{O(t \deg f)}$.

Second, we choose B such that $0 \not\equiv H \pmod p$ with probability $\geq 1-\eta/3$. By facts on the prime number distribution (see [23] for explicit estimates), there is a constant γ_1 such that H has at most $\gamma_1 \log H/\operatorname{loglog} H$ distinct prime factors. Since there are no fewer than $\gamma_2 B/\log B$ primes $\leq B$, the probability that $0 \equiv H \pmod p$ is no more than $\gamma_3(\log H/\operatorname{loglog} H)/(B/\log B)$ for some constants γ_2 and γ_3. Because one has

$$\gamma_3 \frac{\log H/\operatorname{loglog} H}{B/\log B} \leq \frac{\epsilon}{3\,A} \Longleftrightarrow \gamma_3 \frac{3A\log H}{\epsilon \operatorname{loglog} H} \leq \frac{B}{\log B},$$

one may choose

$$B = (A \cdot \log H \cdot 1/\epsilon)^{O(1)} \tag{10}$$

and achieve failure probability $\leq \eta/3$. Note that the number of digits in p is of order $(t+\log(\deg f)+\log\|f\|+\log 1/\epsilon)^{O(1)}$.

The algorithm must succeed to pick a prime $p \leq B$. By iterating the prime selection process $O(\log(A/\epsilon)\cdot \log B)$ times we can assume that to happen with probability $\geq 1-\eta/3$. Thus a single false factor combination is eliminated with probability $\geq (1-\eta/3)^3 \geq 1 - 1/\eta$. Therefore no false factor combination or multiplicity is accepted with probability $\geq (1-\eta)^A \geq 1 - A\eta \geq 1-\epsilon$.

The bit complexity measure (5) follows from the bounds (9) and (10) together with the repeated squaring algorithm and a polynomial primality test used in Step 2.

Remark 1. Our algorithm can be extended to compute in polynomial time all irreducible factors g_j with $\deg_X(g_j) = O(1)$, i.e., of constant degree in X, and simultaneously of $\deg_Y(g_j) \leq 2$. The input condition of monicity of f can be relaxed to accept polynomials with a leading coefficient (or trailing coefficient) in X that does not vanish for $Y=0$, $Y=-1$ or $Y=1$. One imposes a factor of the leading coefficient on the interpolated polynomials, which is a technique from sparse Hensel lifting [11]. One may also switch the roles of X and Y. However, at this time we do not know at all how to interpolate the factors of polynomials such as

$$\sum_i (X^{2d_i}-1)(Y^{2e_i}-1)f_i(X,Y) \tag{11}$$

where the f_i are supersparse. □

However, in the next section, we can show how to compute in deterministic polynomial time all factors of *total* degree 1 of *any* supersparse bivariate rational polynomial, including those of the form (11).

4. DETERMINISTIC LINEAR FACTORS

In this section we give a deterministic polynomial time algorithm that finds the linear factors of a supersparse polynomial. In contrast to the randomized algorithm of section 3, this deterministic algorithm can handle all (bivariate) supersparse polynomials. Our approach is based on the observation that a polynomial $g(X,Y)$ is divisible by $Y - bX - a$ iff $g(X, a+bX) = 0$. We will first give an algorithm that decides whether a polynomial of the form

$$\sum_{j=0}^{t} a_j X^{\alpha_j}(a+bX)^{\beta_j} \tag{12}$$

is identically equal to zero. Here a and b and the a_j are rational numbers; the α_j and β_j are non-negative integers. This algorithm can be used to check with certainty whether a "candidate factor" $Y - bX - a$ (for instance generated by an interpolation technique as in section 3) really is a factor of the bivariate polynomial $\sum_j a_j X^{\alpha_j} Y^{\beta_j}$. In general, deciding deterministically whether a straight-line program computes the identically zero polynomial is a notorious open problem. It turns out, however, that for polynomials of the form (12) this problem has an efficient solution. We will then see that this verification algorithm can be easily converted into an algorithm that actually finds all linear factors.

Even though our input polynomials have rational coefficients as in the remainder of the paper, the results of this section rely heavily on algebraic number theory.[‡] We review the necessary material in section 4.1. A suitable gap theorem is established in section 4.2. Here, some crucial ideas are borrowed from Lenstra's [19] paper. In particular, Proposition 1 closely follows Proposition 2.3 of [19]. Finally, our deterministic algorithms are presented in section 4.3.

4.1 Heights of algebraic numbers

In this section we quickly recall some number theoretic background. For any prime number p, the p-adic absolute value on $\mathbb{Q}$ is characterized by the following properties: $|p|_p = 1/p$, and $|q|_p = 1$ if q is a prime number different from p. For any $x \in \mathbb{Q}\setminus\{0\}$, $|x|_p$ can be computed as follows:

[‡]It is an interesting open problem whether they have more elementary proofs such as the one given in section 2.

write $x = p^\alpha y$ where p is relatively prime to the numerator and denominator of y, and $\alpha \in \mathbb{Z}$. Then $|x|_p = 1/p^\alpha$ (and of course $|0|_p = 0$). We denote by $M_\mathbb{Q}$ the union of the set of p-adic absolute values and of the usual (archimedean) absolute value on $\mathbb{Q}$.

