Computing Sparse Multiples of Polynomials*

Mark Giesbrecht, Daniel S. Roche, and Hrushikesh Tilak

Cheriton School of Computer Science, University of Waterloo
{mwg,droche,htilak}@cs.uwaterloo.ca
http://www.cs.uwaterloo.ca/~{mwg,droche}

Abstract. We consider the problem of finding a sparse multiple of a polynomial. Given $f \in \mathsf{F}[x]$ of degree d, and a desired sparsity t, our goal is to determine if there exists a multiple $h \in \mathsf{F}[x]$ of f such that h has at most t non-zero terms, and if so, to find such an h. When $\mathsf{F} = \mathbb{Q}$ and t is constant, we give a polynomial-time algorithm in d and the size of coefficients in h. When F is a finite field, we show that the problem is at least as hard as determining the multiplicative order of elements in an extension field of F (a problem thought to have complexity similar to that of factoring integers), and this lower bound is tight when $t = 2$.†

1 Introduction

Let F be a field, which will later be specified either to be the rational numbers ($\mathbb{Q}$) or a finite field with q elements ($\mathbb{F}_q$). We say a polynomial $h \in \mathsf{F}[x]$ is *t-sparse* (or *has sparsity t*) if it has at most t nonzero coefficients in the standard power basis; that is, h can be written in the form

$$h = h_1 x^{d_1} + h_2 x^{d_2} + \cdots + h_t x^{d_t} \quad \text{for } h_1, \ldots, h_t \in \mathsf{F} \text{ and } d_1, \ldots, d_t \in \mathbb{N}. \quad (1.1)$$

Sparse polynomials have a compact representation as a sequence of coefficient-degree pairs $(h_1, d_1), \ldots, (h_t, d_t)$, which allow representation and manipulation of very high degree polynomials. Let $f \in \mathsf{F}[x]$ have degree d. We examine the computation a t-sparse multiple of f. That is, we wish to determine if there exist $g, h \in \mathsf{F}[x]$ such that $fg = h$ and h has prescribed sparsity t, and if so, to find such an h. We do not attempt to find g, as it may have a superpolynomial number of terms even though h has a compact representation (see Theorem 3.6).

Sparse multiples over finite fields have cryptographic applications. Their computation is used in correlation attacks on LFSR-based stream ciphers (Aimani and von zur Gathen, 2007; Didier and Laigle-Chapuy, 2007). The security of the TCHo cryptosystem is also based on the conjectured computational hardness of sparsest multiple computation over $\mathbb{F}_2[x]$ (Aumasson et al., 2007); our results provide further evidence that this is in fact a computationally difficult problem.

* The authors would like to thank the Natural Sciences and Engineering Research Council of Canada (NSERC), and MITACS

† Proofs of all statements in this paper may be found at http://arxiv.org/abs/1009.3214

Sparse multiples can be useful for extension field arithmetic (Brent and Zimmermann, 2003) and designing interleavers for error-correcting codes (Sadjadpour et al., 2001). The linear algebra formulation in Section 2 relates to finding the minimum distance of a binary linear code (Berlekamp et al., 1978; Vardy, 1997) as well as "sparsifications" of linear systems (Egner and Minkwitz, 1998).

One of our original motivations was to understand the complexity of sparse polynomial *implicitization* over $\mathbb{Q}$ or $\mathbb{R}$: Given a list of zeros of a (typically multivariate) function, we wish find a sparse polynomial with those zeros (see, e.g., Emiris and Kotsireas (2005)). Our work here can be thought of as a univariate version of implicitization, though a reduction from the multi- to univariate case via Kronecker-like substitutions seems quite feasible.

In general, we assume that the desired sparsity t is a constant. This seems reasonable given that over a finite field, even for $t = 2$, the problem is probably computationally hard (Theorem 5.1). In fact, we have reason to conjecture that the problem is intractable over $\mathbb{Q}$ or $\mathbb{F}_q$ when t is a parameter. Our algorithms are exponential in t but polynomial in the other input parameters when $t \in O(1)$.

Over $\mathbb{Q}[x]$, the analysis must consider coefficient size, and we will count machine word operations in our algorithms to account for coefficient growth. We follow the conventions of Lenstra (1999) and define the *height* of a polynomial: Let $f \in \mathbb{Q}[x]$ and $r \in \mathbb{Q}$ the least positive rational number such that $rf \in \mathbb{Z}[x]$. If $rf = \sum_i a_i x^{d_i}$ with each $a_i \in \mathbb{Z}$, then the *height* of f, written $\mathcal{H}(f)$, is $\max_i |a_i|$.

We examine variants of the sparse multiple problem over $\mathbb{F}_q$ and $\mathbb{Q}$. Since every polynomial in $\mathbb{F}_q$ has a 2-sparse multiple of high degree, given $f \in \mathbb{F}_q[x]$ and $n \in \mathbb{N}$ we consider the problem of finding a t-sparse multiple of f with degree at most n. For input $f \in \mathbb{Q}[x]$ of degree d, we consider algorithms which seek t-sparse multiples of height bounded above by an additional input value $c \in \mathbb{N}$. We present algorithms requiring time polynomial in d and $\log c$.

The remainder of the paper is structured as follows.

In Section 2, we consider the straightforward linear algebra formulation of the sparse multiple problem. This is useful over $\mathbb{Q}[x]$ once a bound on the output degree is derived, and also allows us to bound the output size. In addition, it connects our problems with related NP-complete coding theory problems.