Let $d, e \in \mathbb{Z}$ be two non-zero relatively prime integers. By definition, the height of the rational number d/e is $\max(|d|, |e|)$. There is an equivalent definition in terms of absolute values: for $x \in \mathbb{Q}$, $H(x) = \prod_{\nu \in M_\mathbb{Q}} \max(1, |x|_\nu)$. Note in particular that $H(0) = 1$.

More generally, let K be a number field (an extension of $\mathbb{Q}$ of finite degree). The set M_K of *normalized absolute values* is the set of absolute values on K which extend an absolute value of $M_\mathbb{Q}$. For $\nu \in M_K$, we write $\nu|\infty$ if ν extends the usual absolute value, and $\nu|p$ if ν extends the p-adic absolute value. One defines a "relative height" H_K on K by the formula

$$H_K(x) = \prod_{\nu \in M_K} \max(1, |x|_\nu)^{d_\nu}. \tag{13}$$

Here d_ν is the so-called "local degree". For every p (either prime or infinite), $\sum_{\nu|p} d_\nu = [K : \mathbb{Q}]$. Sometimes, instead of (13) one just writes $H_K(x) = \prod_\nu \max(1, |x|_\nu)$ if it is understood that each absolute value may occur several times (in fact, d_ν times) in the product. The absolute height $H(x)$ of x is $H_K(x)^{1/n}$, where $n = [K : \mathbb{Q}]$. It is independent of the choice of K.

There is a nice connection between the height of algebraic numbers and the Mahler measure of polynomials. Recall that the Mahler measure $M(f)$ of a polynomial of degree d:

$$f(X) = a_d X^d + \cdots + a_1 X + a_0 = a_d \prod_{i=1}^{d} (X - \alpha_i)$$

is by definition equal to $|a_d| \prod_{i=1}^d \max(1, |\alpha_i|)$. It turns out that if α is an algebraic number of degree d and $f \in \mathbb{Z}[X]$ its minimal polynomial, $M(f) = H(\alpha)^d$ ([27], section 3.3). This connection is often helpful when one has to estimate heights, but here we will use directly the definition of height in terms of absolute values. In Proposition 1 we will also use the product formula:

$$\prod_{\nu \in M_K} |x|_\nu^{d_\nu} = 1 \tag{14}$$

for any $x \in K \setminus \{0\}$. More details on absolute values and height functions can be found for instance in [18] or [27].

4.2 A gap theorem

We define a notion of height for an expression of the form (12) by the formula

$$H(f) = \prod_{\nu \in M_\mathbb{Q}} |f|_\nu,$$

where $|f|_\nu = \max_{0 \le j \le t} |a_j|_\nu$. There is a classical notion of height for a point in projective space ([10], section B.2) and in fact $H(f)$ is simply the height of the point $(a_0, a_1, \ldots, a_t)$. A nice feature of $H(f)$ is its invariance by scalar multiplication: if $\lambda \in \mathbb{Q} \setminus \{0\}$, $H(\lambda f) = H(f)$. Indeed, if we multiply a polynomial by p^α where p is prime, the archimedean absolute value is multiplied by p^α and the p-adic absolute value is divided by p^α. The other absolute values are unchanged. Note also that $H(f) = \max_j |a_j|$ if the a_j are relatively prime integers. Computing $H(f)$ in the general case $a_j \in \mathbb{Q}$ is therefore quite easy: reduce to the same denominator to obtain integer coefficients, divide by their gcd and take the maximum of the absolute values of the resulting integers (so in particular $H(f) \in \mathbb{Z}_{>0}$ for any f). Finally, a word of caution: our notion of height is not intrinsic to the given polynomial in X, since it is not invariant of the particular *representation* (12). Given a bivariate polynomial $g(X, Y)$ one could, however, define an intrinsic height $H(G)$ as done above (i.e., as the projective height of its tuple of coefficients), and we would have $H(f(X, a + bX)) = H(G)$.

THEOREM 1. *Let $f(X)$ be a polynomial of the form (12) where (a, b) is a pair of rational numbers different from the five pairs $(0,0)$, $(\pm 1, 0)$, $(0, \pm 1)$. Assume without loss of generality that the sequence (β_j) is nondecreasing, and assume also that there exists l such that*

$$\beta_{l+1} - \beta_l > \log(t\, H(f)) / \log \kappa,$$

where $\kappa > 1$ is an absolute constant defined in Lemma 2. If f is identically zero, the polynomials $g = \sum_{j=0}^{l} a_j X^{\alpha_j} (a + bX)^{\beta_j}$ and $h = \sum_{j=l+1}^{t} a_j X^{\alpha_j} (a + bX)^{\beta_j}$ are both identically zero.

PROOF. Let $U(a, b)$ be the set of roots of unity defined in Lemma 2 below. By hypothesis, $f(\theta) = 0$ for each $\theta \in U(a, b)$. By Proposition 1 below, g and h are both identically zero on $U(a, b)$. The result follows since $U(a, b)$ is an infinite set. □

We denote by $\mathcal{U}$ the set of complex roots of unity of prime order, and by $\mathcal{U}_{\ge 5}$ the set of complex roots of unity of prime order ≥ 5.

LEMMA 1. *There is an absolute constant $\kappa_1 > 1.045$ such that the following holds. For any $\theta \in \mathcal{U}_{\ge 5}$, if $a \in \mathbb{Z} \setminus \{0\}$ and $b \in \mathbb{Z} \setminus \{0\}$ then $H(a + b\theta) \ge \kappa_1$.*

REMARK 2. The hypothesis that θ is of order at least 5 is necessary. Indeed, if θ is of order 3 then $H(1 + \theta) = 1$ since $1 + \theta = -\theta^2$. Moreover, the restriction to roots of prime order can probably be removed with some additional work. □

PROOF OF LEMMA 1. Note that $|a + b\theta|_\nu \le 1$ for any ultrametric absolute value. Indeed,

$$|a + b\theta|_\nu \le \max(|a|_\nu, |b\theta|_\nu) = \max(|a|_\nu, |b|_\nu).$$

Hence we only need to take the archimedean absolute values into account to estimate the height. Recall that if θ is of order d, its conjugates are the other roots of unity of order d. Hence

$$H(a + b\theta)^{d-1} = \prod_{k=1}^{d-1} \max(1, |a + b e^{2ik\pi/d}|).$$

Assume first that a and b are of the same sign, and for instance positive. Then $|a + b e^{2ik\pi/d}| \ge a + b \cos(2ik\pi/d) \ge 1 + \cos(2\pi/5)$ if $k \le d/5$. Hence

$$H(a + b\theta) \ge (1 + \cos(2\pi/5))^{\lfloor d/5 \rfloor / (d-1)}.$$

This lower bound is always > 1.045 since $d \ge 5$, and its limit as $d \to +\infty$, which is equal to $(1 + \cos(2\pi/5))^{1/5}$, is > 1.055.

To complete the proof, we now consider the case where a and b have opposite signs. Assume for instance that $a \ge 1$

and $b \leq -1$. Then $|a + be^{2ik\pi/d}| \geq a + b\cos(2ik\pi/d) \geq 3/2$ if $d/3 \leq k \leq 2d/3$. Hence $H(a + b\theta) \geq (3/2)^{\lfloor d/3 \rfloor/(d-1)}$. This lower bound is again always > 1.10 and its limit as $d \to +\infty$, which is equal to $(3/2)^{1/3}$, is > 1.14. □

We now deal with the case where a and b are rational numbers.

LEMMA 2. *There is an absolute constant $\kappa > 1.045$ such that the following holds: for any pair (a, b) of rational numbers, different from the 5 excluded pairs of Theorem 1, there exists an infinite set $U(a, b)$ of roots of unity such that $H(a+b\theta) \geq \kappa$ for any $\theta \in U(a, b)$.*

PROOF. Let (a, b) be a pair of rational numbers different from the 5 excluded pairs. If $b = 0$, $H(a + b\theta) = H(a) \geq 2$ since $a \notin \{-1, 0, 1\}$. If $a = 0$, $H(a + b\theta) = H(b\theta) = H(b) \geq 2$ since $b \notin \{-1, 0, 1\}$ (indeed, for any ν we have $|b\theta|_\nu = |b|_\nu|\theta|_\nu = |b|_\nu$). One may therefore take for $U(a, b)$ the set of all roots of unity if $a = 0$ or $b = 0$.

Also, we have shown in Lemma 1 that one may take $U(a, b) = \mathcal{U}_{\geq 5}$ if $a \in \mathbb{Z} \setminus \{0\}$ and $b \in \mathbb{Z} \setminus \{0\}$. We therefore assume for the remainder of the proof that a and b are both nonzero, and that they are not both integers.

By reduction to the same denominator one finds integers $c, d, e \in \mathbb{Z}\setminus\{0\}$ such that $e \geq 2$, $\gcd(c, d, e) = 1$, and $a+b\theta = (d\theta - c)/e$ for any root of unity θ. Let $x = a + b\theta$, let p be a prime factor of e, and fix any ν such that $\nu|p$. Since $|x|_\nu \geq p|y|_\nu$ where $y = d\theta - c$, it remains to lower bound $|y|_\nu$ (note that we have the upper bound $|y|_\nu \leq 1$). If θ is a n-th root of unity, we have

$$(y + c)^n = d^n. \tag{15}$$

We first assume that p divides c. In this case p cannot divide d since $\gcd(c, d, e) = 1$. Hence (15) implies that $|y|_\nu = 1$, so that $|x|_\nu \geq p$. Since this is true for any ν such that $\nu|p$, we have $H(x) \geq p \geq 2$. If p divides c, one may therefore take $U(a, b)$ equal to the set of all roots of unity.

We now examine the case $p \nmid c$. We assume that $\theta \neq 1$ is a n-th root of unity, and distinguish 3 subcases.

(i) If $c = d$, we shall see that $|y|_\nu = 1$ whenever $p \nmid n$. Indeed, $|y|_\nu = |c|_\nu|\theta - 1|_\nu = |\theta - 1|_\nu$. Set $z = \theta - 1$. Since $(z + 1)^n = 1$ and $z \neq 0$, it follows from the binomial formula that

$$z^{n-1} + nz^{n-2} + \binom{n}{2}z^{n-3} + \cdots + \binom{n}{2}z = -n.$$

Hence $|z|_\nu = 1$ since $|n|_\nu = 1$. We conclude that $H(x) \geq p \geq 2$, and one may take $U(a, b)$ equal to the union for all integers n such that $p \nmid n$ of the set of n-th roots of unity different from 1.