In Section 3 we consider the problem of finding the least-degree binomial multiple of a rational polynomial. A polynomial-time algorithm in the size of the input is given which completely resolves the question in this case. This works despite the fact that we show polynomials with binomial multiples whose degrees and heights are both exponential in the input size!

In Section 4 we consider the more general problem of finding a t-sparse multiple of an input $f \in \mathbb{Q}[x]$ without repeated cyclotomic factors. We present a polynomial-time algorithm in the size of f and a given height bound.

Section 5 shows that, even for $t = 2$, finding a t-sparse multiple of a polynomial $f \in \mathbb{F}_q[x]$ is at least as hard as finding multiplicative orders in an extension of $\mathbb{F}_q$ (a problem thought to be computationally difficult). This lower bound is shown to be tight for binomial multiples.

Open questions and avenues for future research are discussed in Section 6.

2 Linear Algebra Formulation

The sparsest multiple problem can be formulated using linear algebra. This requires specifying bounds on degree, height and sparsity; later some of these parameters will be otherwise determined. This approach also highlights the connection to some problems from coding theory. We exhibit a randomized algorithm for finding a t-sparse multiple of a rational polynomial of degree d, given bounds c and n on the height and degree of the multiple respectively. When $t \in O(1)$, the algorithm runs in time polynomial in n and $\log \mathcal{H}(f)$ and returns the desired output with high probability. We also conjecture the intractability of some of these problems, based on similar problems in coding theory. Finally, we show that the problem of finding the sparsest vector in an integer lattice is NP-complete, which was conjectured by Egner and Minkwitz (1998).

Suppose R is a principal ideal domain, and basic ring operations have unit cost. Let $f \in \mathsf{R}[x]$ have degree d and $n \in \mathbb{N}$ be given. Suppose $g, h \in \mathsf{R}[x]$ have degrees $n-d$ and n respectively, with $f = \sum_0^d f_i x^i$, $g = \sum_0^{n-d} g_i x^i$ and $h = \sum_0^n h_i x^i$. The coefficients in equation $fg = h$ satisfy the linear system (2.1).

$$\underbrace{\begin{bmatrix} f_0 & & & \\ f_1 & \ddots & & \\ \vdots & \ddots & f_0 & \\ f_d & \ddots & f_1 & \\ & \ddots & \vdots & \\ & & f_d & \end{bmatrix}}_{A_{f,n}} \underbrace{\begin{bmatrix} g_0 \\ g_1 \\ \vdots \\ g_{n-d} \end{bmatrix}}_{v_g} = \underbrace{\begin{bmatrix} h_0 \\ h_1 \\ \\ \vdots \\ \\ h_n \end{bmatrix}}_{v_h} \quad (2.1) \qquad \underbrace{\begin{bmatrix} 1 & \alpha_1 & \cdots & \alpha_1^n \\ 1 & \alpha_2 & \cdots & \alpha_2^n \\ \vdots & \vdots & \vdots & \vdots \\ 1 & \alpha_d & \cdots & \alpha_d^n \end{bmatrix}}_{A_n(\alpha_1,\ldots,\alpha_d)} \begin{bmatrix} h_0 \\ h_1 \\ \\ \vdots \\ \\ h_n \end{bmatrix} = 0 \quad (2.2)$$

Thus, a multiple of f of degree at most n and sparsity at most t corresponds to a vector with at most t nonzero coefficients (i.e., a t-sparse vector) in the linear span of $A_{f,n}$. If $f \in \mathsf{R}[x]$ is squarefree and has roots $\{\alpha_1, \ldots, \alpha_d\}$, possibly over a finite extension of R, then (2.2) also holds. Thus t-sparse multiples of a squarefree f correspond to t-sparse R-vectors in the nullspace of $A_n(\alpha_1, \ldots, \alpha_d)$.

2.1 Finding Bounded-Height Bounded-Degree Sparse Multiples

We now present an algorithm to find the sparsest bounded-degree, bounded-height multiple $h \in \mathbb{Q}[x]$ of an input $f \in \mathbb{Q}[x]$. Since $\mathcal{H}$ is invariant under scaling, we may assume that $f, g, h \in \mathbb{Z}[x]$.

The basic idea is the following: Having fixed the positions at which the multiple h has nonzero coefficients, finding low-height multiple is reduced to finding the nonzero vector with smallest l_∞ norm in the image of a small lattice.

Let $I = \{i_1, \ldots, i_t\}$ be a t-subset of $\{0, \ldots, n\}$, and $A^I_{f,n} \in \mathbb{Z}^{(n+1-t)\times(n-d+1)}$ the matrix $A_{f,n}$ with rows $i_1, \ldots, i_t$ removed. Denote by $B^I_{f,n} \in \mathbb{Z}^{t\times(n-d+1)}$ the matrix consisting of the removed rows $i_1, \ldots, i_t$ of the matrix $A_{f,n}$. Existence of a t-sparse multiple $h = h_{i_1}x^{i_1} + h_{i_2}x^{i_2} + \cdots + h_{i_t}x^{i_t}$ of input f is equivalent to the existence of a vector v_g such that $A^I_{f,n} \cdot v_g = \mathbf{0}$ and $B^I_{f,n} \cdot v_g = [h_{i_1}, \ldots, h_{i_t}]^T$.