(ii) The second subcase ($c \neq d$ and $p \nmid d - c$) is similar, but slightly more involved. Let $U(a, b)$ be the set of positive integers n such that $p \nmid (d^n - c^n)$. Note that $U(a, b)$ is infinite since $n \in U(a, b)$ or $n + 1 \in U(a, b)$ for any $n \geq 1$, as is shown as follows: assume the contrary, namely that $p \mid d^{n+1} - c^{n+1}$ and $p \mid d^n - c^n$. It follows that $p \mid (d^{n+1}-c^{n+1})-d(d^n-c^n) = c^n(d-c)$. This is impossible since $p \nmid c$.

Let $n \in U(a, b)$. Using again the binomial formula, it follows from (15) that

$$y^n + ncy^{n-1} + \binom{n}{2}c^2y^{n-2} + \cdots + nc^{n-1}y = d^n - c^n.$$

Since $|d^n - c^n|_\nu = 1$, we must have $|y|_\nu \geq 1$ (so that in fact $|y|_\nu = 1$). We conclude that $H(x) \geq p \geq 2$ if $\theta \in U(a, b)$.

(iii) The last subcase occurs when $c \neq d$ and $p \mid d - c$. We can write $y = d\theta - c = c(\theta-1)+(c-d)$. By hypothesis $|c - d|_\nu \leq 1/p$, and by subcase (i) $|c(\theta - 1)|_\nu = 1$ if θ belongs to the set $U(a, b)$ defined in that subcase. We may therefore take the same $U(a, b)$, and we conclude again that $H(x) \geq 2$ if $\theta \in U(a, b)$.

We have shown that $H(x) \geq 2$ whenever $\theta \in U(a, b)$ and $a \notin \mathbb{Z} \setminus \{0\}$, $b \notin \mathbb{Z} \setminus \{0\}$, $a = 0$ or $b = 0$. One may therefore take $\kappa = \min(2, \kappa_1)$ (so in fact $\kappa = \kappa_1$). □

PROPOSITION 1. *Let (a, b) be a pair of rational numbers different from the five excluded pairs of Theorem 1. Let f be a polynomial of the form (12), and let $k \geq 1$ be an integer. Write $f = g + h$ where g collects all the terms of f with $\beta_j \leq k$ and h collects all the terms of f with $\beta_j > k$. Let $u = \min\{\beta_j;\ \beta_j > k\}$. Assume that θ is a zero of f, and that θ belongs to the set $U(a, b)$ of Lemma 2. If*

$$u - k > \log(t\,H(f))/\log\kappa, \tag{16}$$

where κ is as in Lemma 2, then θ is a common zero of g and h.

PROOF. We may assume that each of the two polynomials g and h collects at most t of the $t+1$ terms of f (otherwise, the result is clear). Assume by contradiction that $g(\theta) \neq 0$. Let $K = \mathbb{Q}[\theta]$ and $\nu \in M_K$. If $|a + b\theta|_\nu \geq 1$, each term of $g(\theta)$ satisfies $|a_j\theta^{\alpha_j}(a + b\theta)^{\beta_j}| \leq |f|_\nu|a + b\theta|_\nu^k$, therefore

$$|g(\theta)|_\nu \leq \max(1, |t|_\nu)|f|_\nu|a + b\theta|_\nu^k \ \text{ if }\ |a + b\theta|_\nu \geq 1.$$

A similar argument shows that

$$|h(\theta)|_\nu \leq \max(1, |t|_\nu)|f|_\nu|a + b\theta|_\nu^u \ \text{ if }\ |a + b\theta|_\nu \leq 1.$$

We have $|g(\theta)|_\nu = |h(\theta)|_\nu$, so we can combine these two statements in

$$\max(1, |a + b\theta|_\nu|)^{u-k} \cdot |g(\theta)|_\nu \leq \max(1, |t|_\nu) \cdot |f|_\nu \cdot |a + b\theta|_\nu^u.$$

Raise this to the power $d_\nu/[K : \mathbb{Q}]$ and take the product over $\nu \in M_K$. Using the fact that $H(t) = t$, and applying (14) to $g(\theta)$ and $a+b\theta$ (which are both supposed to be nonzero) one finds that $H(a+b\theta)^{u-k} \leq t \cdot H(f)$. However, $H(a+b\theta) \geq \kappa$ by Lemma 2. This is in contradiction with (16). □

4.3 Deterministic algorithms

THEOREM 2. *We have a polynomial-time deterministic algorithm for deciding whether a polynomial of the form (12) is identically zero.*

Note that there is a trivial algorithm which deals with the case where (a, b) is one of the five excluded pairs of Theorem 1. In the following we therefore assume that (a, b) is not one of these five excluded pairs, and we fix a rational number $\epsilon > 0$ such that one may take $\kappa = 2^\epsilon$ in Lemma 2. Set $\delta = \lceil n/\epsilon \rceil$, where n is the unique integer such that $2^{n-1} \leq t\,H(f) < 2^n$. Assume that the β_j's are sorted by nondecreasing order as in Theorem 1. There is a unique integer $s \geq 1$ and a unique partition of the set $\{0, 1, \ldots, t\}$ in subsets $U_1, \ldots, U_s$ of consecutive integers with the following property: if an integer j belongs to U_l then $j + 1$ also belongs to U_l if $\beta_{j+1} < \beta_j + \delta$, otherwise $j + 1$ belongs to U_{l+1} (to obtain this partition, just sweep the list of the β_j's from left to right and create a new subset whenever an element β_j such that $\beta_{j+1} - \beta_j \geq \delta$ is found). Let $f_l = \sum_{j \in U_l} a_j X^{\alpha_j}(a+bX)^{\beta_j}$. By construction $f = \sum_{j=1}^{s} f_l$ and by Theorem 1, f is identically zero iff all the f_l are identically zero. Indeed, we have $\delta > \log(t\,H(f))/\log\kappa$, where $\kappa = 2^{\epsilon}$. Furthermore, we can write $f_l = (a+bX)^{\gamma_l} g_l$ where

$$g_l = \sum_{j \in U_l} a_j X^{\alpha_j}(a+bX)^{\delta_{j,l}}, \qquad (17)$$

$\gamma_l = \min\{\beta_j; j \in U_l\}$, and $\delta_{j,l} = \beta_j - \gamma_l$. Each exponent $\delta_{j,l}$ satisfies $0 \leq \delta_{j,l} < \delta$. The g_l are all identically zero iff f is identically zero. We can now describe our main algorithm.

1. Compute $H(f)$ as explained before Theorem 1 and the integer δ defined above.
2. Construct the list $(g_1, \ldots, g_s)$ defined by (17).
3. Express each polynomial $(a+bX)^{\delta_{j,l}}$ as a sum of powers of X.
4. Substitute in (17) to express each g_l as a sum of powers of X, and decide whether the g_l are all identically zero. If so, output "$f = 0$". Otherwise, output "$f \neq 0$".

The correctness of this algorithm follows from the discussion after Theorem 2, and it is clear that steps 1 and 2 run in polynomial time. Step 3 also runs in polynomial time since $\delta_{j,l} < \delta$ and δ is bounded by a polynomial in the input size (so we can simply expand $(a+bX)^{\delta_{j,l}}$ by brute force). Finally, in step 4 we express g_l as a sum of at most $\delta|U_l| \leq \delta(t+1)$ terms. This completes the proof of the running time estimate, and of Theorem 2.

Remark 3. One can deal at no additional expense with polynomials of the slightly more general form:

$$f(X) = \sum_{j=0}^{t} a_j (c+dX)^{\alpha_j}(a+bX)^{\beta_j}.$$

Indeed, the change of variable $Y = c + dX$ yields a polynomial $g(Y)$ of form (12). The only case which cannot be handled in this way is the seemingly trival one $b = d = 0$. Here one has to decide whether the rational number $\sum_{j=0}^{t} a_j c^{\alpha_j} a^{\beta_j}$ is equal to zero. It is not clear whether this can be done in deterministic polynomial time, even if a, c and the a_j are all integers. □

Theorem 2 yields a deterministic algorithm for Step 2 with $\mu = 1$ in algorithm Supersparse Factorization in Section 3. However, that variant still fails to compute linear factors of polynomials of the form (11), which we remedy next.

Theorem 3. *We have a polynomial-time deterministic algorithm that finds all linear factors of a supersparse polynomial $g(X,Y) = \sum_{j=0}^{t} a_j X^{\alpha_j} Y^{\beta_j}$.*

Proof. We first find all linear factors of f that are in $\mathbb{Q}[X]$ by applying Lenstra's method to the coefficients of Y^{β_j} (this is similar to step 0 of the algorithm of section 3). After that, it remains to find all factors of the form $Y - bX - a$. There are five special cases for the pair (a,b), which correspond to the five excluded pairs of Theorem 1. As pointed out in the proof of Theorem 2, one can check easily for each of these five pairs whether $g(X, a+bX) = 0$. In the following we therefore look for factors $Y - bX - a$ where (a,b) is different from the five excluded pairs. As in Theorems 1 and 2, we assume that the β_j are sorted by nondecreasing order.

The idea is to use Theorem 1 to reduce this problem to several factoring problems about dense polynomials. Let $U_1, \ldots, U_s$ be the partition of the set of indices $\{0, 1, \ldots, t\}$ which is constructed when the algorithm of Theorem 2 is run on the polynomial $f(X) = g(X, a+bX)$. Crucially, this partition is in fact independent of the pair (a,b). As in the proof of Theorem 2, one can write $g = \sum_{j=1}^{s} Y^{\gamma_l} g_l$, where $g_l = \sum_{j \in U_l} a_j X^{\alpha_j} Y^{\delta_{j,l}}$, $\gamma_l = \min\{\beta_j; j \in U_l\}$, and $\delta_{j,l} = \beta_j - \gamma_l$. By Theorem 1, the linear factors of g are (excluding excluded pairs!) the common linear factors of the g_l. We have therefore reduced our initial problem to the computation of the linear factors of each g_l. This progress is significant since, as shown in the proof of Theorem 2, in every g_l the exponents $\delta_{j,l}$ of variable Y are "small" (polynomially bounded in the size of the input polynomial g). The exponents of X may still be large, however. To deal with this problem we run the same factoring algorithm on input g_l instead of g, with the roles of variables X and Y interchanged. This reduces the problem to the computation of the linear factors of polynomials where the exponents of X and Y are all "small". One can then use any deterministic polynomial time algorithm that finds the linear factors of a dense polynomial. □

5. NP-HARDNESS OF SUPERSPARSE BIVARIATE IRREDUCIBILITY

In [22] and the earlier papers cited there, NP-hardness results are derived for supersparse polynomials over the integers. In [7, 17] several hard problems are extended for supersparse polynomials over finite fields. We give similar NP-hard problems over finite fields, but now for finite fields of arbitrary characteristic.