Algorithm 2.1: Bounded-Degree Bounded-Height Sparsest Multiple

Input: $f \in \mathbb{Z}[x]$ and $t, n, c \in \mathbb{N}$
Output: A t-sparse multiple of f with $\deg(h) \leq n$ and $\mathcal{H}(h) \leq c$, or "NONE"

1 **for** $s = 2, 3, \ldots, t$ **do**
2 **foreach** *s-subset $I = (1, i_2, \ldots, i_s)$ of $\{1, 2, \ldots, n\}$* **do**
3 Compute matrices $A^I_{f,n}$ and $B^I_{f,n}$ as defined above
4 **if** *$A^I_{f,n}$ does not have full column rank* **then**
5 Compute matrix $C^I_{f,n}$ whose columns span the nullspace of $A^I_{f,n}$
6 $\mathbf{h} \leftarrow$ shortest l_∞ vector in the column lattice of $B^I_{f,n} \cdot C^I_{f,n}$
7 **if** $l_\infty(\mathbf{h}) \leq c$ **then return** $h_1 + h_2 x^{i_2} + \cdots + h_t x^{i_t}$
8 **return** "NONE"

Now let $C^I_{f,n}$ be a matrix whose columns span the nullspace of the matrix $A^I_{f,n}$. Since $A_{f,n}$ has full column rank, the nullspace of $A^I_{f,n}$ has dimension $s \leq t$, and hence $C^I_{f,n} \in \mathbb{Z}^{(n-d+1)\times s}$. Thus, a t-sparse multiple $h = h_{i_1} x^{i_1} + \cdots + h_{i_t} x^{i_t}$ of f exists if and only if there exists a $v \in \mathbb{Z}^t$ such that

$$B^I_{f,n} \cdot C^I_{f,n} \cdot v = [h_{i_1}, \ldots, h_{i_t}]^T. \tag{2.3}$$

Note that $B^I_{f,n} \cdot C^I_{f,n} \in \mathbb{Z}^{t\times s}$.

Algorithm 2.1 outlines this approach. The following lemma uses the Smith normal form to show that Step 5 can be computed efficiently.

Lemma 2.1. *Given $T \in \mathbb{Z}^{k\times\ell}$ with $k \geq \ell$ and nullspace of dimension s, we can compute a $V \in \mathbb{Z}^{s\times\ell}$ such that the image of V equals the nullspace of T. The algorithm requires $O(k\ell^2 s \log \|T\|)$ bit operations (ignoring logarithmic factors).*

Lemma 2.2 shows that Step 6 can be performed efficiently. The proof employs the algorithm of (Ajtai et al., 2001) to find all shortest vectors in the l_2 norm within an approximation factor of $\sqrt{t}$. The shortest l_∞ vector must be among the computed set.

Lemma 2.2. *The shortest l_∞ vector in the image of a matrix $U \in \mathbb{Z}^{t\times t}$ can be computed by a randomized algorithm using $2^{O(t \log t)} \cdot \|U\|^{O(1)}$ bit operations.*

The correctness and efficiency of Algorithm 2.1 can be summarized as follows.

Theorem 2.3. *Algorithm 2.1 correctly computes a t-sparse multiple h of f of degree n and height c, if it exists, with $(\log \mathcal{H}(f))^{O(1)} \cdot n^{O(t)} \cdot 2^{O(t \log t)}$ bit operations. The sparsity s of h is minimal over all multiples with degree less than n and height less than c, and $\deg h$ is minimal over all such s-sparse multiples.*

2.2 Relationship to NP-hard Problems

Note that the above algorithms require time exponential in t, and are only polynomial-time for constant t. It is natural to ask whether there are efficient algorithms which require time polynomial in t. We conjecture this problem is

NP-complete, and point out two relatively recent results of Vardy (1997) and Guruswami and Vardy (2005) on related problems that are known to be hard.

The formulation (2.2) seeks the sparsest vector in the nullspace of a (structured) matrix. For an unstructured matrix over finite fields, this is the problem of finding the minimum distance of a linear code, shown by Vardy (1997) to be NP-complete. The same problem over integers translates into finding the sparsest vector in an integer lattice. It was posed as an open problem in Egner and Minkwitz (1998). Techniques similar to Vardy (1997) prove that this problem is also NP-complete over the integers:

Theorem 2.4. *The problem* SparseLatticeVector *of computing the vector with the least Hamming weight in an integer lattice specified by its basis is* NP*complete.*

Of course, the problem may be easier for structured matrices as in (2.2) However, Guruswami and Vardy (2005) show that maximum likelihood decoding of cyclic codes, which seeks sparse solutions to systems of equations of similar structure to (2.2), is also NP-complete. They do require the freedom to choose a right-hand-side vector, whereas we insist on a sparse vector in the nullspace. While these two results certainly do not prove that the bounded-degree sparsest multiple problem is NP-complete, they support our conjecture that it is.

3 Binomial Multiples over $\mathbb{Q}$

In this section we completely solve the problem of determining if there exists a binomial multiple of a rational input polynomial (i.e., of sparsity $t = 2$). That is, given input $f \in \mathbb{Q}[x]$ of degree d, we determine if there exists a binomial multiple $h = x^m - a \in \mathbb{Q}[x]$ of f, and if so, find such an h with minimal degree. The constant coefficient a will be given as a pair $(r, e) \in \mathbb{Q} \times \mathbb{N}$ representing $r^e \in \mathbb{Q}$. The algorithm requires a number of bit operations which is polynomial in d and $\log \mathcal{H}(f)$. No a priori bounds on the degree or height of h are required. We show that m may be exponential in d, and $\log a$ may be exponential in $\log \mathcal{H}(f)$, and give a family of polynomials with these properties.