Figure 1 shows D. Plaisted's model for 3-SAT in n Boolean variables $z_1, \ldots, z_n$. Clauses correspond to factors of $X^N - 1$ with $N = \prod_{j=1}^{n} p_j$, where p_j distinct primes. We note that all $\mathrm{Poly}(C_i)$ are supersparse polynomials for any clause C_i with one, two or three literals. An immediate consequence of the construction, justified by the rootsets in Figure 1 that are associated with the polynomials, is that the conjunctive normal form $C_1 \wedge \cdots \wedge C_s$ is satisfiable if and only if $\mathrm{GCD}(\mathrm{Poly}(C_1), \ldots, \mathrm{Poly}(C_s)) \neq 1$. We first generalize that reduction to coefficients from an arbitrary field.

Let $p \nmid N$ be a fresh prime and let

$$\Psi_N(y) = \prod_{1 \leq b < N, \mathrm{GCD}(b,N)=1} (X - e^{2b\pi \mathbf{i}/N}) \in \mathbb{Z}[y]$$

be the cyclotomic equation of order N.

Since $\zeta = (y \bmod \Psi_N(y))$ is a representation for a primitive N-th root of unity, we have over the integers

$$X^N - 1 \equiv (X - y^1)(X - y^2) \cdots (X - y^N) \bmod (\Psi_N(y)). \quad (18)$$

Let $\mathbb{F}_q \supseteq \mathbb{Z}_p$ be the splitting field of $\Psi_N(y) \bmod p$ (one has $q = p^{\lambda}$ where λ is the multiplicative order of p modulo N) and let $\zeta \in \mathbb{F}_q$ with $\Psi_N(\zeta) = 0$ in $\mathbb{F}_q$. Taking (18) modulo p and evaluating the resulting polynomial identity at $y = \zeta$, that is, taking it modulo $y - \zeta \mid \Psi_N(y)$, one obtains

$$X^N - 1 = (X - \zeta^1)(X - \zeta^2) \cdots (X - \zeta^N) \text{ in } \mathbb{F}_q[X].^{\S} \quad (19)$$

[§] With $N = p^{\mu} - 1$, $1 \leq \mu$, we have proven that the multiplicative

Formula	Polynomial	Rootset
z_j	$X^{N/p_j} - 1$	$\{(e^{2\pi \mathbf{i}/N})^a \mid a \equiv 0 \pmod{p_j}\}$
$\neg z_k$	$\dfrac{X^N - 1}{X^{N/p_k} - 1} = \sum_{i=0}^{p_k-1} X^{iN/p_k}$	$\{(e^{2\pi \mathbf{i}/N})^b \mid b \not\equiv 0 \pmod{p_k}\}$
$L_1 \vee L_2 \vee L_3$	$\mathrm{LCM}(\mathrm{Poly}(L_1), \mathrm{Poly}(L_2), \mathrm{Poly}(L_3))$	$\bigcup_{j=1}^{3} \mathrm{Roots}(L_j)$
$z_j \vee z_k \vee z_l$	$\dfrac{(X^{N/p_j} - 1)(X^{N/p_k} - 1)(X^{N/p_l} - 1)(X^{N/(p_j p_k p_l)} - 1)}{(X^{N/(p_k p_j)} - 1)(X^{N/(p_k p_l)} - 1)(X^{N/(p_j p_l)} - 1)}$	
$z_j \vee \neg z_k \vee z_l$	$\dfrac{(X^{N/(p_j p_k)} - 1)(X^N - 1)(X^{N/(p_k p_l)} - 1)}{(X^{N/p_k} - 1)(X^{N/(p_j p_k p_l)} - 1)}$	
$\neg z_j \vee \neg z_k \vee z_l$	$\dfrac{(X^N - 1)(X^{N/(p_j p_k p_l)} - 1)}{X^{N/(p_j p_k)} - 1}$	
$\neg z_j \vee \neg z_k \vee \neg z_l$	$\dfrac{X^N - 1}{X^{N/(p_j p_k p_l)} - 1}$	
$C_1 \wedge \cdots \wedge C_s$	$\mathrm{GCD}(\mathrm{Poly}(C_1), \ldots, \mathrm{Poly}(C_s))$	$\bigcap_{i=1}^{s} \mathrm{Roots}(C_i)$

Figure 1: D. Plaisted's polynomials for literals, clauses and CNFs ($N = \prod_{j=1}^{n} p_j$).

Since $p \nmid N$, the polynomial $X^N - 1 \bmod p$ has no multiple roots, and we can replace $e^{2\pi \mathbf{i}/N}$ by ζ in D. Plaisted's construction. We thus have established the following.

Proposition 2. *The set of tuples of relatively prime supersparse polynomials in $\mathbb{F}_q[X]$ is co-NP-hard for arbitrary $q = p^\mu$.*

In [22], co-NP-hardness is shown for pairs of supersparse relatively prime polynomials over $\mathbb{Z}$. However, D. Plaisted's pairs do not remain relatively prime modulo all primes p. We overcome that deficiency via randomized reductions.

Lemma 3. *Let $f_i(X) \in K[X]$ be nonzero polynomials for $i = 1, \ldots, s$, $s \geq 2$, K a field, $d = \deg(f_1)$ and $S \subset K$. Then for randomly chosen $c_i \in S$, $3 \leq i \leq s$, we have the probability estimate*

$$\mathrm{Prob}\Big(\operatorname*{GCD}_{1 \leq i \leq s}(f_i) = \mathrm{GCD}(f_1, f_2 + \sum_{i=3}^{s} c_i f_i)\Big) \geq 1 - d/|S| \quad (20)$$

Furthermore, if f_2 is squarefree and $e = \deg(f_2) \geq \deg(f_i)$ for $i \geq 3$, then with probability no less than $1 - (2e-1)/|S|$ the polynomial $f_2 + \sum_{i=3}^{s} c_i f_i$ will remain squarefree.

Proof. The estimate (20) is Lemma 2 in [5]. Squarefreeness follows by the same techniques as given there by considering the discriminant of $F = f_2 + \sum_{i=3}^{s} y_i f_i$ as a Sylvester resultant of F and $\partial F / \partial X$ with symbolic y_i. □

We obtain the following NP-hardness problems under randomized reduction, which generalize the results in [17] to arbitrary characteristic.

Theorem 4. *The set of pairs of relatively prime supersparse polynomials in $K[X]$, the set of squarefree supersparse polynomials in $K[X]$, and the set of irreducible supersparse polynomials in $K[X, Y]$ are co-NP-hard under randomized reduction for $K = \mathbb{Q}$ and $K = \mathbb{F}_q$ with arbitrary p and sufficiently large $q = p^m$. Co-NP-hardness of irreducibility remains valid if we assume that the supersparse bivariate polynomials are monic in X.*

group of a finite field $\mathbb{F}_{p^\mu}$ is cyclic. For that proof the only property of Ψ_N needed is that it is the monic integral minimum polynomial of a primitive N-th root of unity in $\mathbb{C}$.

Proof. Reduction to two polynomials follows from Proposition 2 and Lemma 3 (cf. [7]). NP-hardness of squarefreeness follows by considering the product $f_1(f_2 + \sum_{i=3}^{s} c_i f_i)$. Since D. Plaisted's polynomials $f_i = \mathrm{Poly}(C_i)$ are divisors of $X^N - 1$ and therefore are squarefree, with high probability both factors will be squarefree. Therefore their product is not squarefee if and only if the two factors have a common GCD, which is NP-hard under randomized reduction. NP-hardness of irreducibility follows by considering the polynomial

$$F(X, Y) = X^u f_1(X) + Y(f_2(X) + X \sum_{i=3}^{s} c_i f_i(X))$$

for sufficiently large u to make $F(X, Y)$ monic in X (cf. [17, Proof of Theorem 1]). Note that D. Plaisted's polynomials $f_i = \mathrm{Poly}(C_i)$ are not divisible by X. Shifting $f_3, \ldots, f_s$ by a factor of X ensures that $f_2 + c_3 X f_3 + \cdots + c_s X f_s$ is relatively prime to X^u. Clearly, $F(X, Y)$ is irreducible if and only if $\mathrm{GCD}(f_1, f_2 + X \sum_{i=3}^{s} c_i f_i) = 1$, that if and only if $\mathrm{GCD}(f_1, \ldots, f_s) = 1$ with high probability. □

For example, co-NP-hardness of supersparse bivariate irreducibility yields via [1] the following reduction to integer factoring.

Corollary 1. *Suppose we have a Monte Carlo polynomial-time irreducibility test for supersparse polynomials in $\mathbb{F}_{2^m}[X, Y]$ for sufficiently large m. Then large integers can be factored in Las Vegas polynomial-time.*

Already in [17], Hilbert irreducibility is mentioned as a means to establish NP-hardness of irreducibility of supersparse polynomials in $\mathbb{Z}[X]$. Note that one can evaluate the linear variable Y at a polynomial-sized integer. However, no proven effective versions seem to be known that would yield a randomized polynomial reduction (cf. [26, 24]). Nonetheless, if a fast irreducibility test of supersparse polynomials in $\mathbb{Z}[X]$ were discovered, we believe that Hilbert irreducibility would yield fast algorithms for NP-complete problems, thus resulting in what we call a "good heuristic" for NP-completeness [15]. Of course, the Hilbert irreducibility theorem is not valid over $\mathbb{F}_q$ and the hardness of supersparse irreducibility in $\mathbb{F}_q[X]$ remains open.

In addition, the complexity of root finding of supersparse polynomials over finite fields is open. In [15], we have posed two open problems: Given a prime number p and integers $b, c \in \mathbb{Z}_p$ and α, β with with $p-1 > \alpha > \beta > 0$, compute $a \in \mathbb{Z}_p$ such that $a^\alpha + ba^\beta + c \equiv 0 \pmod{p}$ in $(\log p)^{O(1)}$ bit operations. Alternatively, prove that computing a root in $\mathbb{Z}_p$ of a polynomial given by straight-line program over $\mathbb{Z}_p$ is NP-hard.

Acknowledgments: Pascal Koiran would like to thank Nicolas Brisebarre for useful number-theoretic discussions on Lemma 1. We thank Joachim von zur Gathen and Igor Shparlinksi for pointing out their work on NP-hardness to us. The authors would like to thank the referees whose comments where very useful for improving the final version of this paper.

6. REFERENCES

[1] Agrawal, M., Kayal, N., and Saxena, N. PRIMES is in P. Manuscript, 2002. Available from `http://www.cse.iitk.ac.in/news/primality.pdf`.

[2] Barvinok, A. I., and Woods, K. Short rational generating functions for lattice point problems. *J. Amer. Math. Soc. 16* (2003), 957–979.