Algorithm 3.1 begins by factoring the given polynomial $f \in \mathbb{Q}[x]$ into irreducible factors (using, e.g., the algorithm of Lenstra et al. (1982)). We then show how to find a binomial multiple of each irreducible factor, and finally provide a combining strategy for the different multiples.

The following theorem of Risman (1976) characterizes binomial multiples of irreducible polynomials. Let ϕ be Euler's totient function, the number of positive integers less than or equal to n which are coprime to n.

Fact 3.1 (Risman (1976), Corollary 2.2). *Let $f \in \mathbb{Q}[x]$ be irreducible of degree d. Suppose the least-degree binomial multiple (if one exists) is of degree m. Then there exist $n, t \in \mathbb{N}$ with $n \mid d$ and $\phi(t) \mid d$ such that $m = n \cdot t$.*

Combining Fact 3.1 with number-theoretic bounds from Rosser and Schoenfeld (1962), we obtain the following explicit upper bound on the maximum degree of a binomial multiple of an irreducible polynomial.

Theorem 3.2. *Let $f \in \mathbb{Q}[x]$ be irreducible of degree d. If a binomial multiple of f exists, and has minimal degree m, then $m \leq d \cdot (\lceil 3d \ln \ln d \rceil + 7)$.*

Algorithm 3.1: Lowest degree Binomial Multiple of a Rational Polynomial

Input: $f \in \mathbb{Q}[x]$
Output: The lowest degree binomial multiple $h \in \mathbb{Q}[x]$ of f, or "`NONE`"

1 Factor f into irreducible factors: $f = x^b f_1 f_2 \cdots f_u$
2 **if** *f is not squarefree* **then return** "`NONE`"
3 **for** $i = 1, 2, 3, \ldots, u$ **do**
4 $\quad m_i \leftarrow$ least $k \in \{d_i = \deg f_i, \ldots, (\lceil 3d_i \ln \ln d_i \rceil + 7)d_i\}$ s.t. $x^k \operatorname{rem} f_i \in \mathbb{Q}$
5 $\quad$ **if** *no such m_i is found* **then return** "`NONE`"
6 $\quad$ **else** $r_i \leftarrow x^{m_i} \operatorname{rem} f_i$
7 $m \leftarrow \operatorname{lcm}(m_1, \ldots, m_u)$
8 **foreach** *2-subset* $\{i, j\} \subseteq \{1, \ldots, u\}$ **do**
9 $\quad$ **if** $|r_i|^{m_j} \neq |r_j|^{m_i}$ **then return** "`NONE`"
10 $\quad$ **else if** $sign(r_i^{m/m_i}) \neq sign(r_j^{m/m_j})$ **then** $m \leftarrow 2 \cdot \operatorname{lcm}(m_1, \ldots, m_u)$
11 **return** $x^b(x^m - r_1^{m/m_1})$, *with r_1 and m/m_1 given separately*

The above theorem ensures that for an irreducible f_i, Step 4 of Algorithm 3.1 computes the least-degree binomial multiple $x^{m_i} - r_i$ if it exists, and otherwise correctly reports failure. It clearly runs in polynomial time.

Assume the factorization of f is as computed in Step 1, and moreover f is squarefree (otherwise it cannot have a binomial multiple). If any factor does not have a binomial multiple, neither can the product. If every irreducible factor does have a binomial multiple, Step 4 computes the one with the least degree. The following relates the degrees of the minimal binomial multiple of the input polynomial to those of its irreducible factors.

Lemma 3.3. *Let $f \in \mathbb{Q}[x]$ be such that $f = f_1 \cdots f_u \in \mathbb{Q}[x]$ for distinct, irreducible $f_1, \ldots, f_u \in \mathbb{Q}[x]$. Let $f_i \mid (x^{m_i} - r_i)$ for minimal $m_i \in \mathbb{N}$ and $r_i \in \mathbb{Q}$, and let $f \mid (x^m - r)$ for $r \in \mathbb{Q}$. Then $\operatorname{lcm}(m_1, \ldots, m_u) \mid m$.*

The key step in the proof of Lemma 3.3 is showing that if any m_i does not divide m, then a binomial multiple of f_i with degree $(m \bmod m_i)$ can be constructed, contradicting the minimality of m_i.

Lemma 3.4. *For a polynomial $f \in \mathbb{Q}[x]$ factored into distinct irreducible factors $f = f_1 f_2 \ldots f_u$, with $f_i \mid (x^{m_i} - r_i)$ for $r_i \in \mathbb{Q}$ and minimal such m_i, a binomial multiple of f exists if and only if $|r_i|^{m_j} = |r_j|^{m_i}$ for every pair $1 \leq i, j \leq u$. If a binomial multiple exists, the least-degree binomial multiple of f is $x^m - r_i^{m/m_i}$ such that m either equals the least common multiple of the m_i or twice that number. It can be efficiently checked which of these cases holds.*

The following comes directly from the previous lemma and the fact that Algorithm 3.1 performs polynomially many arithmetic operations.

Theorem 3.5. *Given a polynomial $f \in \mathbb{Q}[x]$, Algorithm 3.1 outputs the least-degree binomial multiple $x^m - r_i^{m/m_i}$ (with r_i and m/m_i output separately) if one exists or correctly reports the lack of a binomial multiple otherwise. Furthermore, it runs in deterministic time $(d + \mathcal{H}(f))^{O(1)}$.*

The constant coefficient of the binomial multiple cannot be output in standard form, but must remain an unevaluated power. Polynomials exist whose minimal binomial multiples have exponentially sized degrees and heights.