[3] Cucker, F., Koiran, P., and Smale, S. A polynomial time algorithm for diophantine equations in one variable. *J. Symbolic Comput. 27*, 1 (1999), 21–29.

[4] De Loera, J. A., Hemmecke, R., Huggins, P., Sturmfels, B., and Yoshida, R. Short rational functions for toric algebras and applications. *J. Symbolic Comput. 38*, 2 (2004), 959–973.

[5] Díaz, A., and Kaltofen, E. On computing greatest common divisors with polynomials given by black boxes for their evaluation. In *Proc. 1995 Internat. Symp. Symbolic Algebraic Comput. (ISSAC'95)* (New York, N. Y., 1995), A. H. M. Levelt, Ed., ACM Press, pp. 232–239.

[6] von zur Gathen, J., and Gerhard, J. *Modern Computer Algebra*. Cambridge University Press, Cambridge, New York, Melbourne, 1999. Second edition 2003.

[7] von zur Gathen, J., Karpinski, M., and Shparlinski, I. Counting curves and their projections. *Computational Complexity 6*, 1 (1996/1997), 64–99.

[8] Goldstein, A. J., and Graham, R. L. A Hadamard-type bound on the coefficients of a determinant of polynomials. *SIAM Rev. 16* (1974), 394–395.

[9] Győry, K., Iwaniec, H., and Urbanowicz, J., Eds. *Number Theory in Progress* (1999), vol. 1 Diophantine Problems and Polynomials, Stefan Banach Internat. Center, Walter de Gruyter Berlin/New York. Proc. Internat. Conf. Number Theory in Honor of the 60th Birthday of Andrzej Schinzel, Zakopane, Poland June 30–July 9, 1997.

[10] Hindry, M., and Silverman, J. H. *Diophantine Geometry: an Introduction*, vol. 201 of *Graduate Texts in Mathematics*. Springer, 2000.

[11] Kaltofen, E. Sparse Hensel lifting. In *EUROCAL 85 European Conf. Comput. Algebra Proc. Vol. 2* (Heidelberg, Germany, 1985), B. F. Caviness, Ed., Lect. Notes Comput. Sci., Springer Verlag, pp. 4–17. Proofs in [12].

[12] Kaltofen, E. Sparse Hensel lifting. Tech. Rep. 85-12, Rensselaer Polytechnic Instit., Dept. Comput. Sci., Troy, N. Y., 1985.

[13] Kaltofen, E. Greatest common divisors of polynomials given by straight-line programs. *J. ACM 35*, 1 (1988), 231–264.

[14] Kaltofen, E. Polynomial factorization 1987-1991. In *Proc. LATIN '92* (Heidelberg, Germany, 1992), I. Simon, Ed., vol. 583 of *Lect. Notes Comput. Sci.*, Springer Verlag, pp. 294–313.

[15] Kaltofen, E. Polynomial factorization: a success story. In *ISSAC 2003 Proc. 2003 Internat. Symp. Symbolic Algebraic Comput.* (New York, N. Y., 2003), J. R. Sendra, Ed., ACM Press, pp. 3–4. Abstract for invited talk. Slides: `http://www.math.ncsu.edu/~kaltofen/bibliography/03/issac.pdf`.

[16] Kaltofen, E., and Lee, W. Early termination in sparse interpolation algorithms. *J. Symbolic Comput. 36*, 3–4 (2003), 365–400. Special issue Internat. Symp. Symbolic Algebraic Comput. (ISSAC 2002). Guest editors: M. Giusti & L. M. Pardo.

[17] Karpinski, M., and Shparlinski, I. On the computational hardness of testing square-freeness of sparse polynomials. In *Proc. AAECC-13* (Heidelberg, Germany, 1999), vol. 1719 of *Lect. Notes Comput. Sci.*, Springer Verlag, pp. 492–497.

[18] Lang, S. *Algebra*. Addison-Wesley, 1993.

[19] Lenstra, Jr., H. W. Finding small degree factors of lacunary polynomials. In Győry et al. [9], pp. 267–276.

[20] Lenstra, Jr., H. W. On the factorization of lacunary polynomials. In Győry et al. [9], pp. 277–291.

[21] Loos, R. Computing rational zeros of integral polynomials by p-adic expansion. *SIAM J. Comput. 12*, 2 (1983), 286–293.

[22] Plaisted, D. A. New NP-hard and NP-complete polynomial and integer divisibility problems. *Theoretical Comput. Sci. 13* (1984), 125–138.

[23] Rosser, J. B., and Schoenfeld, L. Approximate formulas of some functions of prime numbers. *Illinois J. Math. 6* (1962), 64–94.

[24] Schinzel, A., and Zannier, U. The least admissible value of the parameter in Hilbert's Irreducibility Theorem. *Acta Arithm. 65* (1995), 371–391.

[25] Shparlinski, I. E. Computing Jacobi symbols modulo sparse integers and polynomials and some applications. *J. Algorithms 36* (2000), 241–252.

[26] Sprindžuk, V. G. Arithmetic specializations in polynomials. *J. reine angew. Math. 340* (1983), 26–52.

[27] Waldschmidt, M. *Diophantine approximation on linear algebraic groups*. Springer Verlag, Heidelberg, Germany, 2000.

[28] Zippel, R. E. *Probabilistic algorithms for sparse polynomials*. PhD thesis, Massachusetts Inst. of Technology, Cambridge, USA, Sept. 1979.