Theorem 3.6. *For any $d \geq 841$ there exists a polynomial $f \in \mathbb{Z}[x]$ of degree at most $d \log d$ and height $\mathcal{H}(f) \leq \exp(2d \log d)$ whose minimal binomial multiple $x^m - a$ is such that $m > \exp(\sqrt{d})$ and $\mathcal{H}(a) > 2^{\exp(\sqrt{d})}$.*

4 *t*-sparse Multiples over $\mathbb{Q}$

We examine the problem of computing t-sparse multiples of rational polynomials, for any fixed positive integer t. As with other types of polynomial computations, it seems that cyclotomic polynomials behave quite differently from cyclotomic-free ones. Accordingly, we first examine the case that our input polynomial f consists only of cyclotomic or cyclotomic-free factors. Then we see how to combine them, in the case that none of the cyclotomic factors are repeated.

Specifically, we will show that, given any rational polynomial f which does not have repeated cyclotomic factors, and a height bound $c \in \mathbb{N}$, we can compute a sparsest multiple of f with height at most c, or conclude that none exists, in time polynomial in the size of f and $\log c$ (but exponential in t).

First, notice that multiplying a polynomial by a power of x does not affect the sparsity, and so without loss of generality we may assume all polynomials are relatively prime to x; we call such polynomials *non-original* since they do not pass through the origin.

4.1 The Cyclotomic Case

Suppose the input polynomial f is a product of cyclotomic factors, and write the complete factorization of f as

$$f = \Phi_{i_1}^{e_i} \cdot \Phi_{i_2}^{e_2} \cdots \Phi_{i_k}^{e_k}. \tag{4.1}$$

Now let $m = \operatorname{lcm}(i_1, \ldots, i_k)$. Then m is the least integer such that $\Phi_{i_1} \cdots \Phi_{i_k}$ divides $x^m - 1$. Let $\ell = \max_i e_i$, the maximum multiplicity of any factor of f. This means that $(x^m - 1)^\ell$ is an $(\ell + 1)$-sparse multiple of f. To prove that this is in fact a sparsest multiple of f, we first require the following simple lemma. Here and for the remainder, for a univariate polynomial $f \in \mathsf{F}[x]$, we denote by f' the first derivative with respect to x, that is, $\frac{\mathrm{d}}{\mathrm{d}x} f$.

Lemma 4.1. *Let $h \in \mathbb{Q}[x]$ be a t-sparse and non-original polynomial, and write $h = a_1 + a_2 x^{d_2} + \cdots + a_t x^{d_t}$. Assume the complete factorization of h over $\mathbb{Q}[x]$ is $h = a_t h_1^{e_1} \cdots h_k^{e_k}$, with each h_i monic and irreducible. Then $\max_i e_i \leq t - 1$,*

An immediate consequence is the following:

Corollary 4.2. *Let $f \in \mathbb{Q}[x]$ be a product of cyclotomic factors, written as in (4.1). Then*

$$h = (x^{\operatorname{lcm}(i_1, \ldots, i_k)} - 1)^{\max_i e_i}$$

is a sparsest multiple of f.

4.2 The Cyclotomic-Free Case

We say a polynomial $f \in \mathbb{Q}[x]$ is *cyclotomic-free* if it contains no cyclotomic factors. Here we will show that a sparsest multiple of a cyclotomic-free polynomial must have degree bounded by a polynomial in the size of the input and output.

First we need the following elementary lemma.

Lemma 4.3. *Suppose $f, h \in \mathbb{Q}[x]$ with f irreducible, and k is a positive integer. Then $f^k | h$ if and only if $f | h$ and $f^{k-1} | h'$.*

The following technical lemma provides the basis for our degree bound on the sparsest multiple of a non-cyclotomic polynomial. The proof, omitted, is by induction on k, using the "gap theorem" from (Lenstra, 1999) in the base case.

Lemma 4.4. *Let $f, h_1, h_2, \ldots, h_\ell \in \mathbb{Q}[x]$ be non-original polynomials, where f is irreducible and non-cyclotomic with degree d, and each h_i satisfies $\deg h_i \leq u$ and $\mathcal{H}(h_i) \leq c$. Also let $k, m_1, m_2, \ldots, m_\ell$ be positive integers such that*

$$f^k | (h_1 x^{m_1} + h_2 x^{m_2} + \cdots + h_\ell x^{m_\ell}).$$

Then f^k divides each h_i whenever every "gap length", for $1 \leq i < \ell$, satisfies

$$m_{i+1} - m_i - \deg h_i \geq \frac{1}{2} d \cdot \ln^3(3d) \cdot \ln\left(u^{k-1} c\,(t-1)\right). \tag{4.2}$$

Our main tool in proving that Algorithm 2.1 is useful for computing the sparsest multiple of a rational polynomial, given only a bound c on the height, in polynomial time in the size of f and $\log c$, is the following degree bound on the sparsest height-bounded multiple of a rational polynomial. The proof follows from Lemma 4.4, observing that if the degree of h is sufficiently large compared to the sparsity t, then h must have at least one large gap.

Theorem 4.5. *Let $f \in \mathbb{Q}[x]$ with $\deg f = d$ be cyclotomic-free, and let $t, c \in \mathbb{N}$ such that f has a nonzero t-sparse multiple with height at most c. Denote by n the smallest degree of any such multiple of f. Then n satisfies*

$$n \leq 2(t-1) B \ln B, \tag{4.3}$$

where B is the formula polynomially bounded by d, $\log c$, and $\log t$ defined as

$$B = \frac{1}{2} d^2 \cdot \ln^3(3d) \cdot \ln\left(\hat{c}\,(t-1)^d\right), \tag{4.4}$$

and $\hat{c} = \max(c, 35)$.

In order to compute the sparsest multiple of a rational polynomial with no cyclotomic or repeated factors, we can therefore simply call Algorithm 2.1 with the given height bound c and degree bound as specified in (4.3).

4.3 Handling Cyclotomic Factors

Suppose f is any non-original rational polynomial with no repeated cyclotomic factors. Factor f as $f = f_C \cdot f_D$, where f_C is a squarefree product of cyclotomics and f_D is cyclotomic-free. Write the factorization of f_C as $f_C = \Phi_{i_1} \cdots \Phi_{i_k}$, where Φ_n is the n^{th} cyclotomic polynomial. Since every i^{th} root of unity is also a

Algorithm 4.1: Rational Sparsest Multiple

Input: Bounds $t, c \in \mathbb{N}$ and $f \in \mathbb{Q}[x]$ a non-original polynomial of degree d with no repeated cyclotomic factors

Output: t-sparse multiple h of f with $\mathcal{H}(h) \leq c$, or "`NONE`"

1 Factor f as $f = \Phi_{i_1} \cdot \Phi_{i_2} \cdots \Phi_{i_k} \cdot f_D$, where f_D is cyclotomic-free
2 $n \leftarrow$ degree bound from (4.3)
3 $\hat{h} \leftarrow$ sparsest multiple of f_D with $\mathcal{H}(\hat{h}) \leq c$ and $\deg \hat{h} \leq n$, using Algorithm 2.1
4 $\tilde{h} \leftarrow$ sparsest multiple of f with $\mathcal{H}(h) \leq c$ and $\deg h \leq n$, using Algorithm 2.1
5 **if** $\hat{h} =$"`NONE`" *and* $\tilde{h} =$"`NONE`" **then return** "`NONE`"
6 **else if** $\hat{h} =$"`NONE`" *or* $\text{sparsity}(\tilde{h}) \leq 2 \cdot \text{sparsity}(\hat{h})$ **then return** $\tilde{h}$
7 $m \leftarrow \text{lcm}\{i_1, i_2, \ldots, i_k\}$
8 **return** $\hat{h} \cdot (x^m - 1)$

$(mi)^{\text{th}}$ root of unity for any $m \in \mathbb{N}$, f_C must divide the binomial $x^{\text{lcm}\{i_1,\ldots,i_k\}} - 1$, which is in fact the sparsest multiple of f_C and clearly has minimal height.

Then we will show that the sparsest height-bounded multiple of f is either of small degree, or is equal to the sparsest height-bounded multiple of f_D times the binomial multiple of f_C specified above. Algorithm 4.1 uses this fact to compute the sparsest multiple of any such f.

Theorem 4.6. *Let $f \in \mathbb{Q}[x]$ be a degree-d non-original polynomial with no repeated cyclotomic factors. Given f and integers c and t, Algorithm 4.1 correctly computes a t-sparse multiple h of f satisfying $\mathcal{H}(h) \leq c$, if one exists. The sparsity of h will be minimal over all multiples with height at most c. The cost of the algorithm is $(d \log c)^{O(t)} \cdot 2^{O(t \log t)} \cdot (\log \mathcal{H}(f))^{O(1)}$.*

The main idea in the proof is that, if the sparsest multiple of f has very high degree, then it can be written as $h = h_1 + h_2 x^m$, and both h_1 and h_2 must be sparse multiples of f_D, the cyclotomic-free part of f.

4.4 An Example

Say we want to find a sparsest multiple of the following polynomial over $\mathbb{Q}[x]$.

$$f = x^{10} - 5x^9 + 10x^8 - 8x^7 + 7x^6 - 4x^5 + 4x^4 + x^3 + x^2 - 2x + 4.$$

First factor using (Lenstra et al., 1982) and identify cyclotomic factors:

$$f = \underbrace{(x^2 - x + 1)}_{\Phi_6} \cdot \underbrace{(x^4 - x^3 + x^2 - x + 1)}_{\Phi_{10}} \cdot \underbrace{(x^4 - 3x^3 + x^2 + 6x + 4)}_{f_D}.$$

Next, we calculate a degree bound from Theorem 4.5. Unfortunately, this bound is not very tight (despite being polynomial in the output size); using $t = 10$, $c = 1000$, and f given above, the bound is $n \leq 11\,195\,728$. So for this example, we will use the smaller (but artificial) bound of $n \leq 20$.

The next step is to calculate the sparsest multiples of both f_D and f with degree at most 20 and height at most 1000. Using Algorithm 2.1, these are

$$\hat{h} = x^{12} + 259x^6 + 64.$$
$$\tilde{h} = x^{11} - 3x^{10} + 12x^8 - 9x^7 + 10x^6 - 4x^5 + 9x^4 + 3x^3 + 8.$$

Since the sparsity of $\hat{h}$ is less than half that of $\tilde{h}$, a sparsest multiple is

$$h = (x^{12} + 259x^6 + 64) \cdot (x^{\mathrm{lcm}(6,10)} - 1) = x^{42} + 259x^{36} + 64x^{30} - x^{12} - 259x^6 - 64.$$

5 Sparse Multiples over $\mathbb{F}_q$

We prove that for any constant t, finding the minimal degree t-sparse multiple of an $f \in \mathbb{F}_q[x]$ is harder than finding orders of elements in $\mathbb{F}_{q^e}$. Order finding is reducible to integer factorization and to discrete logarithm, but reductions in the other direction are not known for finite fields (Adleman and McCurley, 1994). However, a fast algorithm for order finding in finite fields would give an efficient procedure for computing primitive elements, "one of the most important unsolved and notoriously hard problems in the computational theory of finite fields" (von zur Gathen and Shparlinski, 1999).

Formal problem definitions are as follows:

$\mathbf{SpMul}^{(t)}_{\mathbb{F}_q}(f, n)$: Given a polynomial $f \in \mathbb{F}_q[x]$ and an integer $n \in \mathbb{N}$, determine if there exists a (nonzero) 2-sparse multiple $h \in \mathbb{F}_q[x]$ of f with $\deg h \leq n$.

$\mathbf{Order}_{\mathbb{F}_{q^e}}(a, n)$: Given an element $a \in \mathbb{F}^*_{q^e}$ and an integer $n < q^e$, determine if there exists a positive integer $m \leq n$ such that $a^m = 1$.

The problem $\mathbf{Order}_{\mathbb{F}_{q^e}}(a, n)$ is well-studied (see for instance Meijer (1996)), and has been used as a primitive in several cryptographic schemes. Note that an algorithm to solve $\mathbf{Order}_{\mathbb{F}_{q^e}}(a, n)$ will allow us to determine the *multiplicative order* of any $a \in \mathbb{F}^*_{q^e}$ (the smallest nonzero m such that $a^m = 1$) with essentially the same cost (up to a factor of $O(e \log q)$) by using binary search.

The reduction from $\mathbf{Order}_{\mathbb{F}_{q^e}}(a, n)$ to $\mathbf{SpMul}^{(t)}_{\mathbb{F}_q}(f, n)$ works as follows: Given an instance of $\mathbf{Order}_{\mathbb{F}_{q^e}}(a, n)$, we first check if the order o_a of a is less than t by brute-force. Otherwise, we construct the minimal polynomial g_{a^i} for each $a^0, a^1, a^2, \ldots, a^{t-1}$. We only keep distinct g_{a^i}, and call the product of these distinct polynomials $f_{a,t}$. We then run the $\mathbf{SpMul}^{(t)}_{\mathbb{F}_q}(f, n)$ subroutine to search for the existence of a degree n, t-sparse multiple of the polynomial $f_{a,t}$.

Theorem 5.1. *Let $a \in \mathbb{F}_q$ be an element of order at least t. Then the least degree t-sparse multiple of $f_{a,t}$ is $x^{o_a} - 1$ where o_a is the order of a.*

The proof uses the null vector formulation of (2.2) in Section 2 to reason that low-degree t-sparse multiples correspond to weight t null vectors of a certain matrix. It is similar to the construction of BCH codes of specified design distance. Of cryptographic interest is that fact that these order-finding polynomials are frequent enough in $\mathbb{F}_q[x]$ so that the reduction also holds in the average case.

Next we give a probabilistic algorithm for finding the least degree binomial multiple for polynomials $f \in \mathbb{F}_q$. This algorithm makes repeated calls to an $\mathbf{Order}_{\mathbb{F}_{q^e}}(a, n)$ (defined in the previous section) subroutine. Combined with the hardness result of the previous section (with t=2), this precisely characterizes the complexity of finding least-degree binomial multiples in terms of the complexity of $\mathbf{Order}_{\mathbb{F}_{q^e}}(a, n)$.

Algorithm 5.1: Least degree binomial multiple of f over $\mathbb{F}_q$

Input: $f \in \mathbb{F}_q[x]$
Output: The least degree binomial multiple h of f

Factor $f = x^b f_1^{e_1} \cdot f_2^{e_2} \cdot f_\ell^{e_\ell}$ for irreducible $f_1, \ldots, f_\ell \in \mathbb{F}_q[x]$, and set $d_i \leftarrow \deg f_i$
for $i = 1, 2, \ldots, \ell$ **do**
    $a_i \leftarrow x \in \mathbb{F}_q[x]/(f_i)$, a root of f_i in the extension $\mathbb{F}_{q^{d_i}}$
    Calculate o_i, the order of a_i in $\mathbb{F}_q[x]/(f_i)$.
$n_1 \leftarrow \mathrm{lcm}(\{o_i / \gcd(o_i, q-1)\})$ for all i such that $d_i > 1$
$n_2 \leftarrow \mathrm{lcm}(\{order(a_i/a_j)\})$ over all $1 \le i, j \le u$
$n \leftarrow \mathrm{lcm}(n_1, n_2)$
$\tilde{h} \leftarrow (x^n - a_1^n)$
$e \leftarrow \lceil \log_p \max e_i \rceil$, the smallest e such that $p^e \ge e_i$ for all i
return $h = x^b (x^n - a_1^n)^{p^e}$

Algorithm 5.1 solves the binomial multiple problem in $\mathbb{F}_q$ by making calls to an $\mathbf{Order}_{\mathbb{F}_{q^e}}(a, n)$ procedure that computes the order of elements in extension fields of $\mathbb{F}_q$. Thus $\mathbf{SpMul}^{(2)}_{\mathbb{F}_q}(f)$ reduces to $\mathbf{Order}_{\mathbb{F}_{q^e}}(a, n)$ in probabilistic polynomial time. Construction of an irreducible polynomial (required for finite field arithmetic) as well as the factoring step in the algorithm make it probabilistic.

Theorem 5.2. *Given $f \in \mathbb{F}_q[x]$ of degree d, Algorithm 5.1 correctly computes a binomial multiple h of f with least degree. It uses at most d^2 calls to a routine for order finding in $\mathbb{F}_{q^e}$, for various $e \le d$, and $d^{O(1)}$ other operations in $\mathbb{F}_q$. It is probabilistic of the Las Vegas type.*

The proof of correctness (omitted here) works in three steps. The algorithm is first shown to work correctly for irreducible polynomials, then for squarefree polynomials and finally for all polynomials.

6 Conclusions and Open Questions

We have presented an efficient algorithm to compute the least-degree binomial multiple of any rational polynomial. We can also compute t-sparse multiples of rational polynomials that do not have repeated cyclotomic factors, for any fixed t, and given a bound on the height of the multiple.

We have also shown that, even for fixed t, finding a t-sparse multiple of a degree-d polynomial over $\mathbb{F}_q[x]$ is at least as hard as finding the orders of elements in $\mathbb{F}_{q^d}$. In the $t = 2$ case, there is also a probabilistic reduction in the other direction, so that computing binomial multiples of degree-d polynomials over $\mathbb{F}_q[x]$ probabilisticly reduces to order finding in $\mathbb{F}_{q^d}$.

Several important questions remain unanswered. Although we have an unconditional algorithm to compute binomial multiples of rational polynomials, computing t-sparse multiples for fixed $t \ge 3$ requires an *a priori* height bound on the output as well as the requirement that the input contains no repeated cyclotomic factors. Removing these restrictions would be desirable (if possible).

Regarding lower bounds, we know that computing t-sparse multiples over finite fields is at least as hard as order finding, a result which is tight (up to

randomization) for $t = 2$, but for larger t we believe the problem is even harder. Specifically, we suspect that computing t-sparse multiples is NP-complete over both $\mathbb{Q}$ and $\mathbb{F}_q$, when t is a parameter in the input.

References

L. M. Adleman and K. S. McCurley. Open problems in number-theoretic complexity. II. In *Algorithmic number theory (Ithaca, NY, 1994)*, volume 877 of *Lecture Notes in Comput. Sci.*, pages 291–322. Springer, Berlin, 1994.

L. El Aimani and J. von zur Gathen. Finding low weight polynomial multiples using lattices. Cryptology ePrint Archive, Report 2007/423, 2007. `http://eprint.iacr.org/2007/423.pdf`.

M. Ajtai, R. Kumar, and D. Sivakumar. A sieve algorithm for the shortest lattice vector problem. In *Symp. Theory of Computing (STOC'01)*, pages 601–610, 2001.

J.-P. Aumasson, M. Finiasz, W. Meier, and S. Vaudenay. TCHo: a hardware-oriented trapdoor cipher. In *ACISP'07: Proceedings of the 12th Australasian conference on Information security and privacy*, pages 184–199, Berlin, Heidelberg, 2007. Springer-Verlag. ISBN 978-3-540-73457-4.

E. R. Berlekamp, R. J. McEliece, and H. C. van Tilborg. On the inherent intractability of certain coding problems. *IEEE Transactions on Information Theory*, 24(3), 1978.

R. P. Brent and P. Zimmermann. Algorithms for finding almost irreducible and almost primitive trinomials. In *Primes and Misdemeanours: Lectures in Honour of the Sixtieth Birthday of Hugh Cowie Williams, Fields Institute*, page 212, 2003.

F. Didier and Y. Laigle-Chapuy. Finding low-weight polynomial multiples using discrete logarithms. In *Proc. IEEE International Symposium on Information Theory (ISIT 2007)*, pages 1036–1040, 2007.

S. Egner and T. Minkwitz. Sparsification of rectangular matrices. *J. Symb. Comput.*, 26(2):135–149, 1998.

I. Z. Emiris and I. S. Kotsireas. Implicitization exploiting sparseness. In *Geometric and algorithmic aspects of computer-aided design and manufacturing*, volume 67 of *DIMACS Ser. Discrete Math. Theoret. Comput. Sci.*, pages 281–297, 2005.

J. von zur Gathen and I. Shparlinski. Constructing elements of large order in finite fields. In *Applied Algebra, Algebraic Algorithms and Error-Correcting Codes*, volume 1719 of *Lecture Notes in Computer Science*, pages 730–730. Springer, 1999.

V. Guruswami and A. Vardy. Maximum-likelihood decoding of Reed-Solomon codes is NP-hard. In *SODA '05: Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 470–478, 2005.

A. K. Lenstra, H. W. Lenstra, Jr., and L. Lovász. Factoring polynomials with rational coefficients. *Math. Ann.*, 261(4):515–534, 1982.

H. W. Lenstra, Jr. Finding small degree factors of lacunary polynomials. In *Number theory in progress, Vol. 1*, pages 267–276. De Gruyter, Berlin, 1999.

A. R. Meijer. Groups, factoring, and cryptography. *Math. Mag.*, 69(2):103–109, 1996.

L. J. Risman. On the order and degree of solutions to pure equations. *Proc. Amer. Math. Soc.*, 55(2):261–266, 1976.

J. B. Rosser and L. Schoenfeld. Approximate formulas for some functions of prime numbers. *Ill. J. Math.*, 6:64–94, 1962.

H.R. Sadjadpour, N.J.A. Sloane, M. Salehi, and G. Nebe. Interleaver design for turbo codes. *IEEE J. Selected Areas in Communications*, 19(5):831 –837, may. 2001.

A. Vardy. The intractability of computing the minimum distance of a code. *IEEE Transactions on Information Theory*, 43(6):1757–1766, 1997.